Native-Task Performance Test Report

Intel Software

Wang, Huafeng, Huafeng.wang@intel.com
Zhong, Xiang, xiang.zhong@intel.com

1. Background

2. Related Work

3. Preliminary Experiments

3.1 Experimental Environment

Workbench	Peculiarity
Wordcount	CPU-intensive
Sort	IO-intensive
DFSIO	IO-intensive
Pagerank	Map :CPU-intensive Reduce :IO-intensive
Hivebench-Aggregation	Map :CPU-intensive Reduce :IO-intensive
Hivebench-Join	CPU-intensive
Terasort	Map :CPU-intensive Reduce : IO-intensive
K-Means	Iteration stage: CPU-intensive Classification stage: IO-intensive
Nutchindexing	CPU-intensive & IO-intensive

Cluster settings

Hadoop version	1.0.3-Intel (patched with native task)
Cluster size	4
Disk per machine	7 SATA Disk per node
Network	GbE network
CPU	E5-2680(32 core per node)
L3 Cache size	20480 KB
Memory	64GB per node
Map Slots	3*32+1*26=122
Reduce Slots	3*16+1*13=61

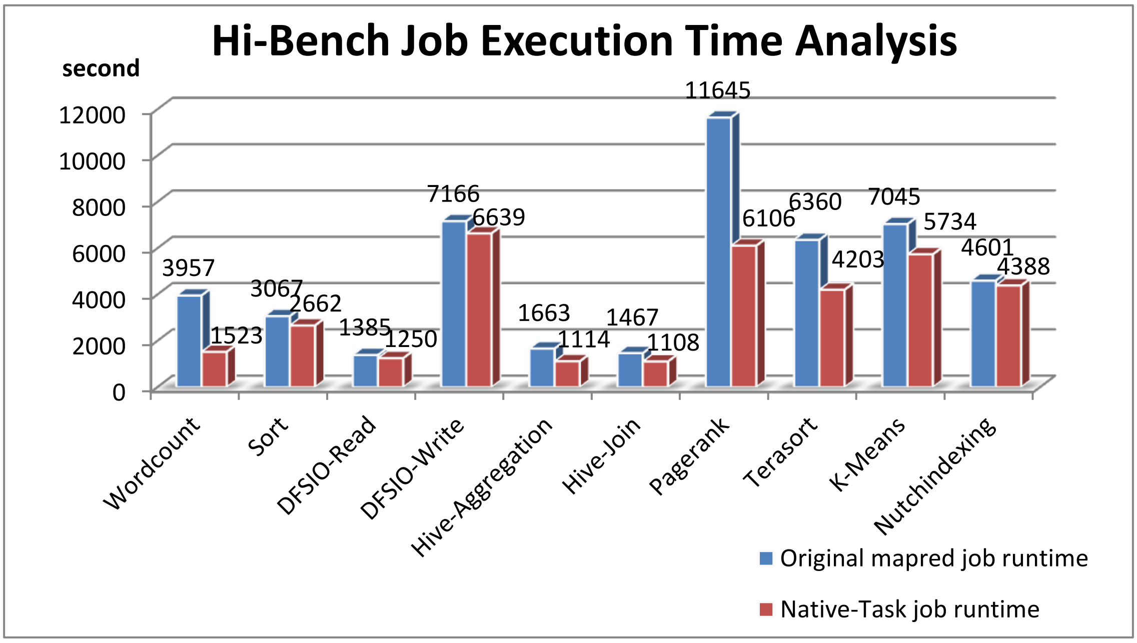
Job Configuration

io.sort.mb	1GB
compression	Enabled
Compression algo	snappy
Dfs.block.size	256MB
Io.sort.record.percent	0.2
Dfs replica	3

3.2 Performance Metrics

	Data size before compression	Data size after compression	Native job run time(s	Original job run time(s)	Job performance improve-	Map stage performance

			)		ment	improve ment
Wordcount	1TB	500GB	1523.43	3957.11	159.8%	159.8%
Sort	500GB	249GB	2662.43	3066.97	15.2%	45.4%
DFSIO-Read	1TB	NA	1249.68	1384.52	10.8%	26%
DFSIO-Write	1TB	NA	6639.22	7165.97	7.9%	7.9%
Pagerank	Pages:500M Total:481GB	217GB	6105.71	11644.63	90.7%	133.8%
Hive-Aggregation	Uservisits:5G Pages:600M Total:820GB	345GB	1113.82	1662.74	49.3%	76.2%
Hive-Join	Uservisits:5G Pages:600M Total:860GB	382GB	1107.55	1467.08	32.5%	42.8%
Terasort	1TB	NA	4203.35	6360.49	51.3%	109.1%
K-Means	Clusters:5 Samples:2G Inputfilesecondample:400M Total:378GB	350GB	5734.11	7045.62	22.9%	22.9%
Nutchindexing	Pages:40M 22G	NA	4387.6	4600.56	4.9%	13.2%

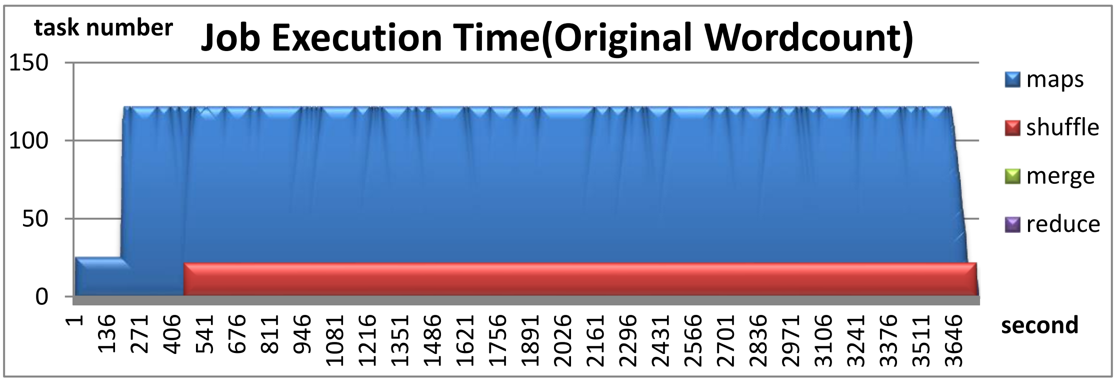


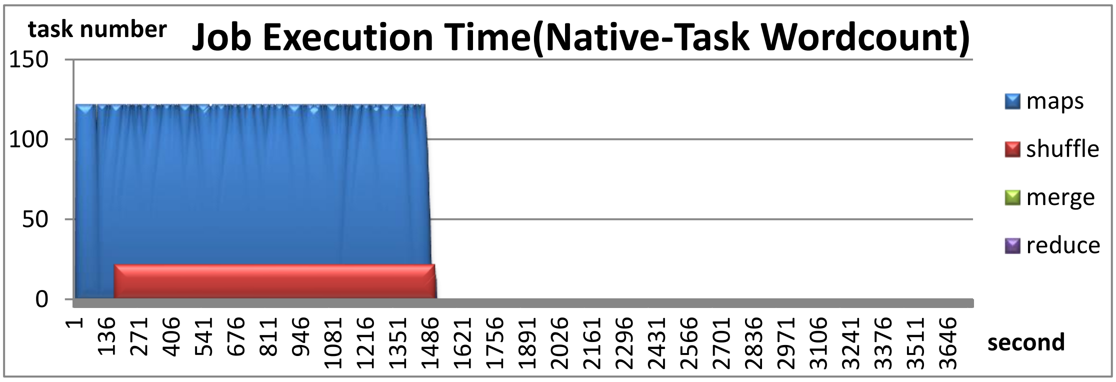
3.3 Results

3.3.1 Wordcount

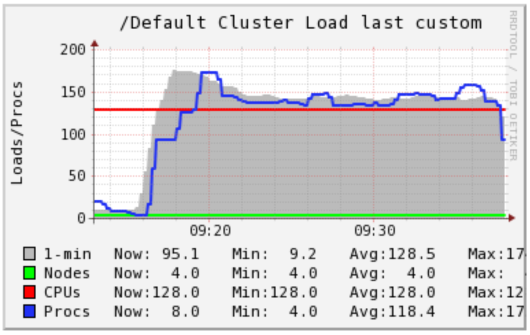
- Job Details:

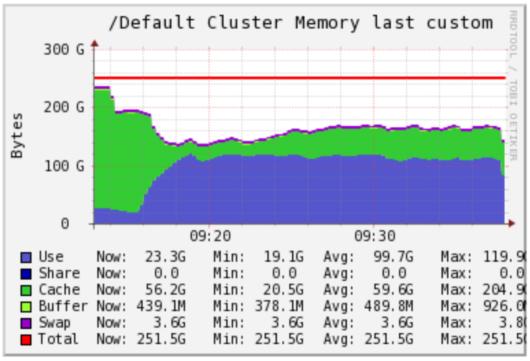
Name	Maps	Reducers
wordcount	1984	22

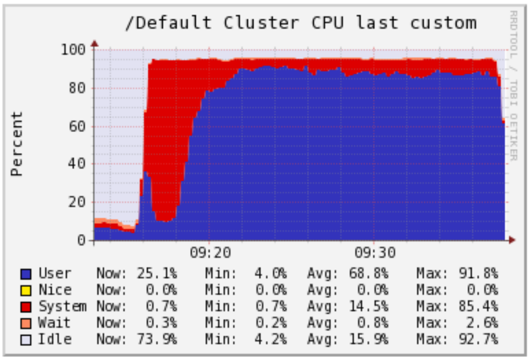


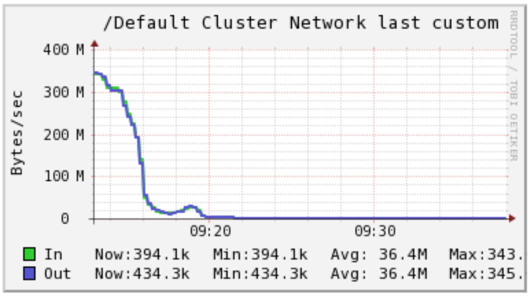


- Native-Task running state:





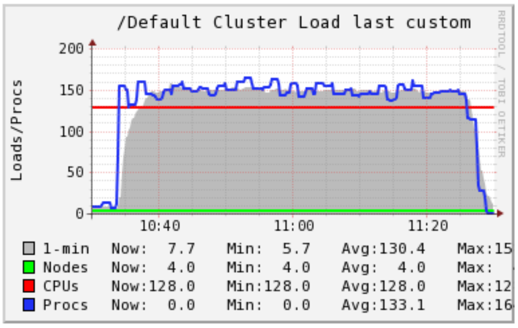


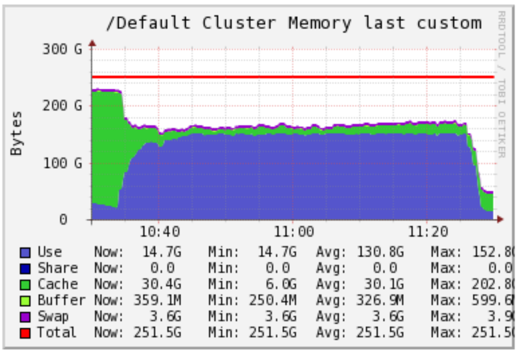


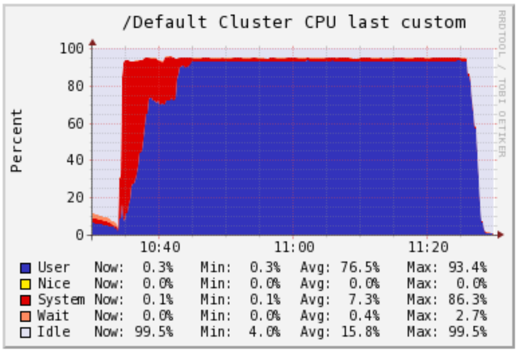
Start time: 9:14

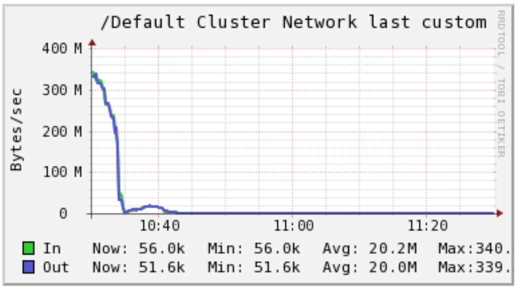
Finish time: 9:37

- Original running state:









Start time: 10:32

Finish time: 11:28

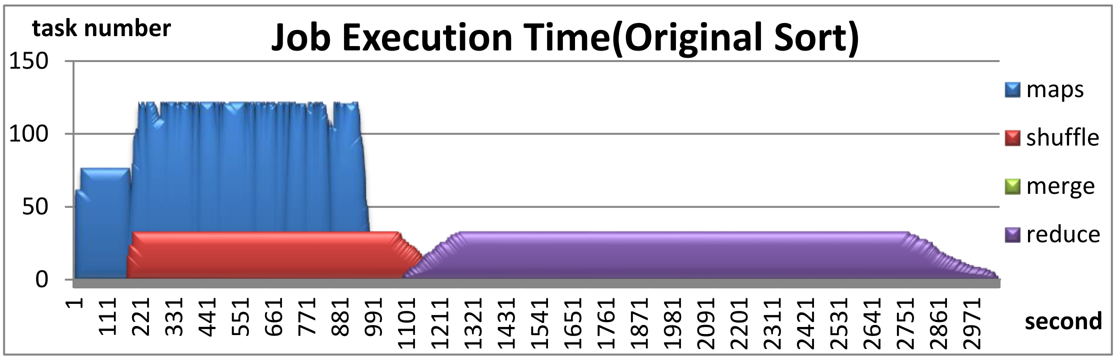
Analysis

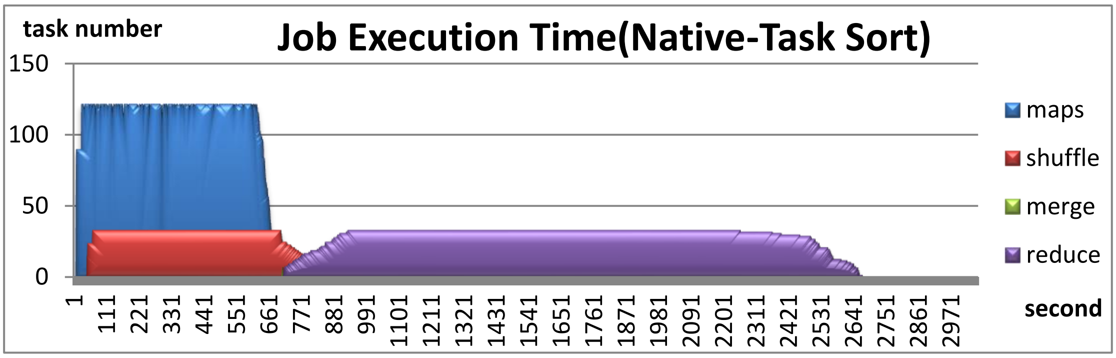
Wordcount is a CPU-intensive workload and it's map stage run through the whole job. So the native-task has a huge performance improvement.

3.3.2 Sort

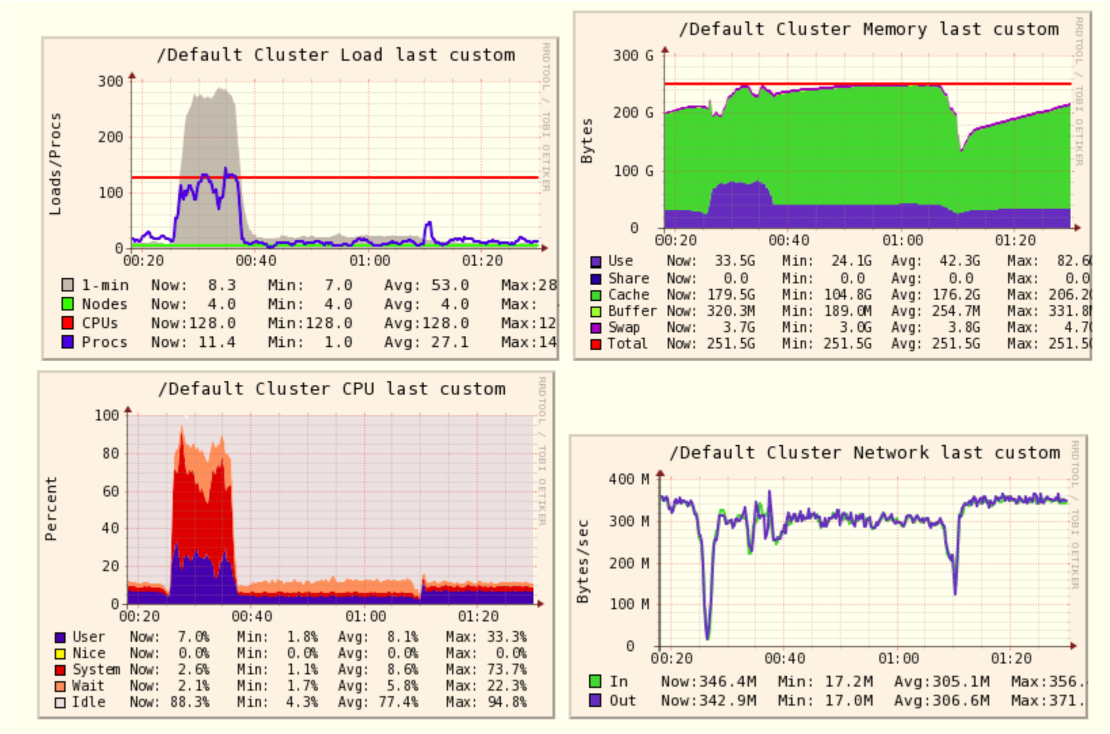
- Job Details:

Name	Maps	Reducers
sorter	1024	33





- Native-Task running state:



Start time :00:25

Finish time :1:10

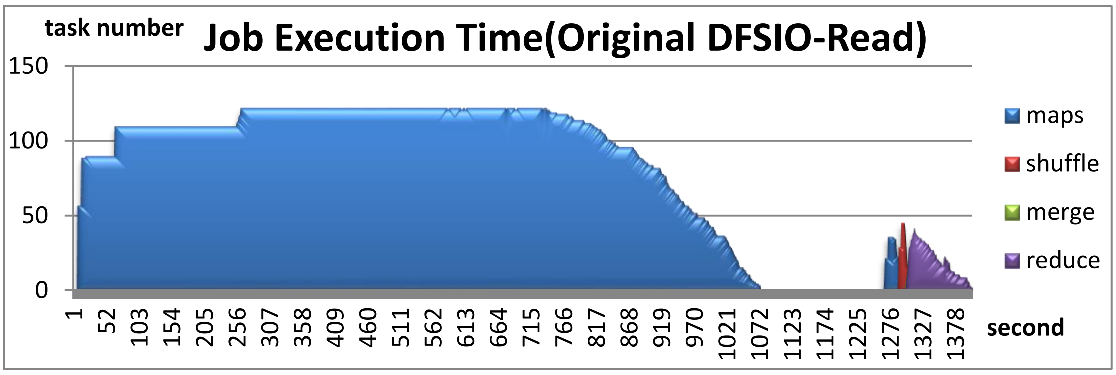
Analysis

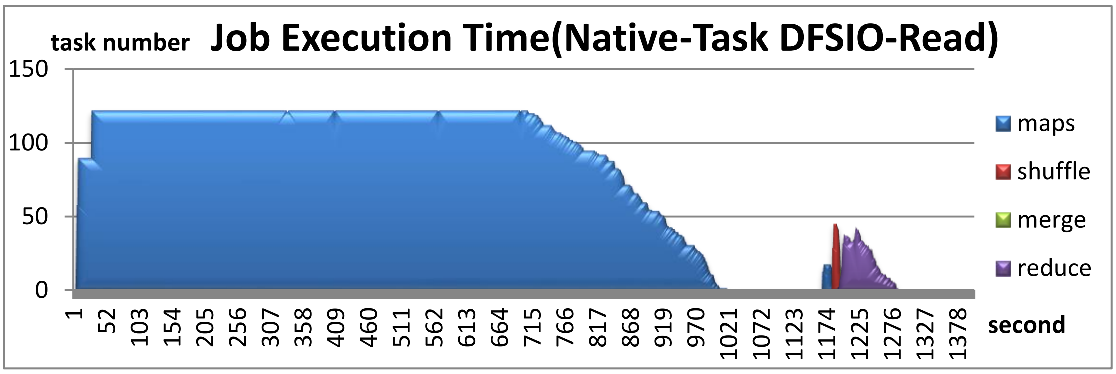
Sort is IO-intensive at both map and reduce stage. We can see that it’s reduce time occupy the most of whole job running time, because of that, the performance improvement is limited.

3.3.3 DFSIO-Read

- Job Details:

Name	Maps	Reducers
Datatools.jar	256	1
Result Analyzer	50	63

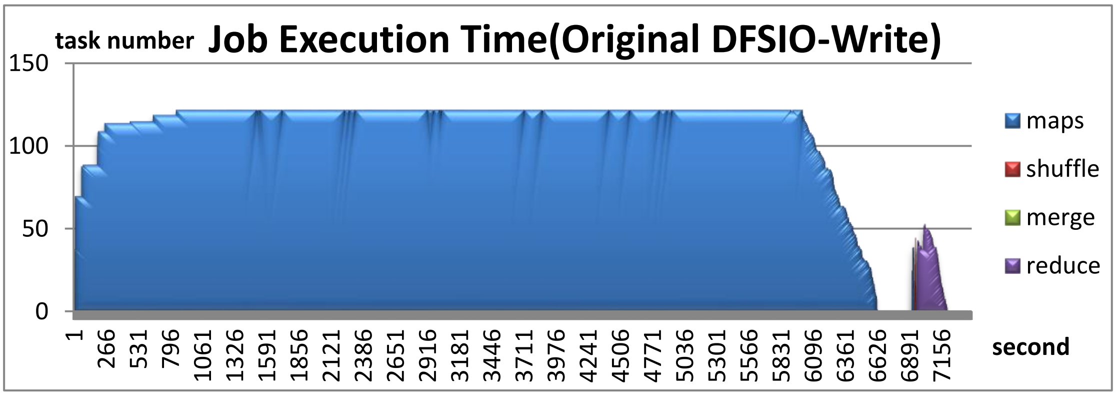


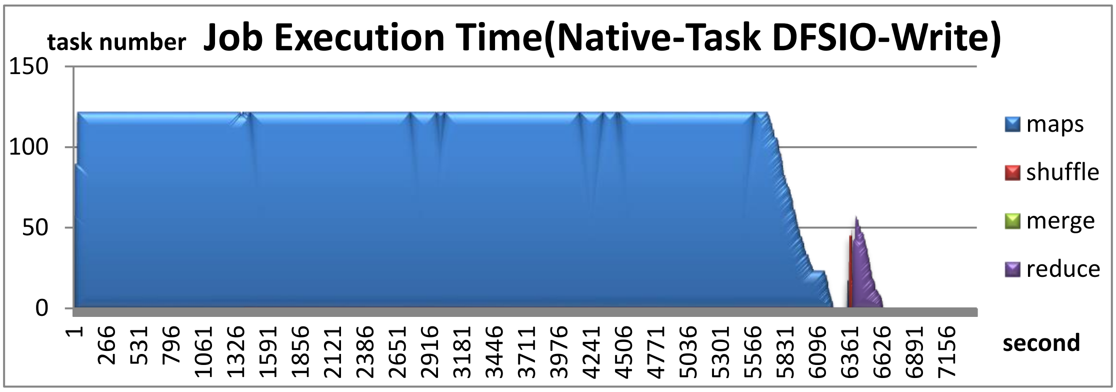


3.3.4 DFSIO-Write

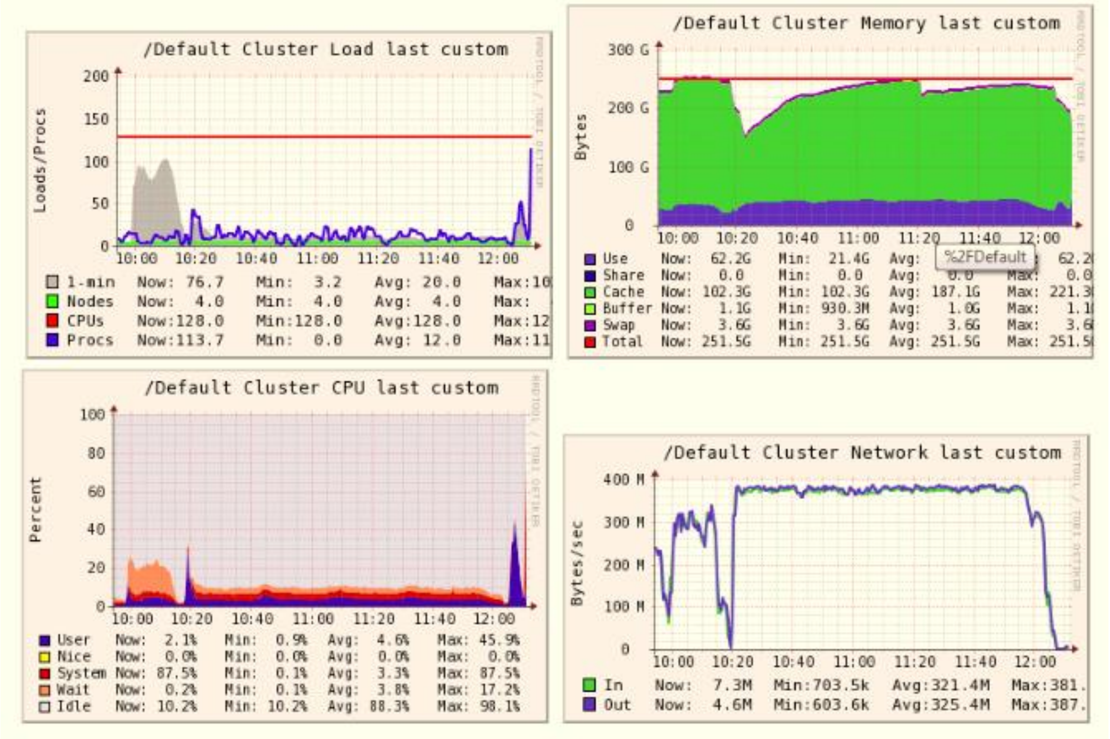
- Job Details:

Name	Maps	Reducers
Datatools.jar	512	1
Result Analyzer	50	63





- Native-Task running state:


/Default Cluster Load last custom
Loads/Procs
200
150
100
50
0
10:00 10:20 10:40 11:00 11:20 11:40 12:00
1-min Now: 76.7 Min: 3.2 Avg: 20.0 Max:16
Nodes Now: 4.0 Min: 4.0 Avg: 4.0 Max:
CPUs Now:128.0 Min:128.0 Avg:128.0 Max:12
Procs Now:113.7 Min: 0.0 Avg: 12.0 Max:11
/Default Cluster Memory last custom
Bytes
300 G
200 G
100 G
0
10:00 10:20 10:40 11:00 11:20 11:40 12:00
Use Now: 62.2G Min: 21.4G Avg: %2FDefault 62.2
Share Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Cache Now: 102.3G Min: 102.3G Avg: 187.1G Max: 221.3
Buffer Now: 1.1G Min: 930.3M Avg: 1.0G Max: 1.1
Swap Now: 3.6G Min: 3.6G Avg: 3.6G Max: 3.6
Total Now: 251.5G Min: 251.5G Avg: 251.5G Max: 251.5
/Default Cluster CPU last custom
Percent
100
80
60
40
20
0
10:00 10:20 10:40 11:00 11:20 11:40 12:00
User Now: 2.1% Min: 0.9% Avg: 4.6% Max: 45.9%
Nice Now: 0.0% Min: 0.0% Avg: 0.0% Max: 0.0%
System Now: 87.5% Min: 0.1% Avg: 3.3% Max: 87.5%
Wait Now: 0.2% Min: 0.1% Avg: 3.8% Max: 17.2%
Idle Now: 10.2% Min: 10.2% Avg: 88.3% Max: 98.1%
/Default Cluster Network last custom
Bytes/sec
400 M
300 M
200 M
100 M
0
10:00 10:20 10:40 11:00 11:20 11:40 12:00
In Now: 7.3M Min:703.5k Avg:321.4M Max:381.
Out Now: 4.6M Min:603.6k Avg:325.4M Max:387.

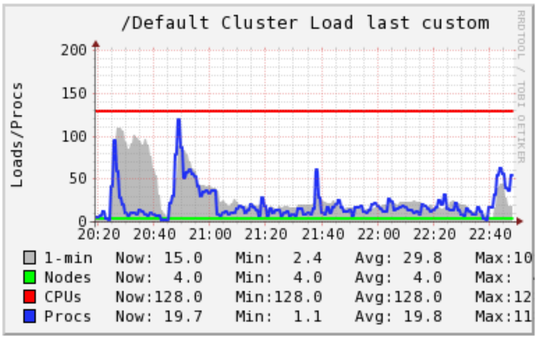
Aggregation start time: 9:58

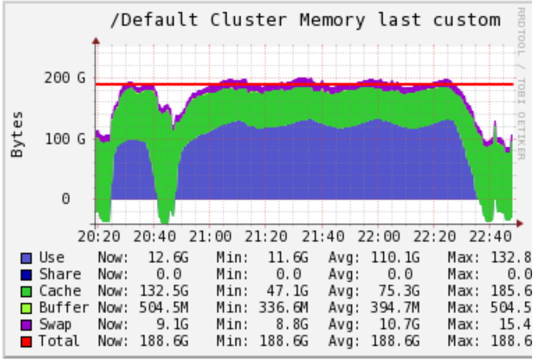
Aggregation finish time: 10:19

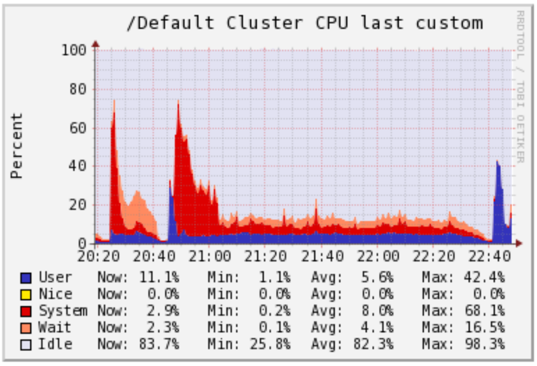
Join start time: 10:19

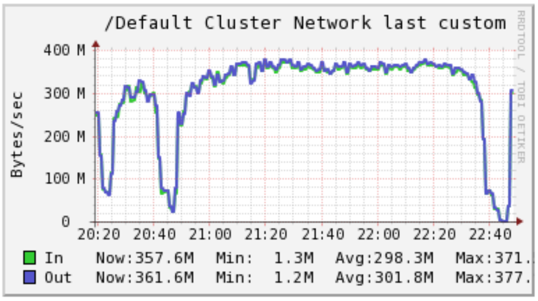
Join finish time: 12:10

- Original running state:









Aggregation start time: 20:22

Aggregation finish time: 20:46

Join start time: 20:46

Join finish time: 22:45

Analysis

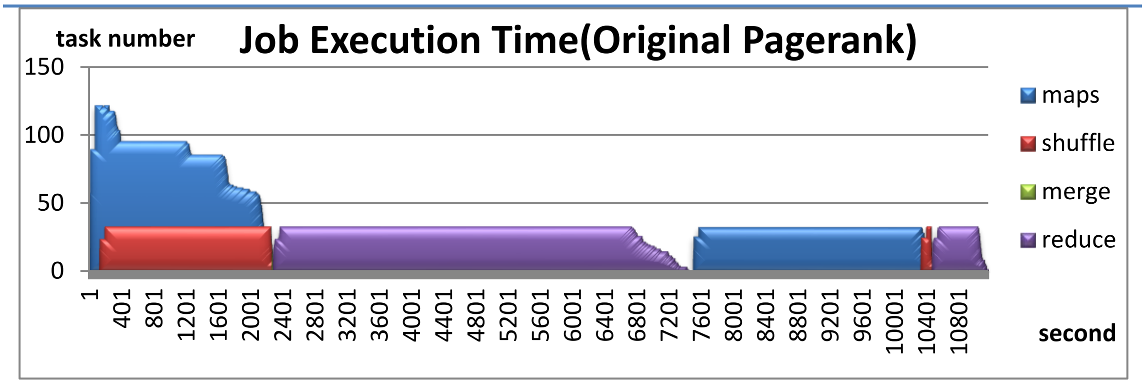
DFSIO is IO-intensive both at read and write stage. It’s bottleneck is network bandwidth so the performance improvement is limited.

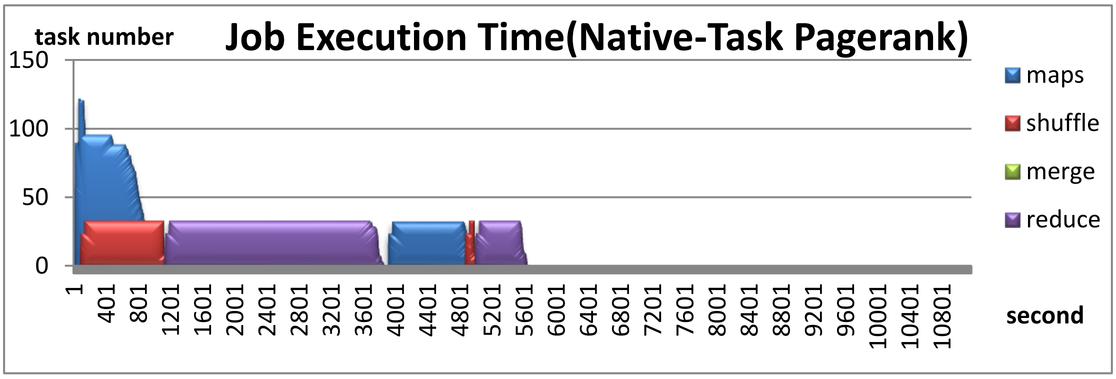
3.3.5 Pagerank

- Job Details:

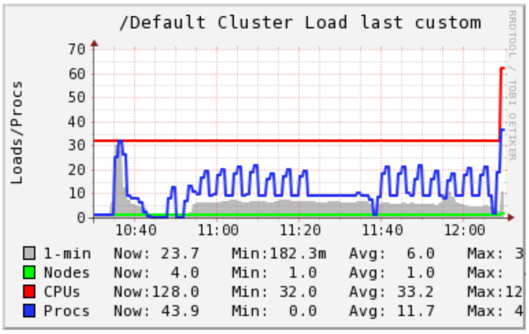
Name	Maps	Reducers
Pagerank_Stage1	163	33

Pagerank_Stage2	33	33

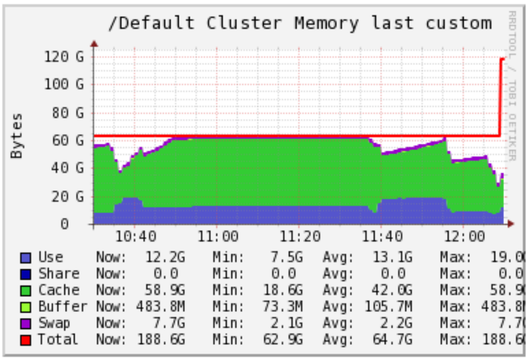




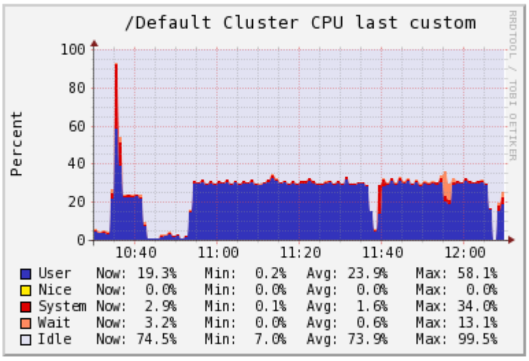
- Native-Task running state:



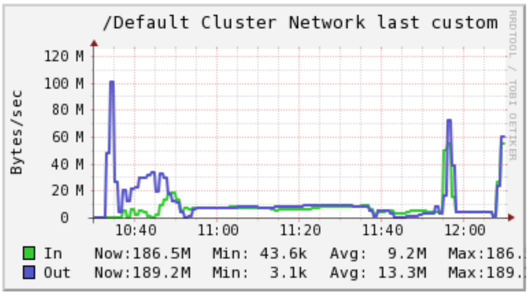
```
□ 1-min   Now: 23.7    Min:182.3m   Avg:  6.0    Max: 3
■ Nodes   Now:  4.0    Min:  1.0    Avg:  1.0    Max:
■ CPUs    Now:128.0    Min: 32.0    Avg: 33.2    Max:12
■ Procs   Now: 43.9    Min:  0.0    Avg: 11.7    Max: 4
```



```
■ Use    Now:  12.2G   Min:   7.5G   Avg:  13.1G   Max:  19.0
■ Share  Now:   0.0    Min:   0.0    Avg:   0.0    Max:   0.0
■ Cache  Now:  58.9G   Min:  18.6G   Avg:  42.0G   Max:  58.9
■ Buffer Now: 483.8M   Min:  73.3M   Avg: 105.7M   Max: 483.8
■ Swap   Now:   7.7G   Min:   2.1G   Avg:   2.2G   Max:   7.7
■ Total  Now: 188.6G   Min:  62.9G   Avg:  64.7G   Max: 188.6
```



```
■ User   Now: 19.3%   Min:  0.2%   Avg: 23.9%   Max: 58.1%
■ Nice   Now:  0.0%   Min:  0.0%   Avg:  0.0%   Max:  0.0%
■ System Now:  2.9%   Min:  0.1%   Avg:  1.6%   Max: 34.0%
■ Wait   Now:  3.2%   Min:  0.0%   Avg:  0.6%   Max: 13.1%
□ Idle   Now: 74.5%   Min:  7.0%   Avg: 73.9%   Max: 99.5%
```

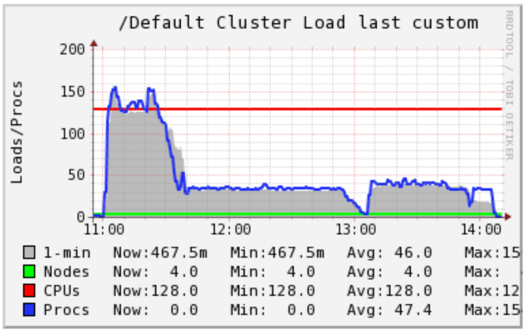


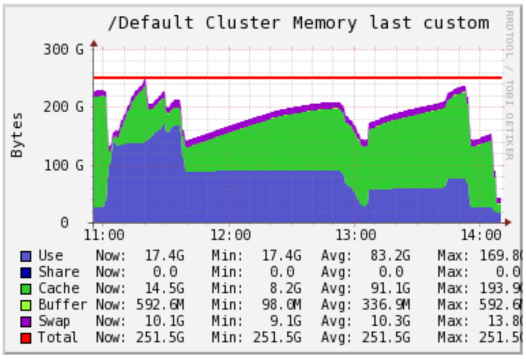
```
■ In    Now:186.5M   Min: 43.6k   Avg:  9.2M   Max:186.
■ Out   Now:189.2M   Min:  3.1k   Avg: 13.3M   Max:189.
```

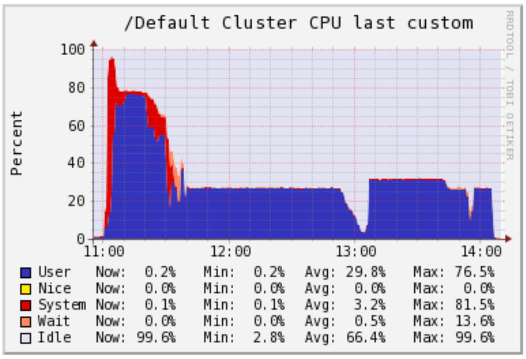
Start time: 10:33

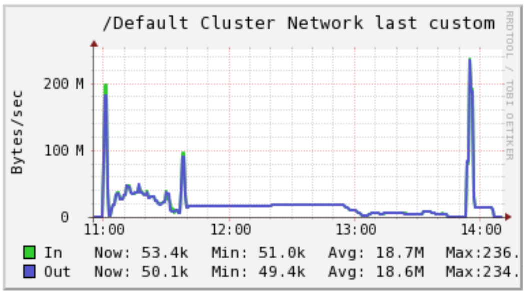
Finish time: 12:08

- Original running state:









Start time: 10:59

Finish time: 14:06

Analysis

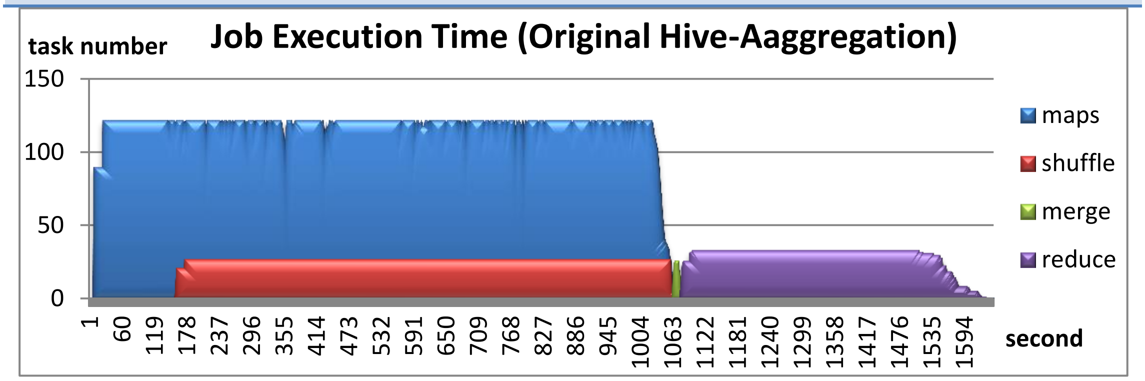
Pagerank is a CPU-intensive workload and it's map stage take about 50% of the whole job running time. So the performance improvement is obvious.

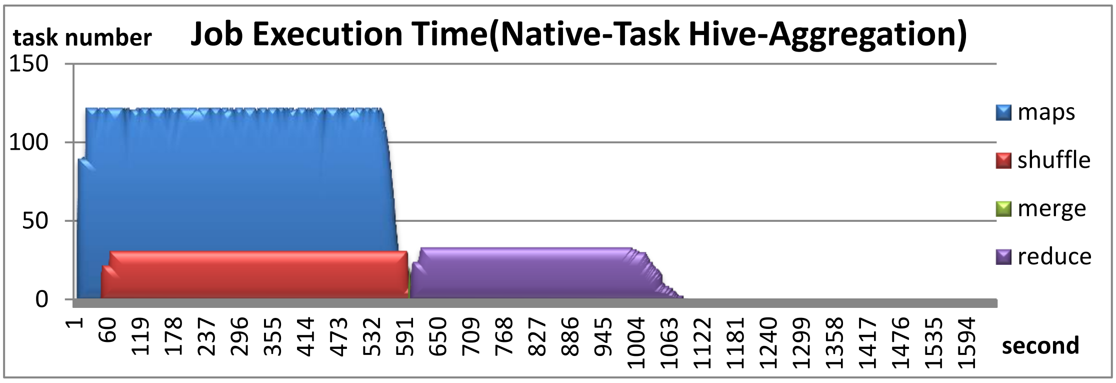
3.3.6 Hive-Aggregation

- Job Details:

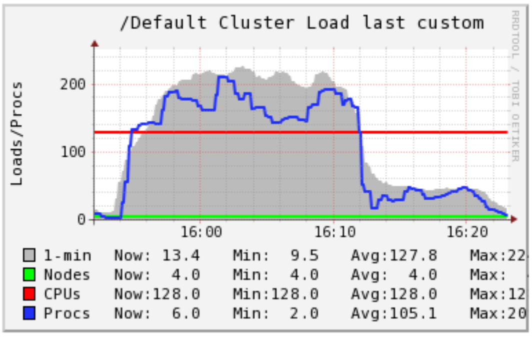
Name	Maps	Reducers
INSERT OVERWRITE TABLE	1386	33

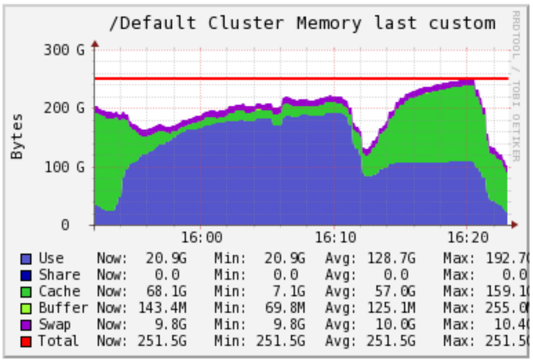
uservisits...sourceIP(Stage-1)

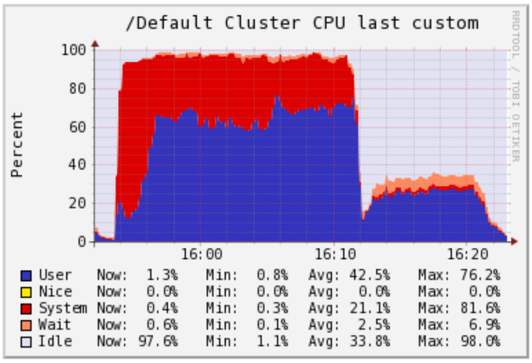


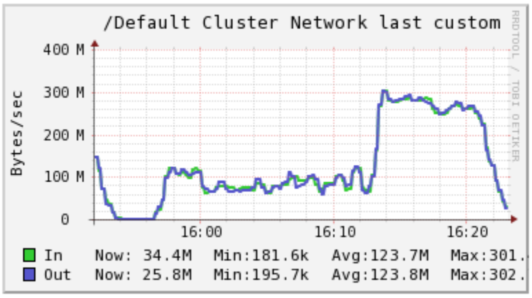


- Original running state:









Start time: 15:52

Finish time :16:22

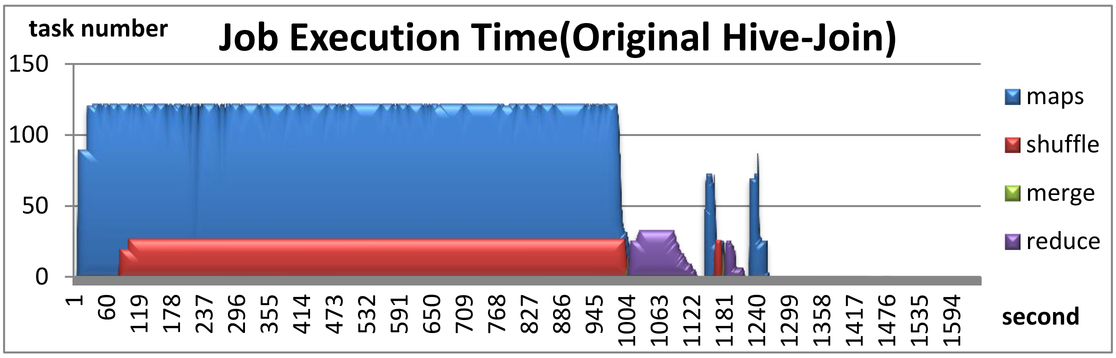
Analysis

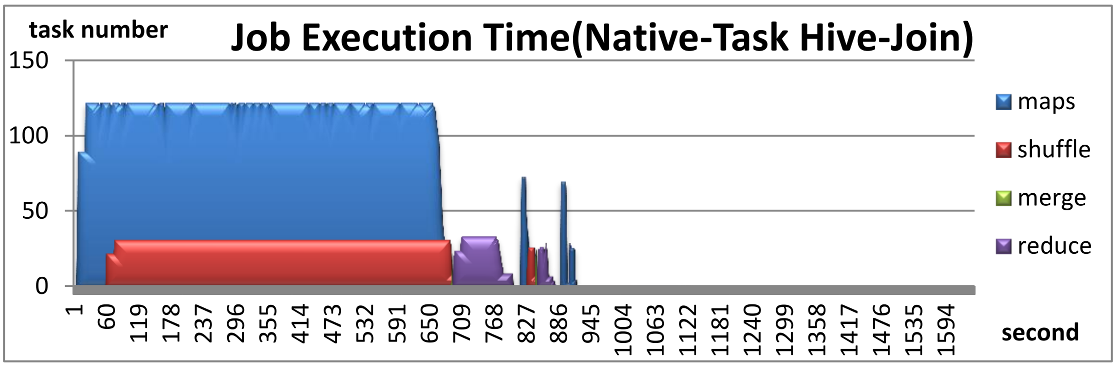
Hive-Aggregation is CPU-intensive at map stage and IO-intensive at reduce stage. It's map stage occupy the most of running time and when it comes to reduce stage, network bandwidth limits the performance. So the performance improvement at map stage is obvious.

3.3.7 Hive-Join

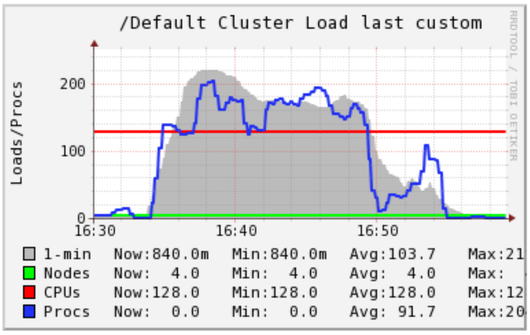
- Job Details:

Name	Maps	Reducers
INSERT OVERWRITE TABLE rankings_uservisi...1(Stage-1)	1551	33
INSERT OVERWRITE TABLE rankings_uservisi...1(Stage-2)	99	33
INSERT OVERWRITE TABLE rankings_uservisi...1(Stage-3)	99	1
INSERT OVERWRITE TABLE rankings_uservisi...1(Stage-4)	1	1

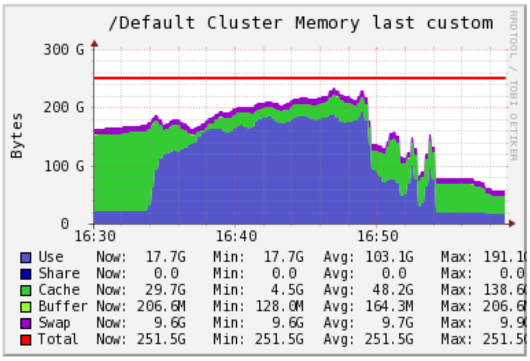




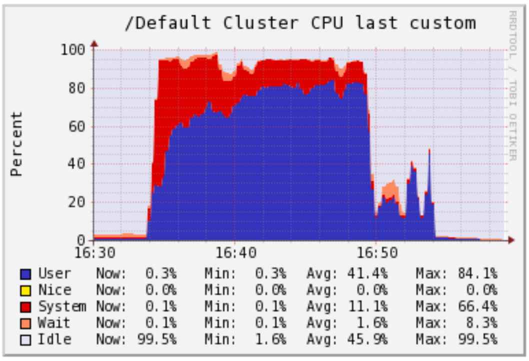
- Original running state:



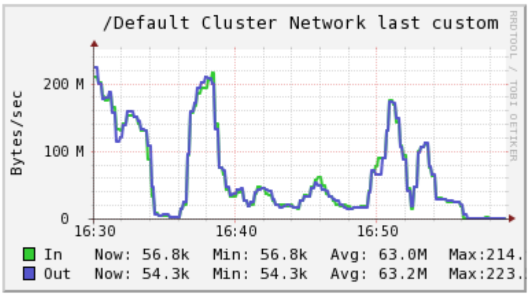
```
□ 1-min   Now:840.0m  Min:840.0m  Avg:103.7   Max:21
🟩 Nodes   Now:  4.0    Min:  4.0    Avg:  4.0    Max:
🟥 CPUs    Now:128.0    Min:128.0    Avg:128.0    Max:12
🟦 Procs   Now:  0.0    Min:  0.0    Avg: 91.7    Max:20
```



```
🟦 Use    Now:  17.7G  Min:  17.7G  Avg: 103.1G  Max: 191.1
🟦 Share  Now:   0.0   Min:   0.0   Avg:   0.0   Max:   0.0
🟩 Cache  Now:  29.7G  Min:   4.5G  Avg:  48.2G  Max: 138.6
🟩 Buffer Now: 206.6M  Min: 128.0M  Avg: 164.3M  Max: 206.6
🟪 Swap   Now:   9.6G  Min:   9.6G  Avg:   9.7G  Max:   9.9
🟥 Total  Now: 251.5G  Min: 251.5G  Avg: 251.5G  Max: 251.5
```



```
🟦 User   Now:  0.3%  Min:  0.3%  Avg: 41.4%  Max: 84.1%
🟨 Nice   Now:  0.0%  Min:  0.0%  Avg:  0.0%  Max:  0.0%
🟥 System Now:  0.1%  Min:  0.1%  Avg: 11.1%  Max: 66.4%
🟧 Wait   Now:  0.1%  Min:  0.1%  Avg:  1.6%  Max:  8.3%
□ Idle   Now: 99.5%  Min:  1.6%  Avg: 45.9%  Max: 99.5%
```



```
🟩 In    Now: 56.8k  Min: 56.8k  Avg: 63.0M  Max:214.
🟦 Out   Now: 54.3k  Min: 54.3k  Avg: 63.2M  Max:223.
```

Start time: 16:32

Finish time :16:58

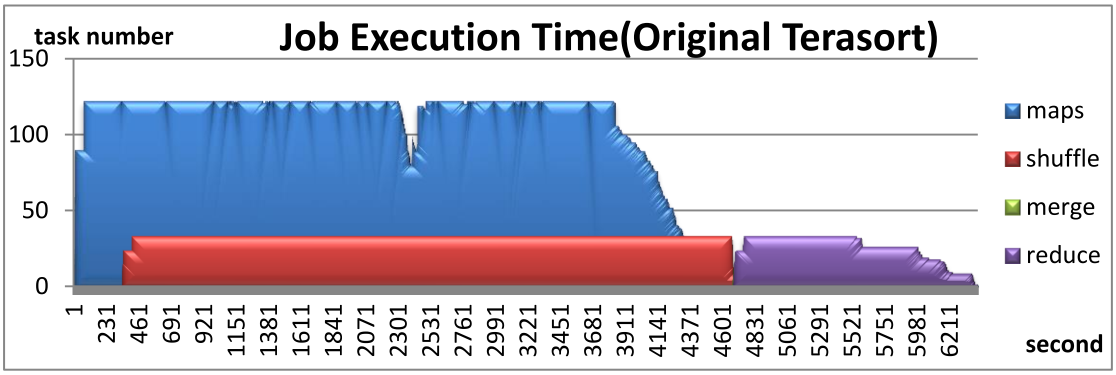
Analysis

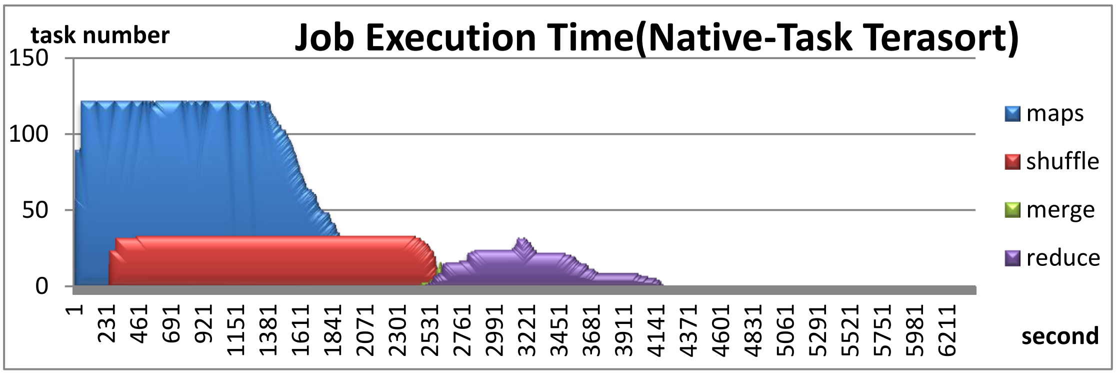
Hive-join is a CPU-intensive workload and it's map stage takes a high percent of whole running time. So we can see at map stage, the performance is improved by native-task.

3.3.8 Terasort

- Job Details:

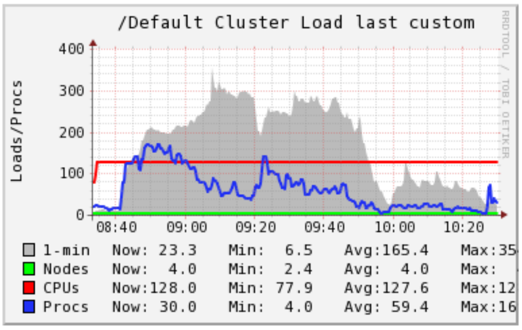
Name	Maps	Reducers
Terasort	458	34

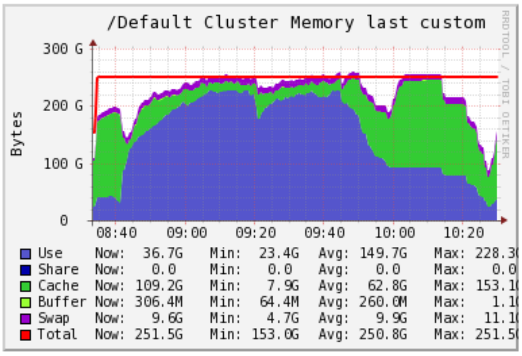


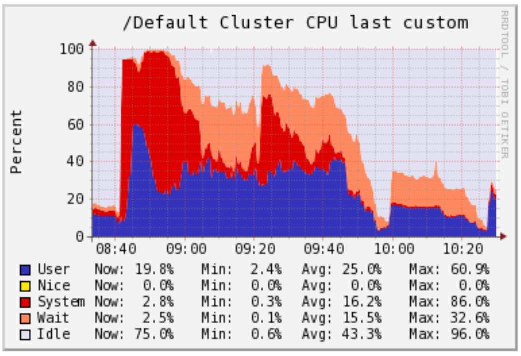


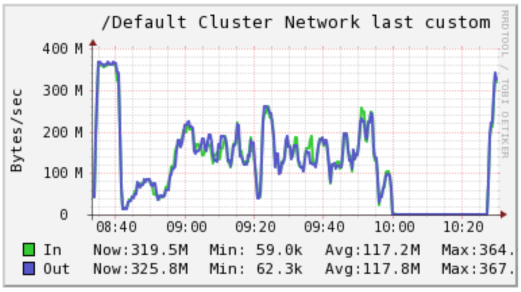
- Native-Task running state:

- Original running state:









Start time: 8:39

Finish time: 10:24

Analysis

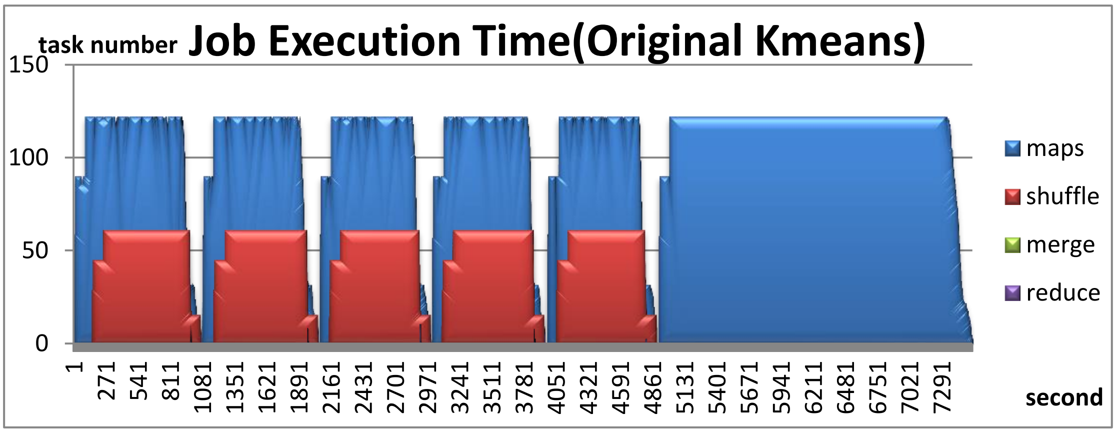
Terasort is CPU-intensive at map stage and IO-intensive at reduce stage.It's map stage occupy the majority of the running time so there is a huge performance improvement at map stage.

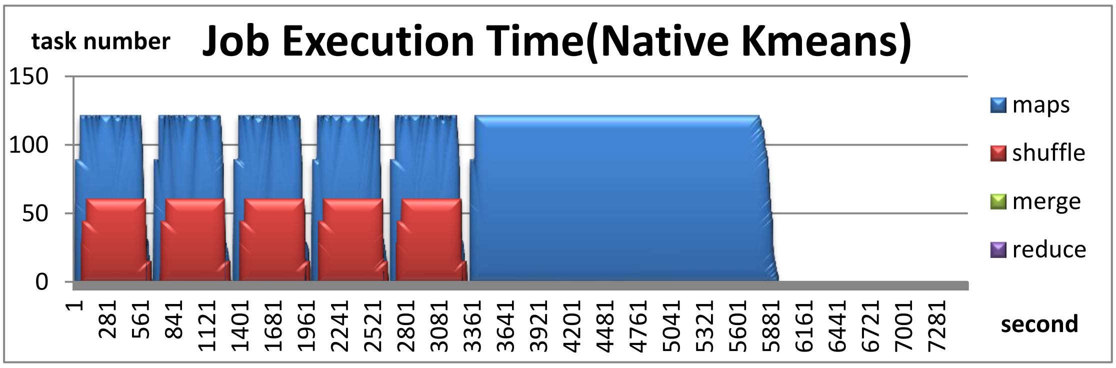
3.3.9 K-Means

- Job Details:

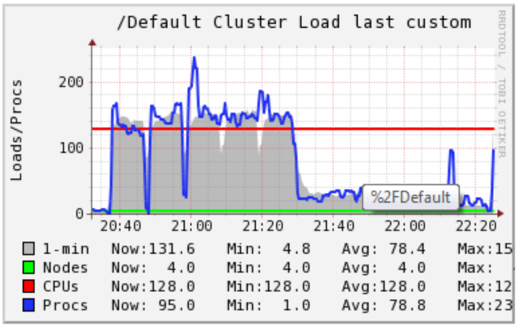
Name	Maps	Reducers
Cluster Iterator running iteration 1 …	1400	63
Cluster Iterator running	1400	63

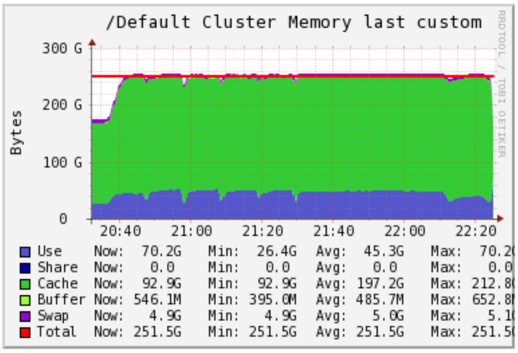
iteration 2 ...		
Cluster Iterator running iteration 3 ...	1400	63
Cluster Iterator running iteration 4 ...	1400	63
Cluster Iterator running iteration 5 ...	1400	63
Cluster Classification Driver running	1400	0

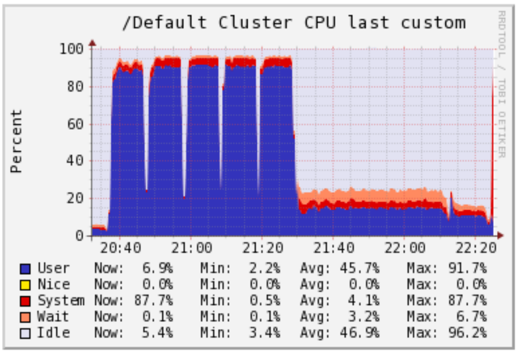


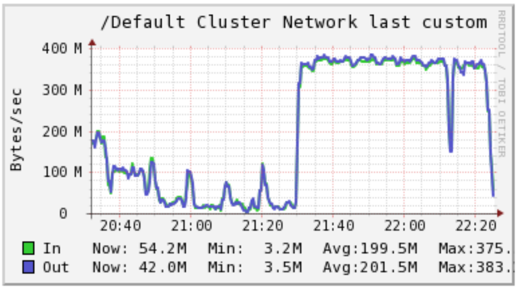


- Native-Task running state:





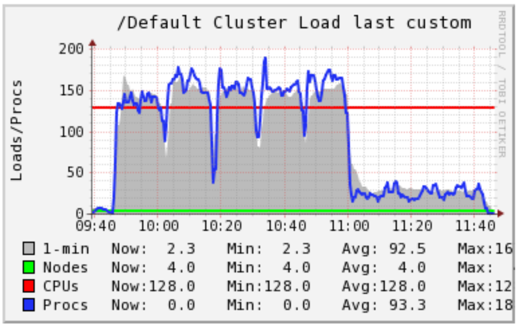


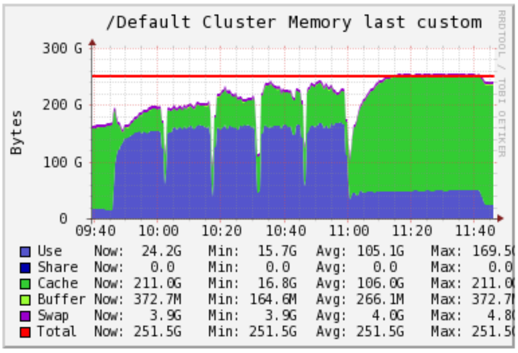


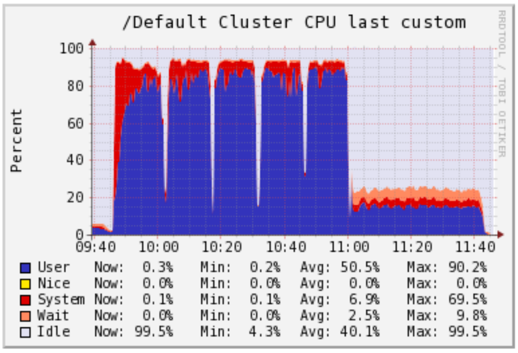
Start time: 20:38

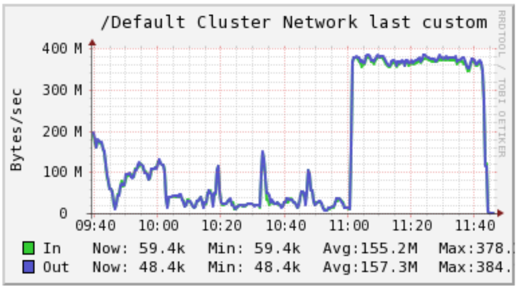
Finish time: 22:23

- Original running state:









Start time: 9:43
Finish time: 11:41

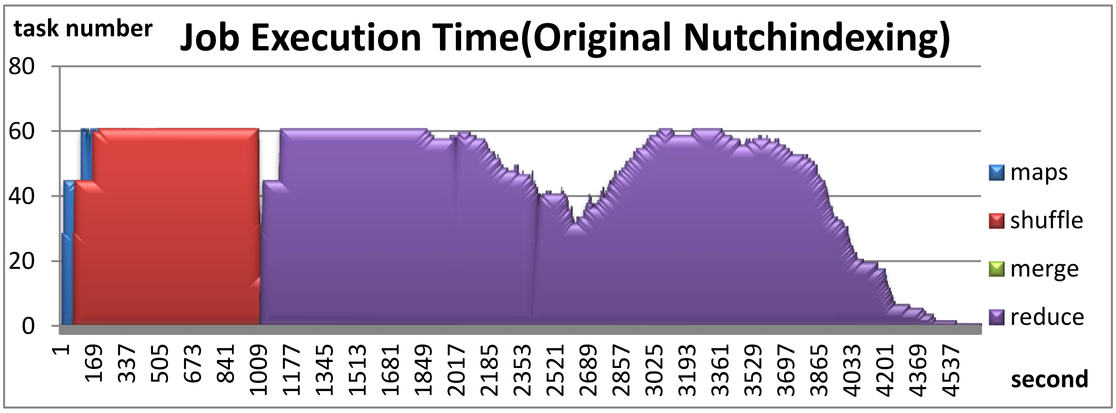
Analisys

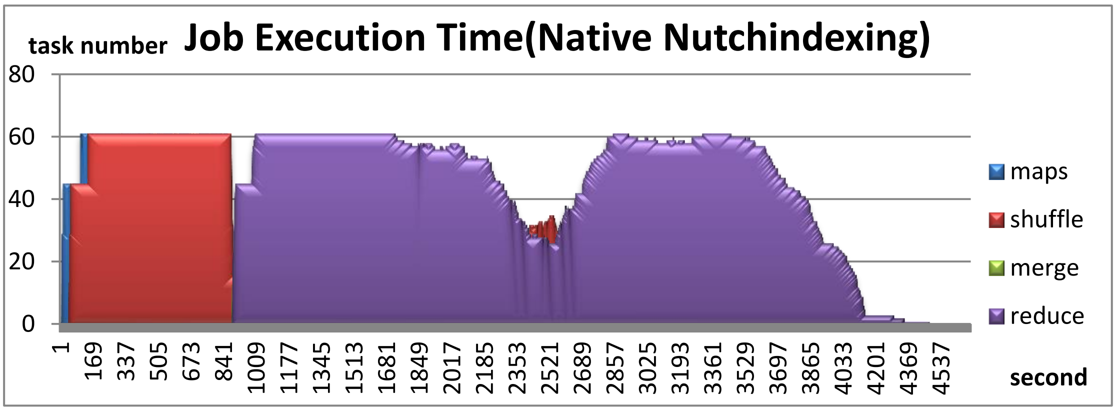
From the running state graph, we can see that the former 5 iteration is CPU-intensive and the last classification stage is IO-intensive. The two stages almost equally split the whole running time. So the performance improvement at map stage is evident.

3.3.10 Nutchindexing

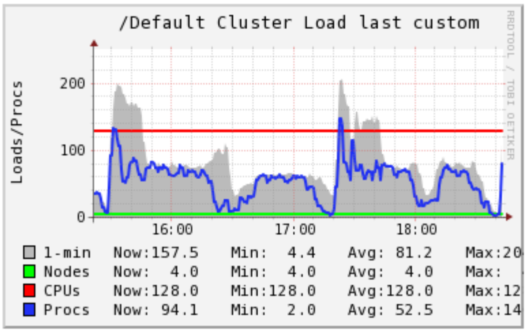
- Job Details:

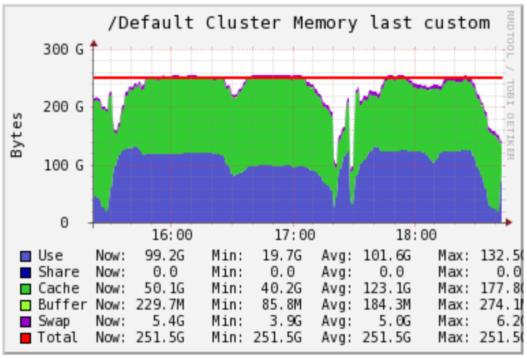
Name	Maps	Reducers
index-lucene /HiBench/Nutch/Input/indexes	1171	133

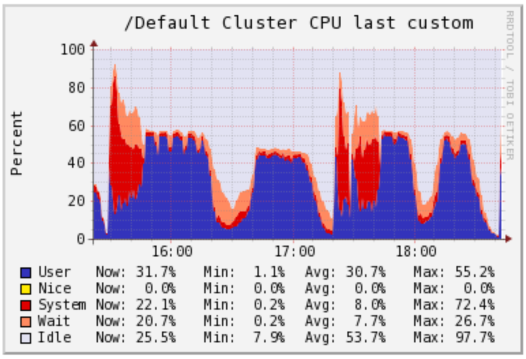


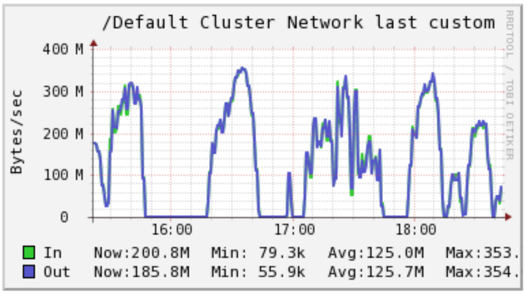


- Native-Task running state:





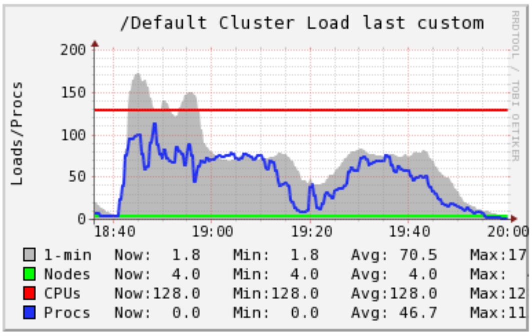


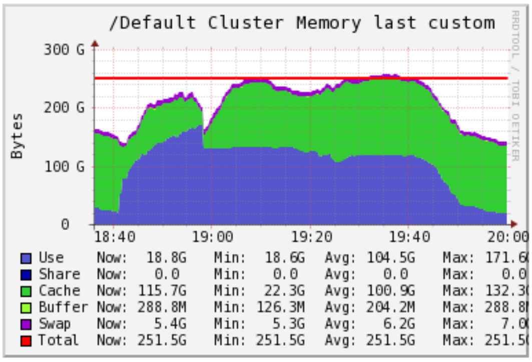


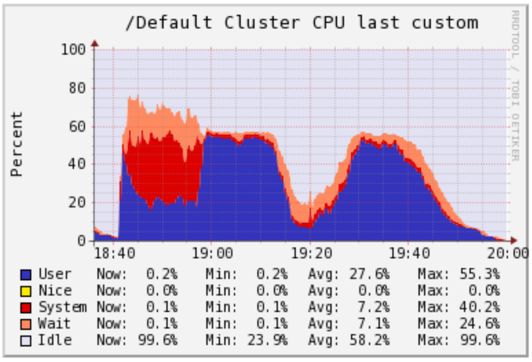
Start time: 17:26

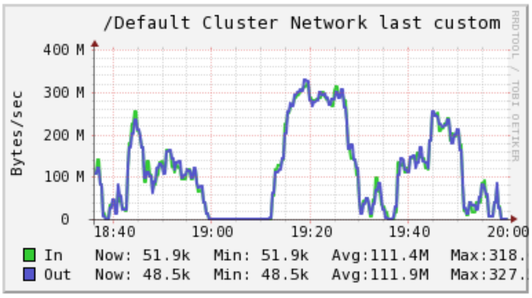
Finish time: 18:40

- Original running state:









Start time: 18:40

Finish time: 19:56

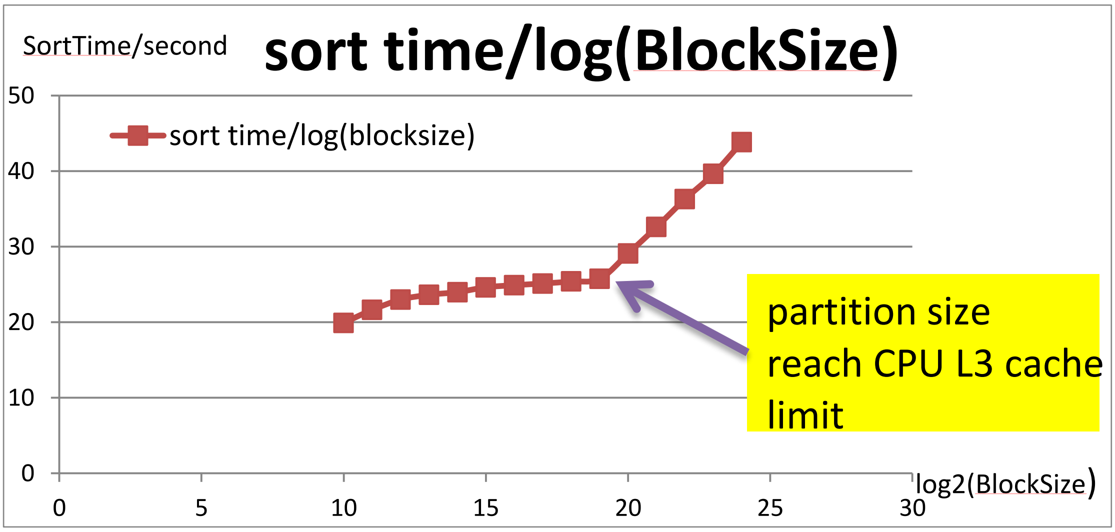
Analysis

Nutchindexing is CPU-intensive at map stage but the reduce stage take the majority of whole running time. So the performance improvement is not so huge.

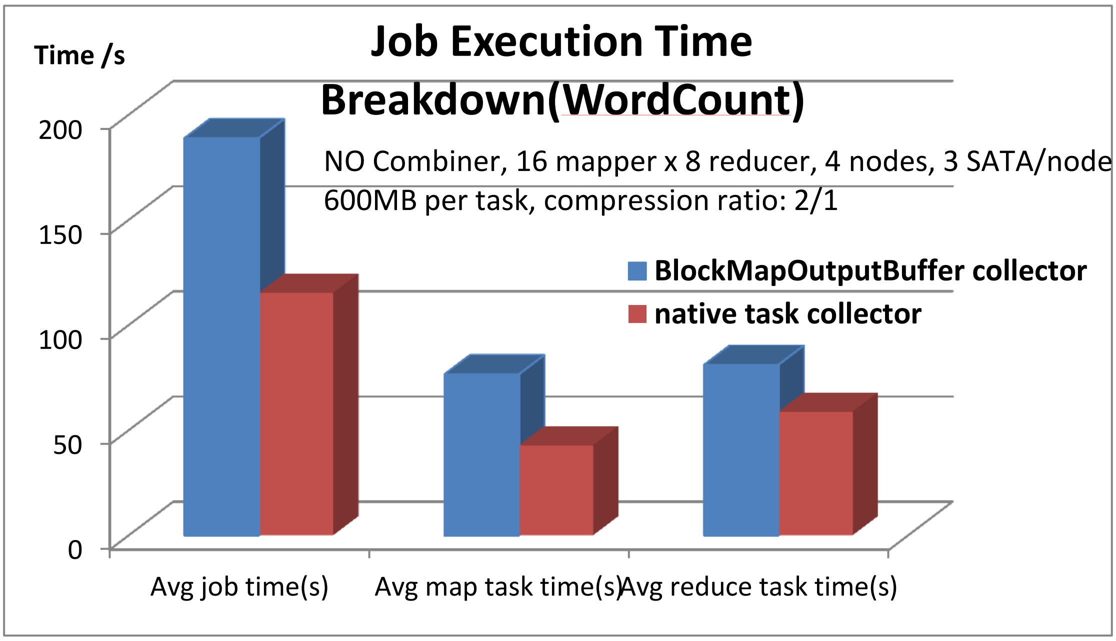
3.4 Other related results

3.4.1 Cache miss hurts Sorting performance

- Sorting time **increase rapidly** as cache miss rate increase
- We divide a large buffer into several memory unit.
- BlockSize is the size of memory unit we doing the sorting.



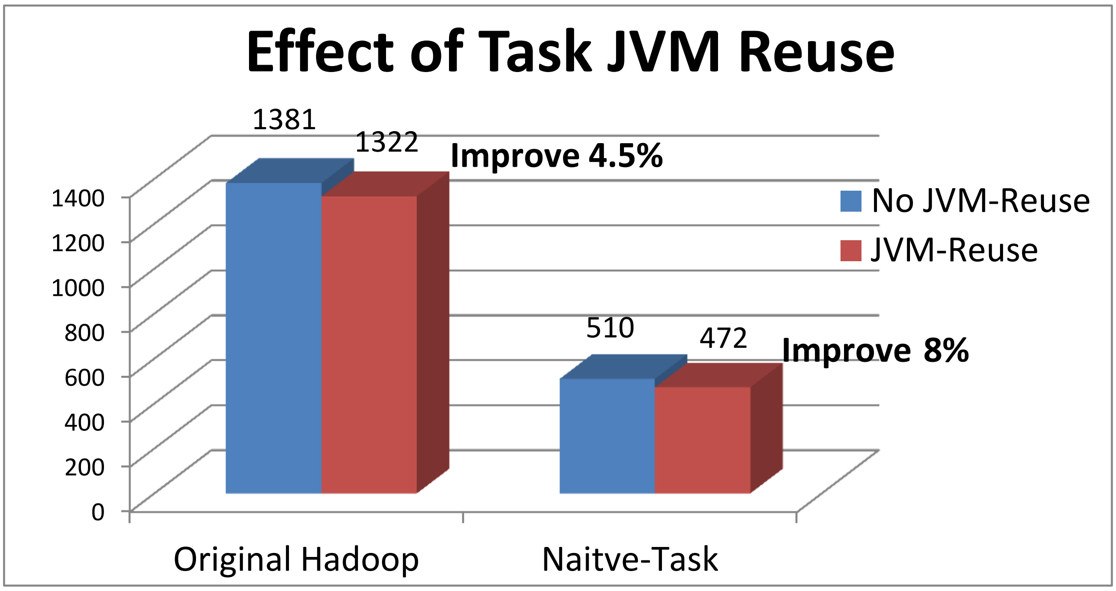
3.4.2 Compare with BlockMapOutputBuffer



- **70% faster** than BlockMapOutputBuffer collector.
- BlockMapOutputBuffer **supports ONLY BytesWritable**

3.4.3 Effect of JVM reuse

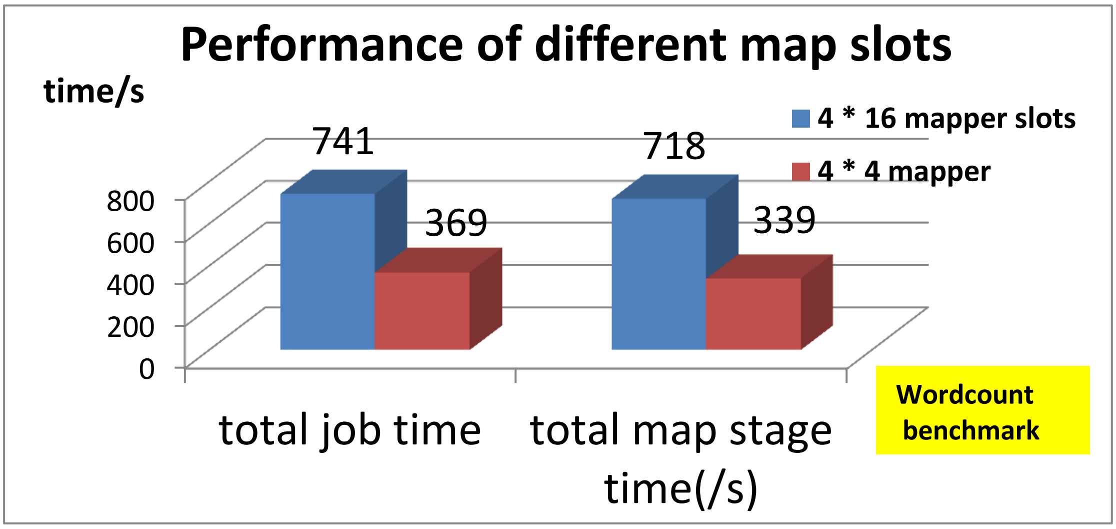
- 4.5% improvement for Original Hadoop, 8% improvement for Native-Task



4 nodes, 4 map slots per node.

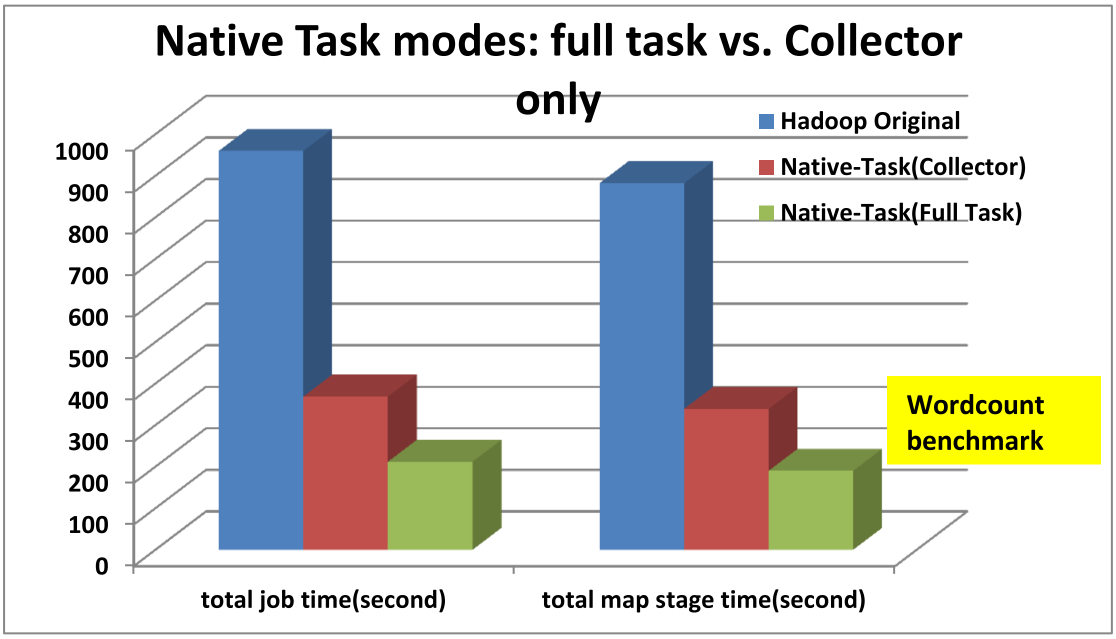
3.4.4 Hadoop don’t scale well when slots number increase

- 4 nodes(32 core per node), 16 map slots max, **CPU, memory, disk are NOT fully used.**
- **Performance drops unexpectedly when slots# increase**



3.4.5 Native-Task mode: full task optimization

- **2x faster further for Native-Task full time optimization,** compared with native collector



4. Conclusions