Contents

1. Shared Memory Sort

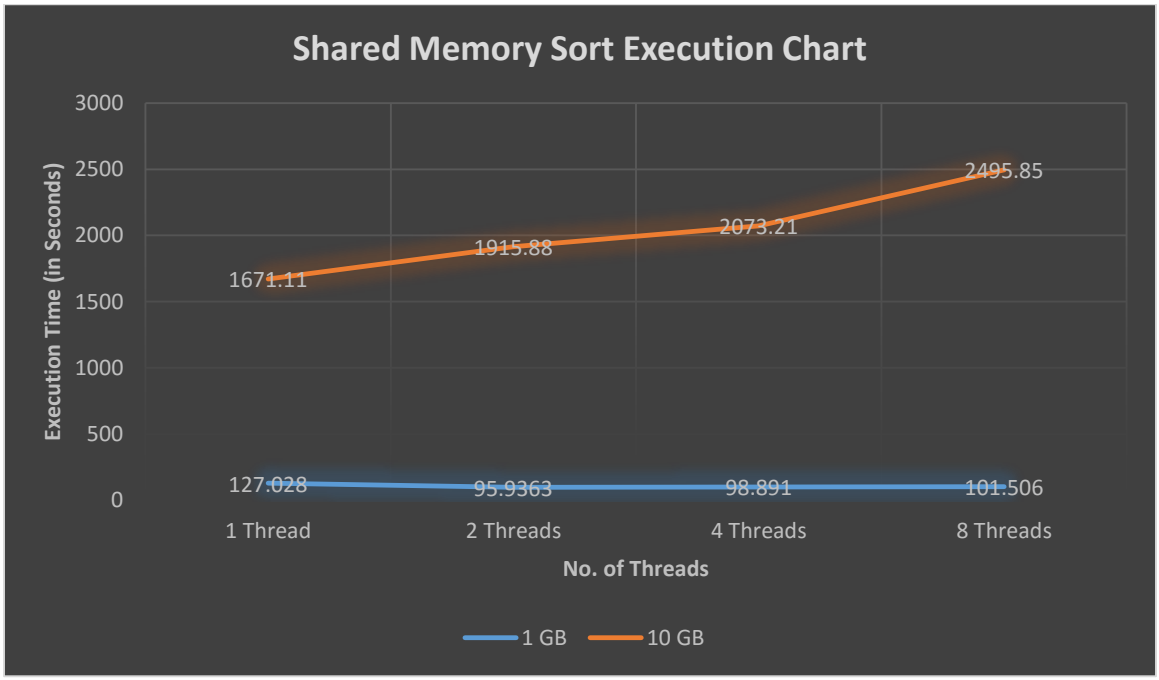
a. Problem

We were to implement the Shared Memory Sort application in any of the favorite language without MapReduce/Hadoop or Spark and measure its performance on 1 node on a c3.large instance. It should sort a file-resident dataset; it should be able to sort datasets that are larger than memory. We must make Shared Memory Sort multithreaded to take advantage of multiple cores. We were also supposed to measure the time to sort on the 10GB dataset by varying the number of threads from 1 to 8 (1, 2, 4 & 8), to find the best performance. Save the first 10 lines of output from the Shared-Memory Sort application, as well as the last 10 lines of output, for each dataset, in Sort-shared-memory-10GB.txt.

b. Performance

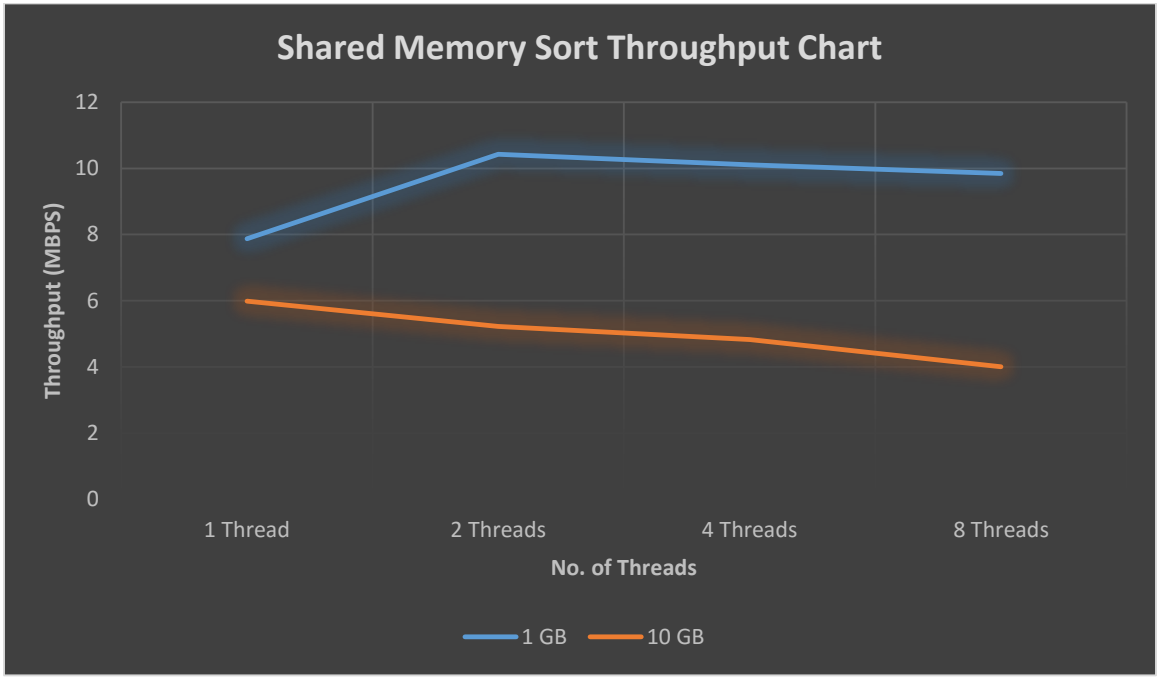
i. Shared Memory Sort Execution

	1 Thread	2 Threads	4 Threads	8 Threads
1 GB	127.028	95.9363	98.891	101.506
10 GB	1671.11	1915.88	2073.21	2495.85



ii. Shared Memory Sort Throughput

	1 Thread	2 Threads	4 Threads	8 Threads
1 GB	7.872280127	10.42358315	10.11214367	9.851634386
10 GB	5.984046532	5.219533582	4.82343805	4.006651041



(assuming 1,000,000 Bytes = 1MB)

Performance comparison analysis

- Shared Memory sort proves to be best for 1 GB dataset considering the Hadoop and Spark overheads
- Shared Memory performance decreases if we increase the no. of threads because of the hardware limitations (no. of cores available, memory available and no. of disk heads)
- Shared Memory performance decreases for 10 GB dataset because of the hardware limitations (memory available and no. of disk heads)

c. Methodology

- Shared Memory Sort application is implemented using C++ language.
- It uses External Merge Sort algorithm with *k – way merging* to achieve the desired results.

- The program accepts Input File Name (file to be sorted), Number of chunks / blocks and Number of Threads as input from user and validates them before proceeding ahead.
- A function *SplitFiles* accepts the Input File Name and splits it into the number of chunks / blocks provided by user.
- After the file splitting process is completed, External Merge Sort process is called for each file chunk in multithreaded environment. External Merge Sort process written in separate C++ program which accepts Input File Name and Output File Name as parameters to sort and store the results.
- Once the individual files chunks are sorted, a function to perform the k – way margining is called. This final merge pass generates the final completely sorted output file.

d. Runtime Environment Settings

- Operating System
  Ubuntu 14.04.3 LTS, Release: 14.04, Codename: trusty
- Operating System Kernel
  Linux ip-172-31-60-162 3.13.0-74-generic #118-Ubuntu SMP Thu Dec 17 22:52:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
- C++ compiler
  Using built-in specs.
  COLLECT_GCC=g++
  COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/4.8/lto-wrapper
  Target: x86_64-linux-gnu
  Configured with: ../src/configure -v --with-pkgversion='Ubuntu 4.8.4-2ubuntu1~14.04.1' --with-bugurl=file:///usr/share/doc/gcc-4.8/README.Bugs --enable-languages=c,c++,java,go,d,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-4.8 --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --with-gxx-include-dir=/usr/include/c++/4.8 --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --enable-gnu-unique-object --disable-libmudflap --enable-plugin --with-system-zlib --disable-browser-plugin --enable-java-awt=gtk --enable-gtk-cairo --with-java-home=/usr/lib/jvm/java-1.5.0-gcj-4.8-amd64/jre --enable-java-home --with-jvm-root-dir=/usr/lib/jvm/java-1.5.0-gcj-4.8-amd64 --with-jvm-jar-dir=/usr/lib/jvm-exports/java-1.5.0-gcj-4.8-amd64 --with-arch-directory=amd64 --with-ecj-jar=/usr/share/java/eclipse-ecj.jar --enable-objc-gc --enable-multiarch --disable-werror --with-arch-32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --with-tune=generic --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu

Thread model: posix

gcc version 4.8.4 (Ubuntu 4.8.4-2ubuntu1~14.04.1)

- Make

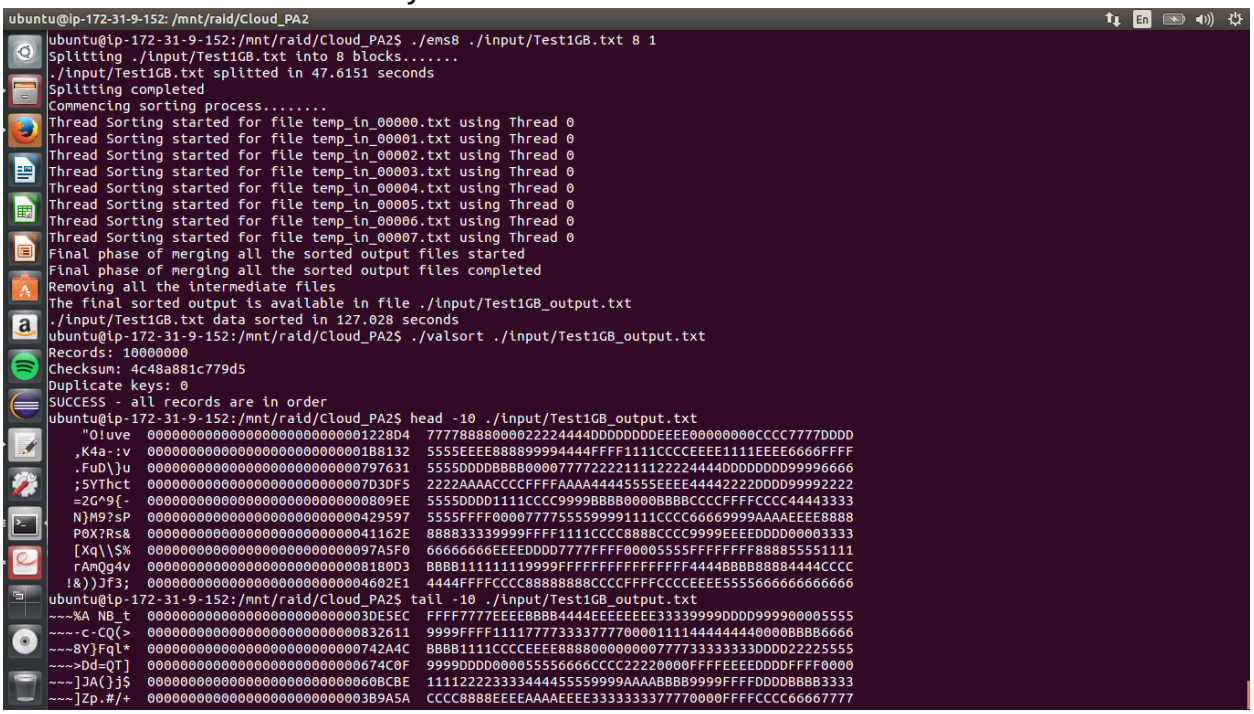
  GNU Make 3.81

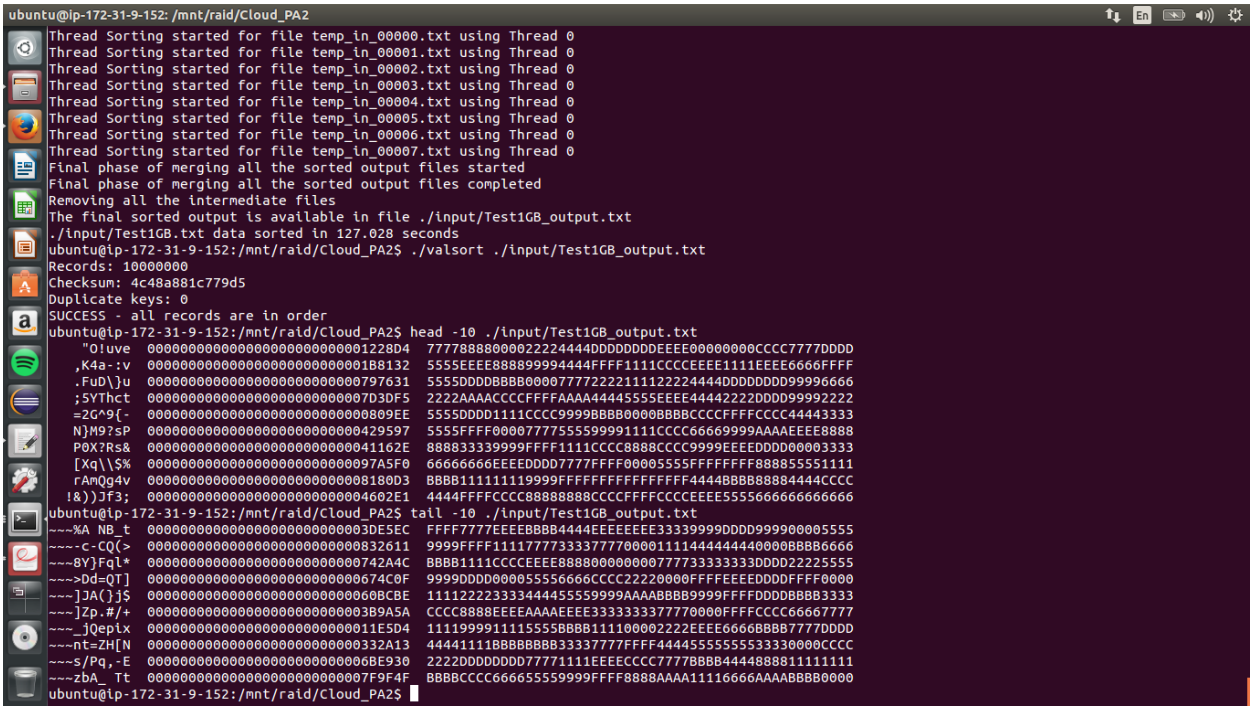
e. Setup & Execution Process

- Copy the *"ExternalMergeSort.cpp"*, *"MergeSort.cpp"* and *"SharedMemorySort.h"* files
- Compile the source files using *makefile* mechanism
- Execute "*ems"* binary with Input File Name to be sorted, Number of Chunks / Blocks and Number of Threads to start the sorting process

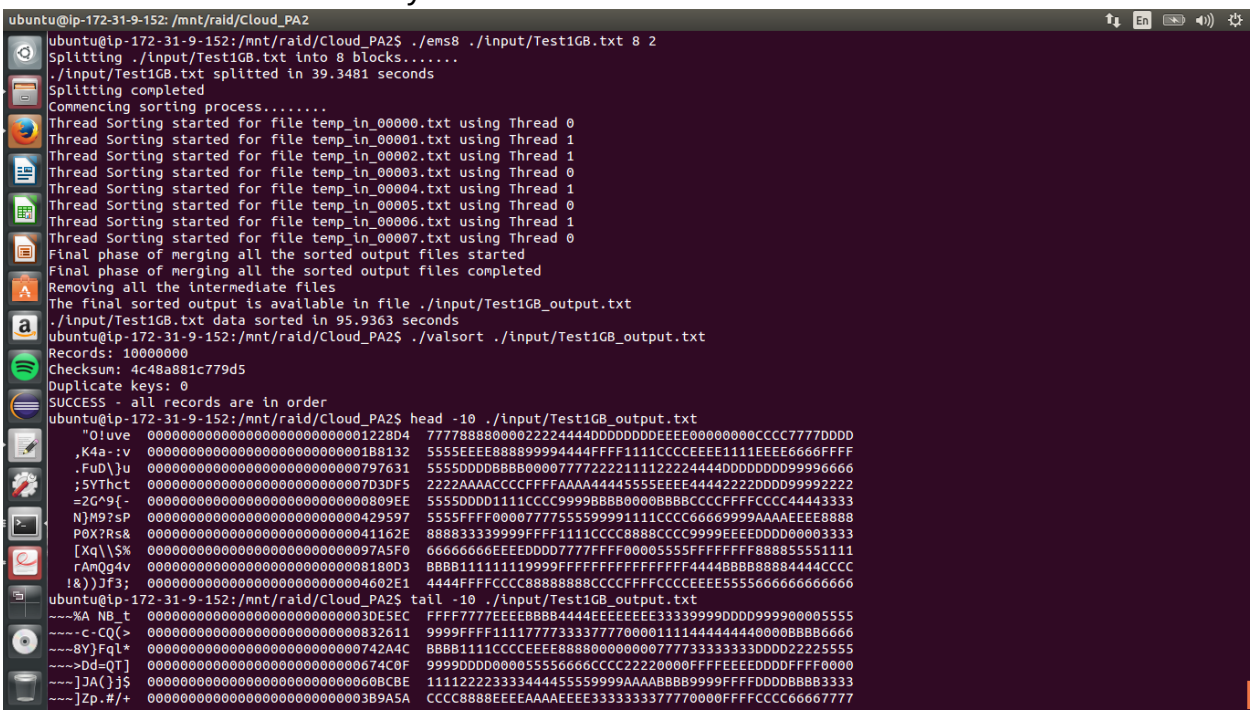
f. Output Screenshots

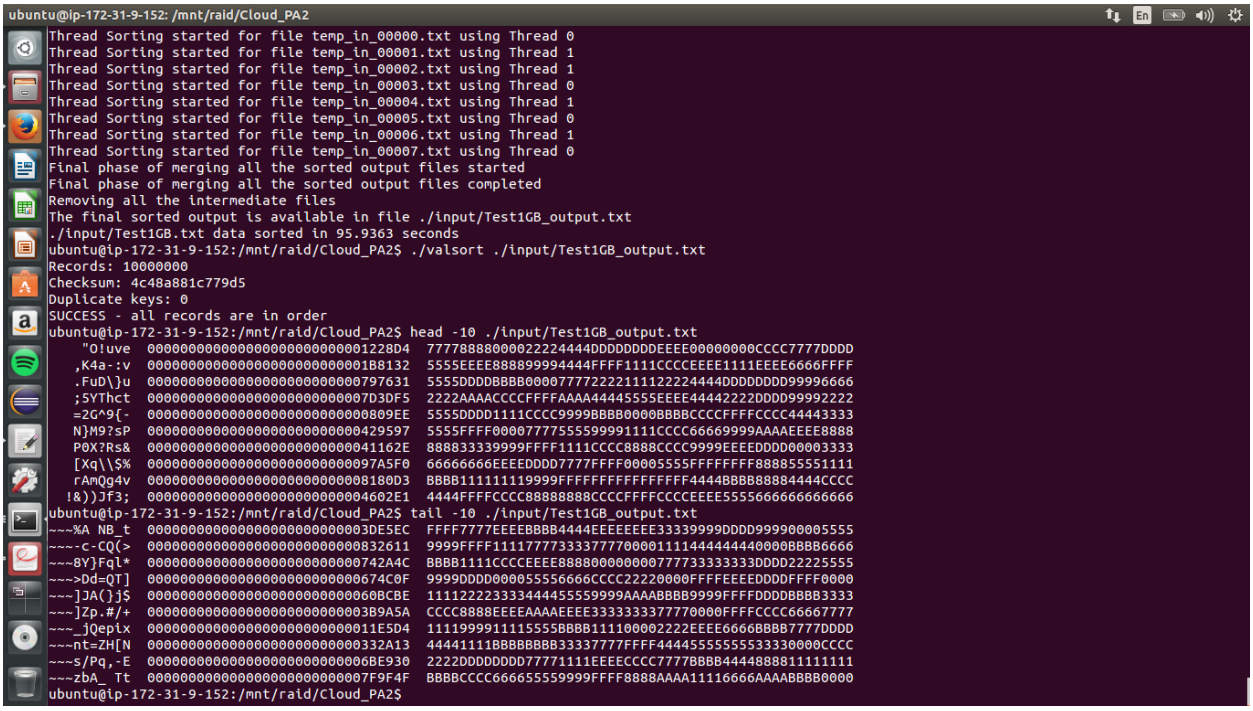
i. Shared Memory Sort 1 GB 1 Thread



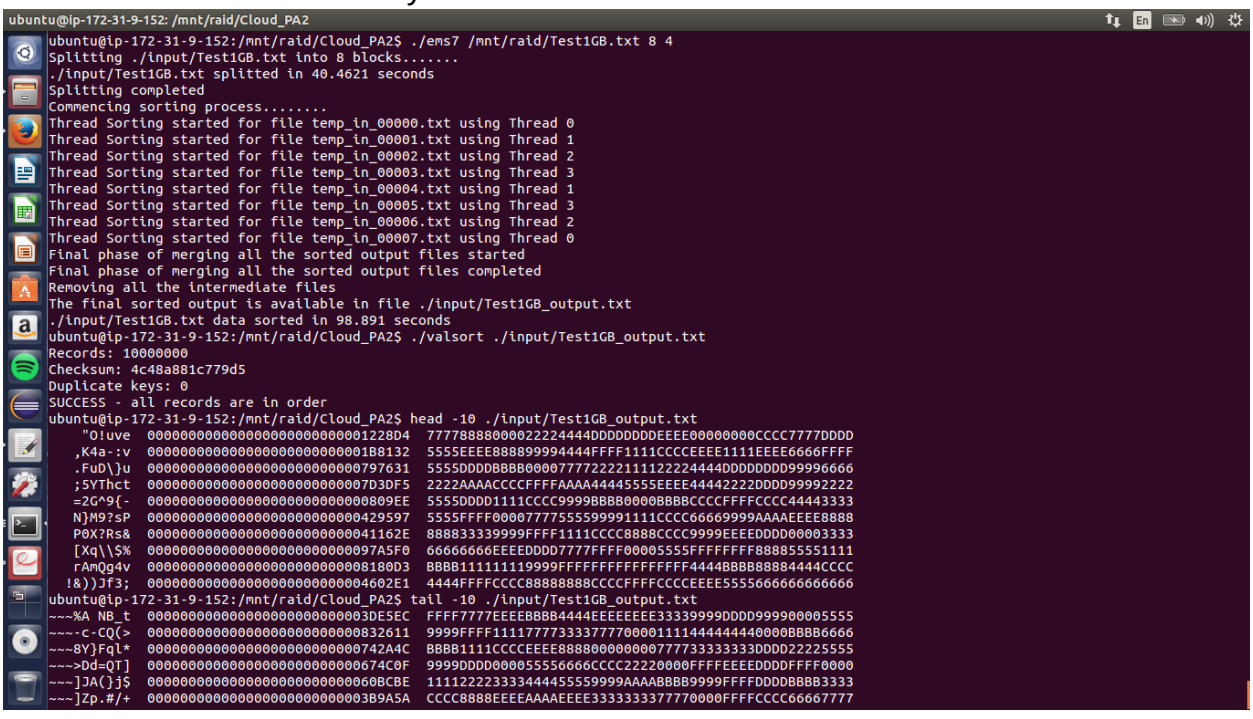


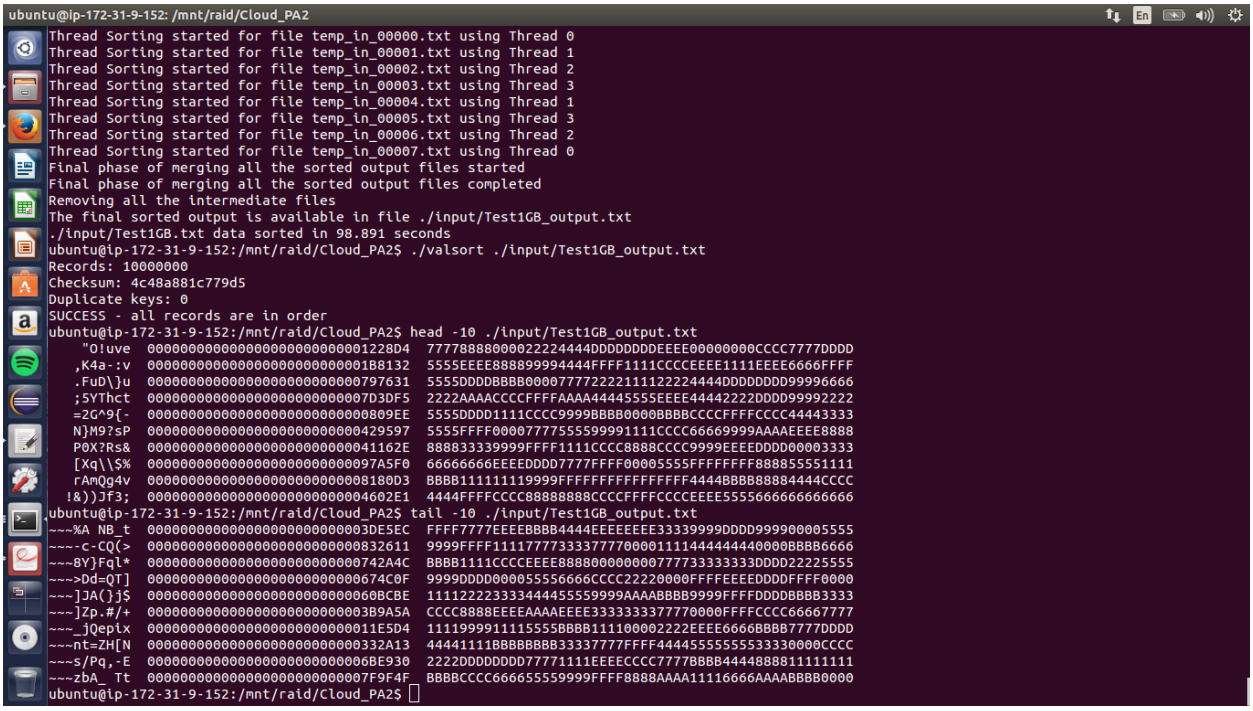
ii. Shared Memory Sort 1 GB 2 Threads



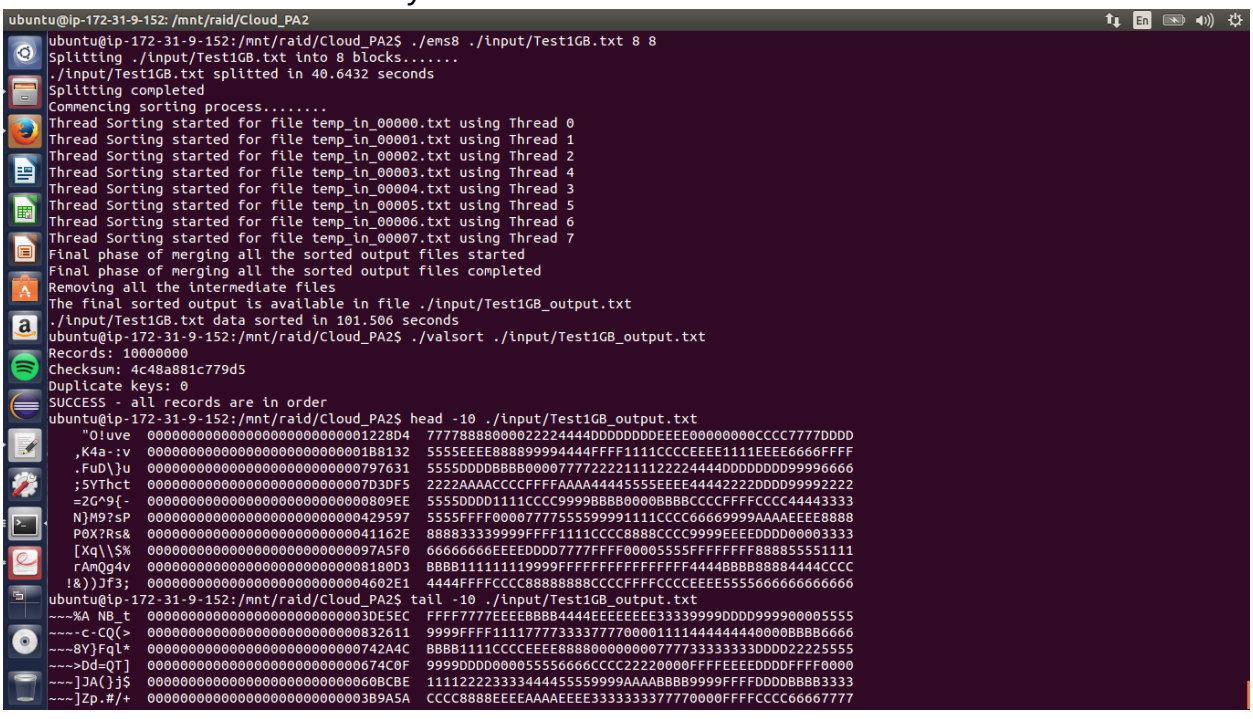


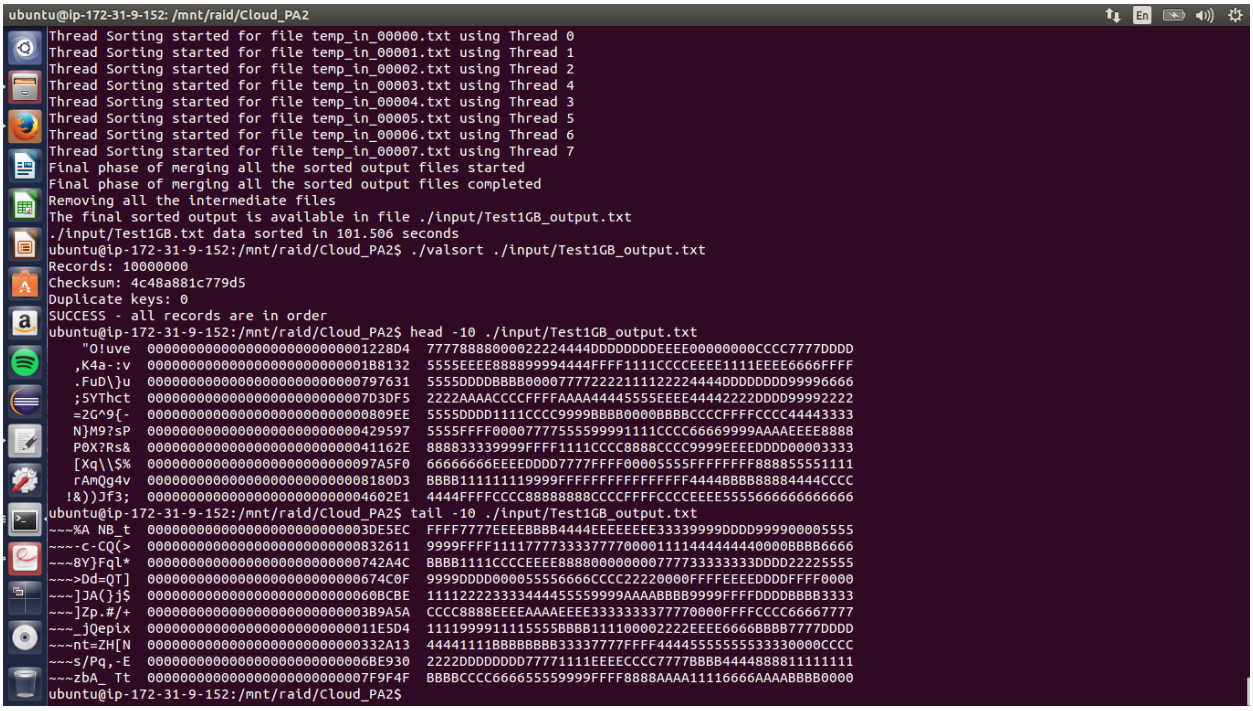
iii. Shared Memory Sort 1 GB 4 Threads



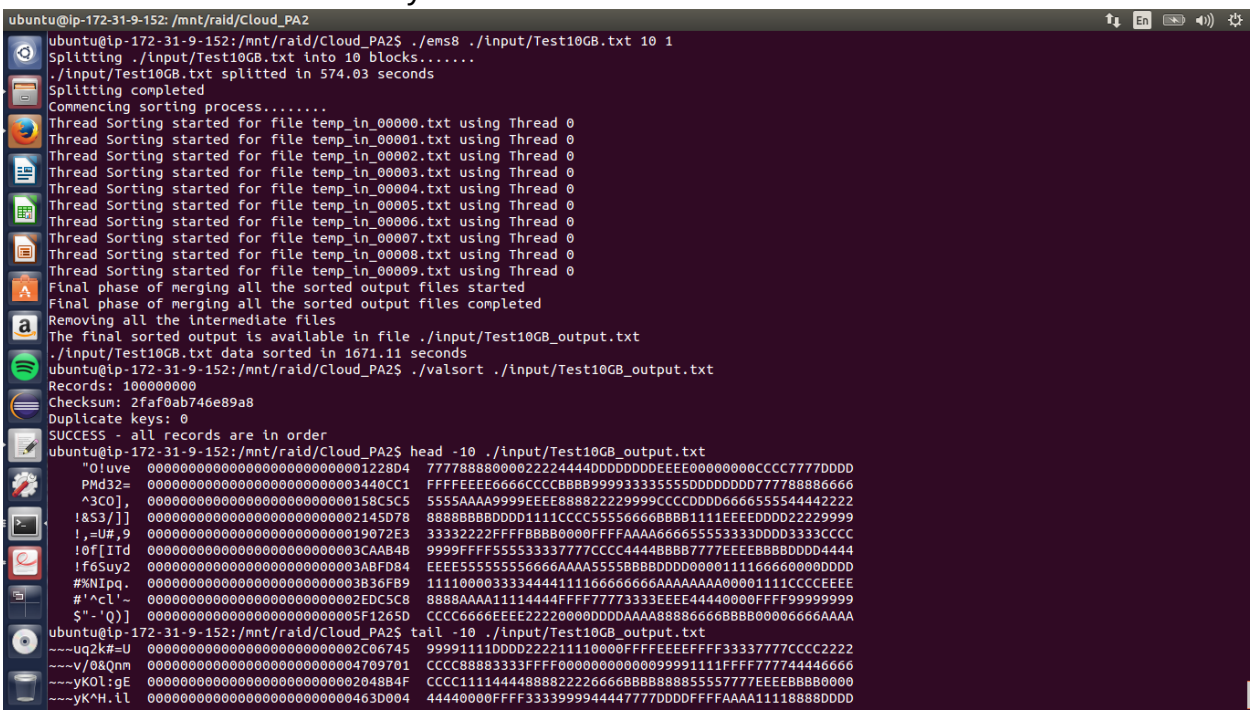


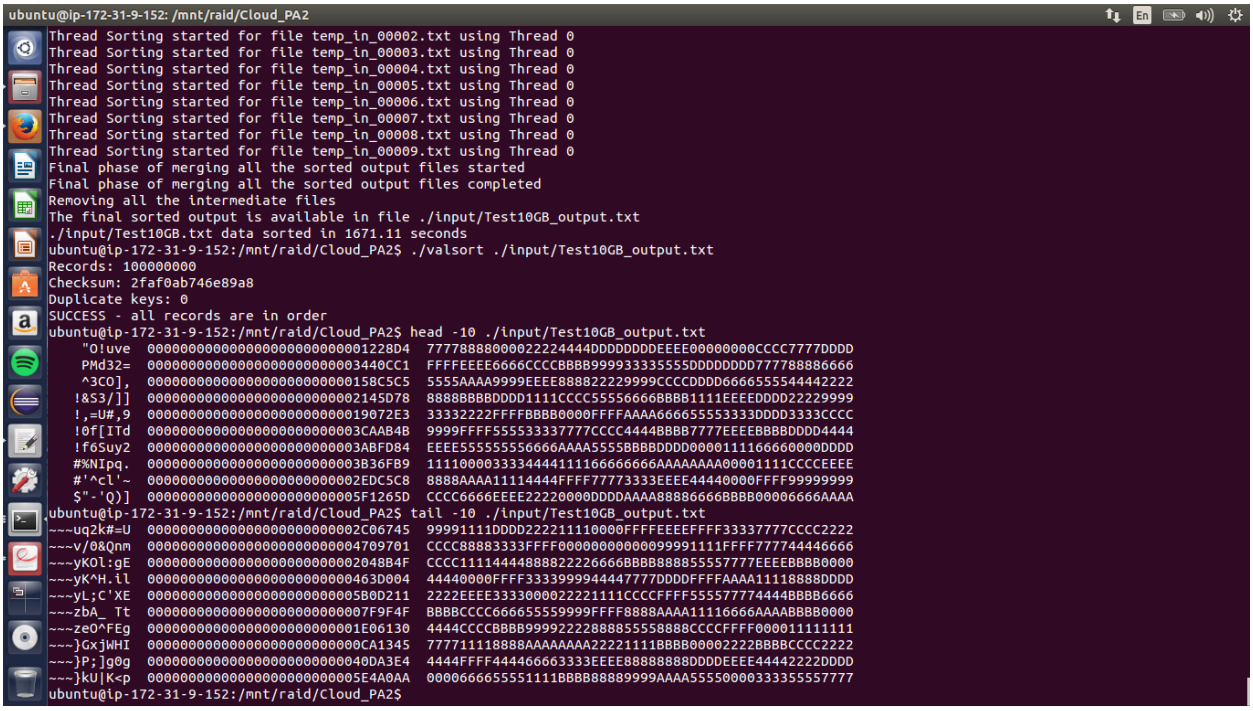
iv. Shared Memory Sort 1 GB 8 Threads



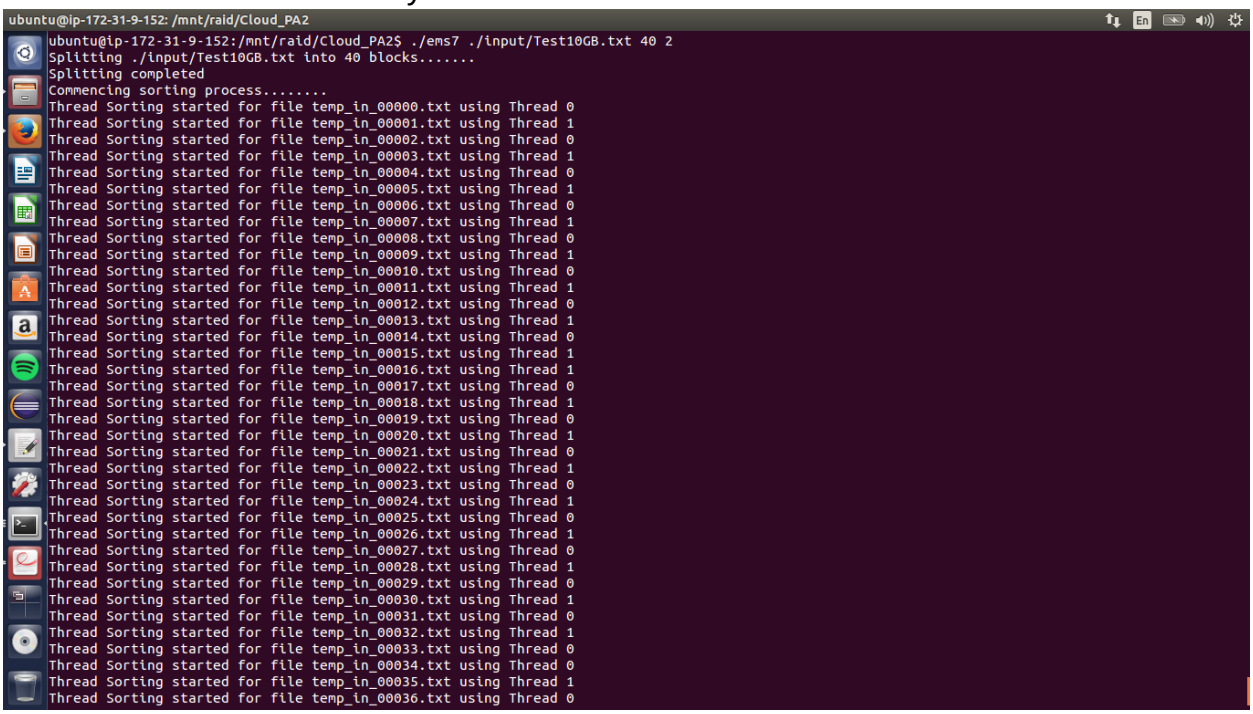


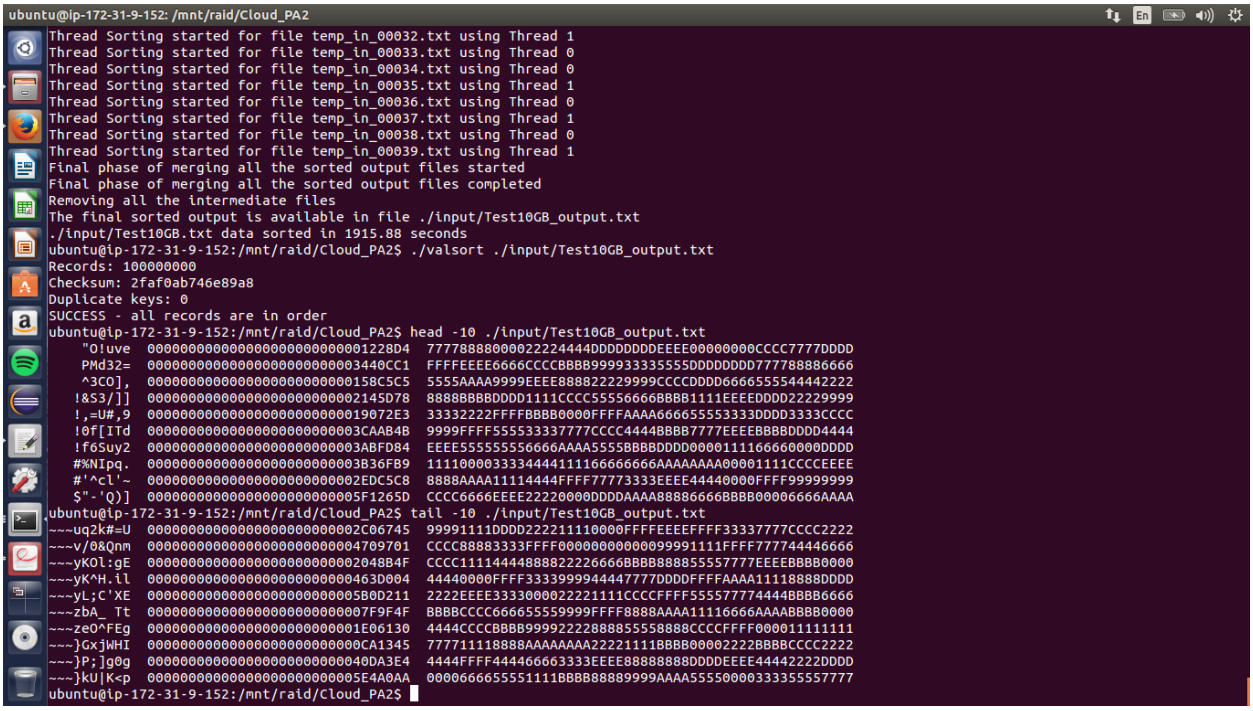
v. Shared Memory Sort 10 GB 1 Thread



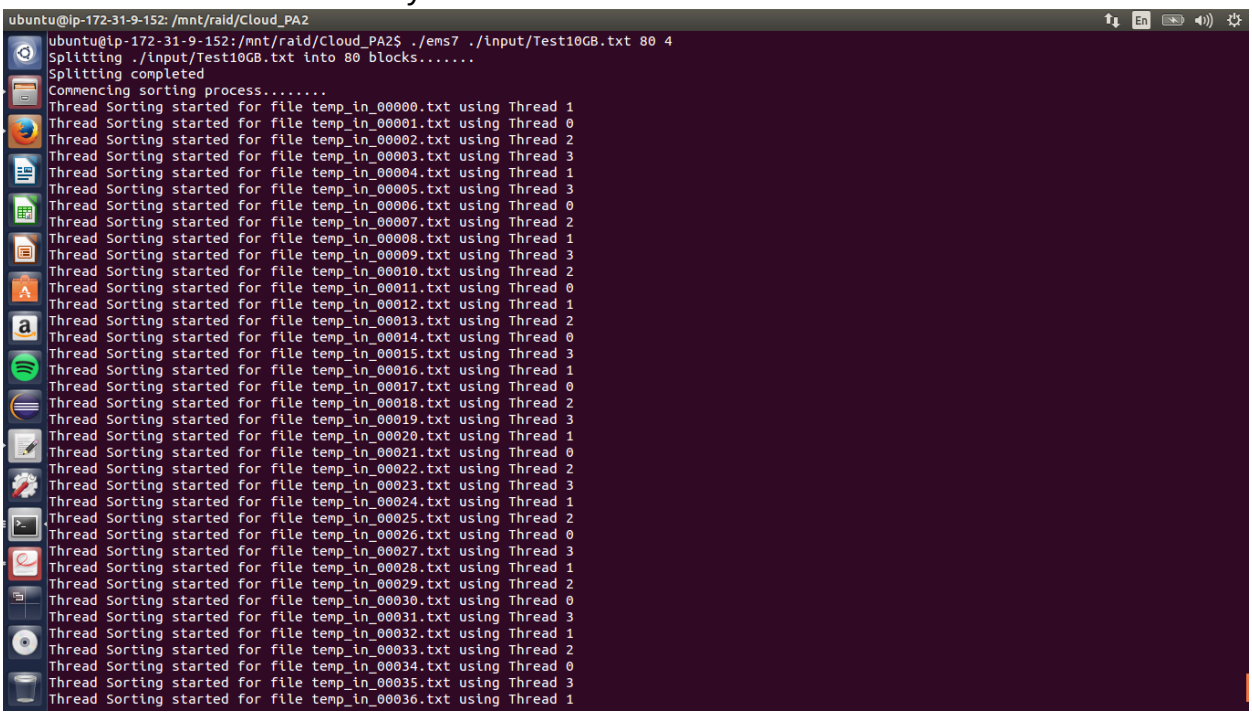


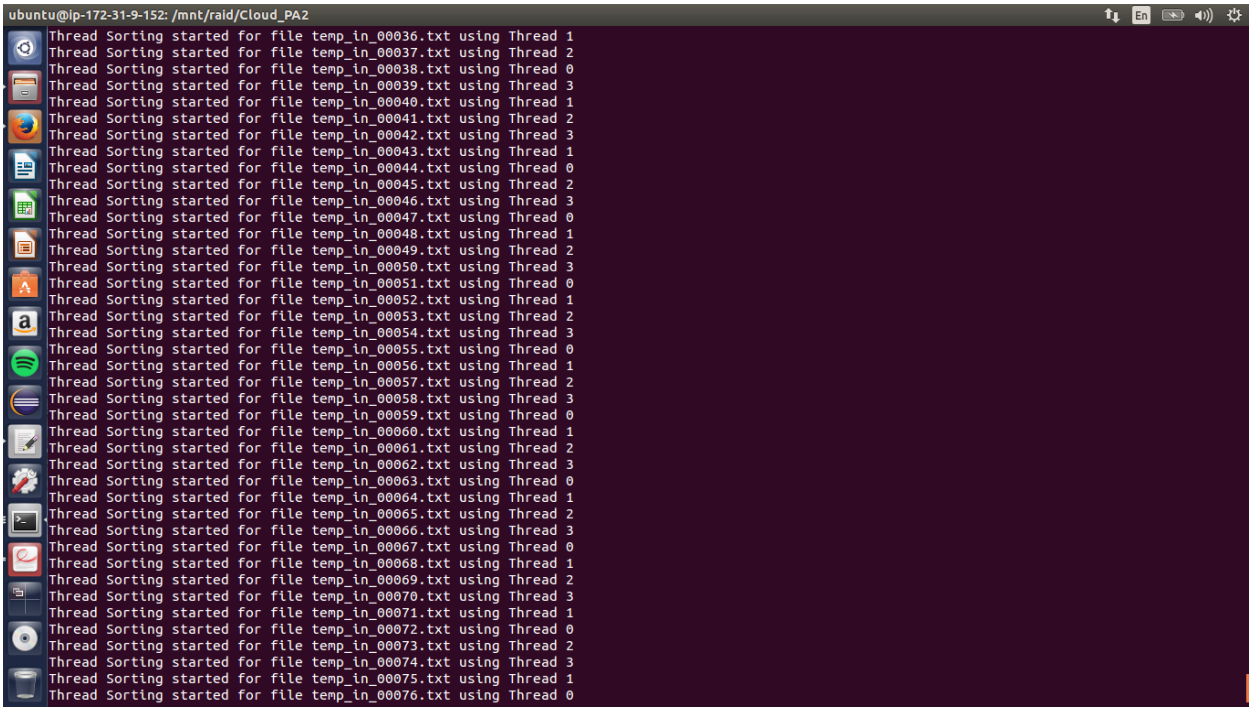
vi. Shared Memory Sort 10 GB 2 Threads

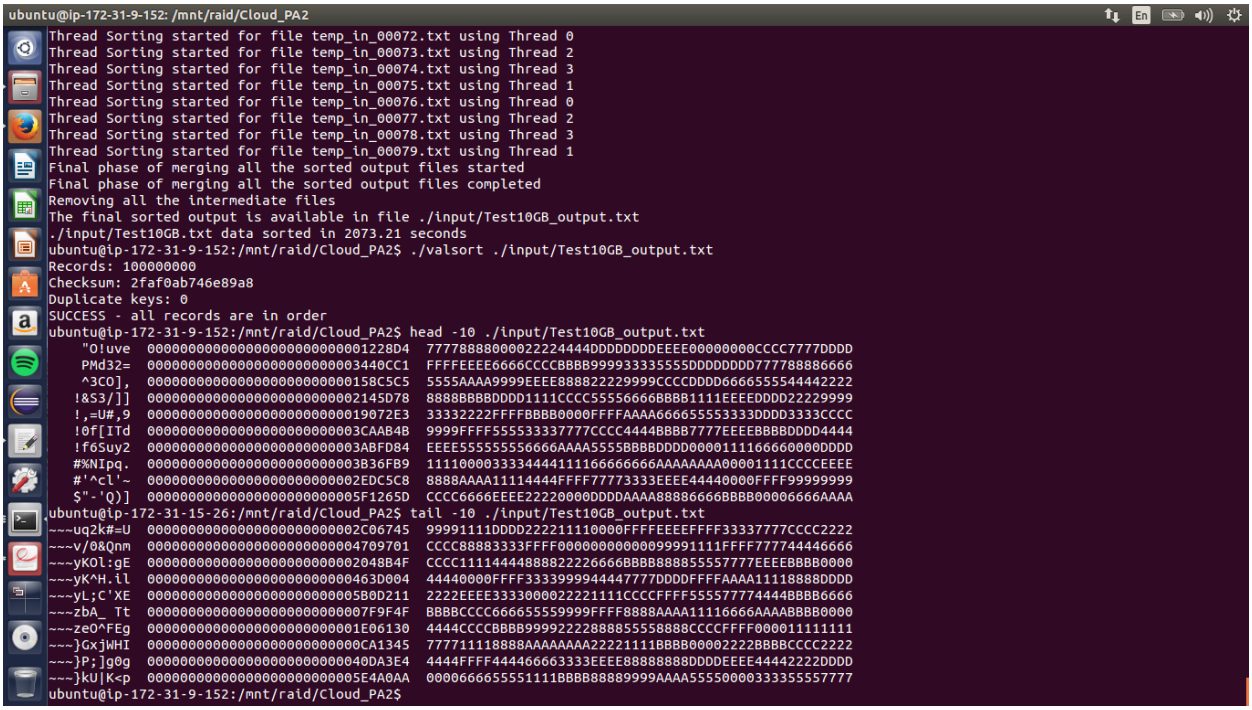




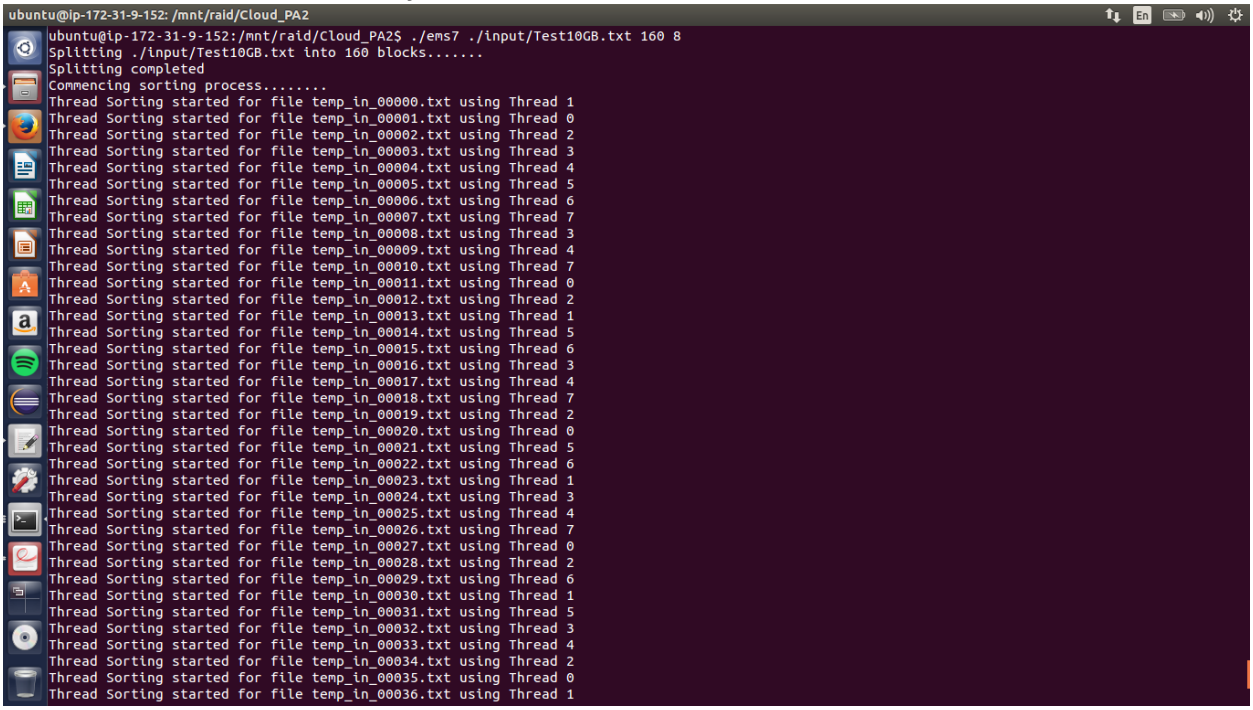
vii. Shared Memory Sort 10 GB 4 Threads

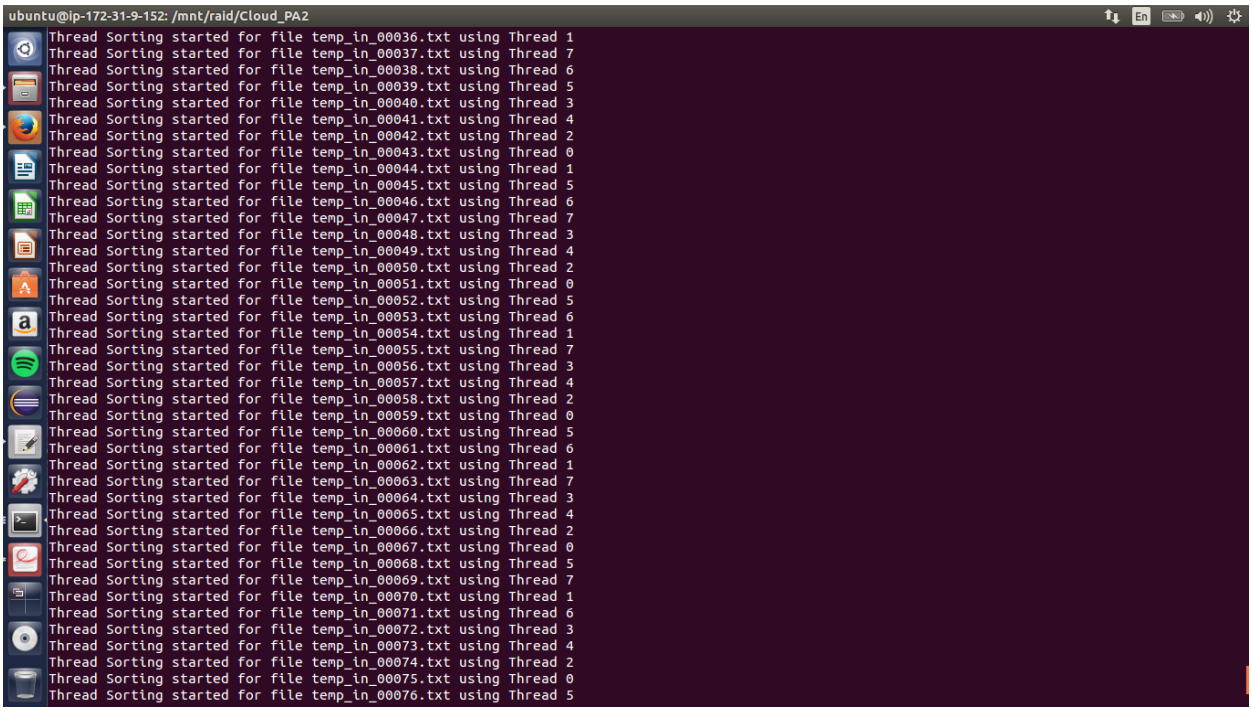


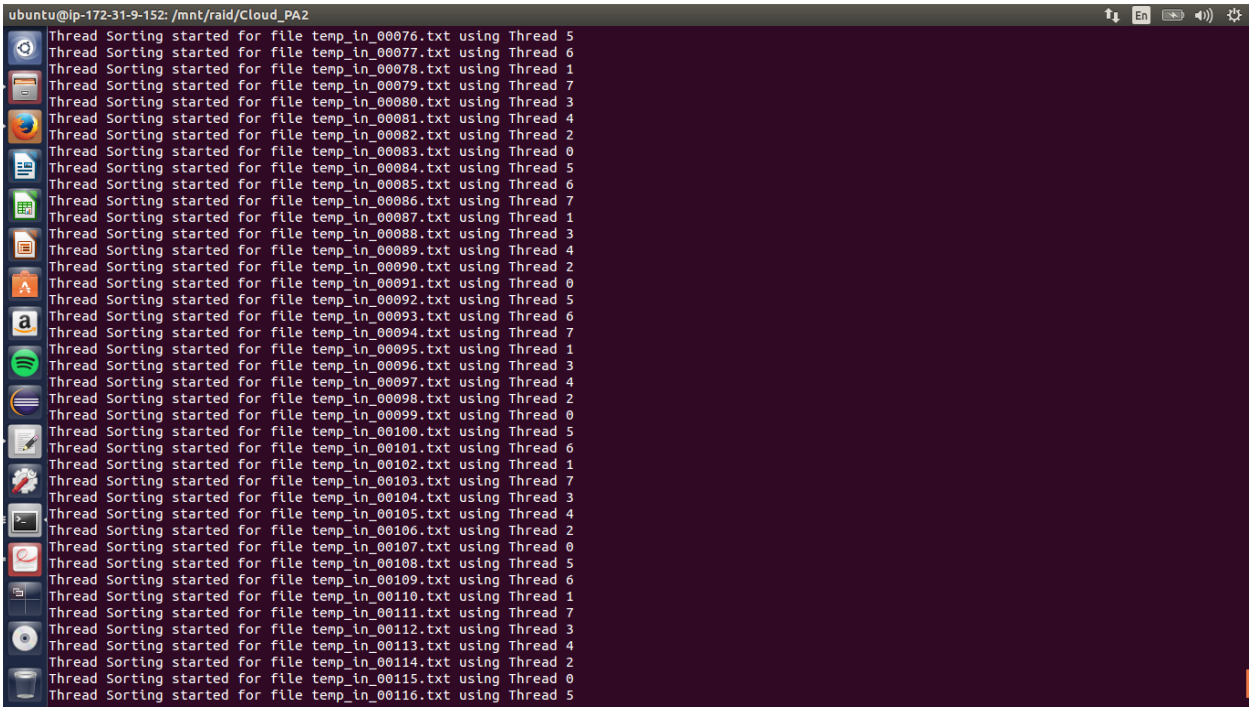

ubuntu@ip-172-31-9-152: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00036.txt using Thread 1
Thread Sorting started for file temp_in_00037.txt using Thread 2
Thread Sorting started for file temp_in_00038.txt using Thread 0
Thread Sorting started for file temp_in_00039.txt using Thread 3
Thread Sorting started for file temp_in_00040.txt using Thread 1
Thread Sorting started for file temp_in_00041.txt using Thread 2
Thread Sorting started for file temp_in_00042.txt using Thread 3
Thread Sorting started for file temp_in_00043.txt using Thread 1
Thread Sorting started for file temp_in_00044.txt using Thread 0
Thread Sorting started for file temp_in_00045.txt using Thread 2
Thread Sorting started for file temp_in_00046.txt using Thread 3
Thread Sorting started for file temp_in_00047.txt using Thread 0
Thread Sorting started for file temp_in_00048.txt using Thread 1
Thread Sorting started for file temp_in_00049.txt using Thread 2
Thread Sorting started for file temp_in_00050.txt using Thread 3
Thread Sorting started for file temp_in_00051.txt using Thread 0
Thread Sorting started for file temp_in_00052.txt using Thread 1
Thread Sorting started for file temp_in_00053.txt using Thread 2
Thread Sorting started for file temp_in_00054.txt using Thread 3
Thread Sorting started for file temp_in_00055.txt using Thread 0
Thread Sorting started for file temp_in_00056.txt using Thread 1
Thread Sorting started for file temp_in_00057.txt using Thread 2
Thread Sorting started for file temp_in_00058.txt using Thread 3
Thread Sorting started for file temp_in_00059.txt using Thread 0
Thread Sorting started for file temp_in_00060.txt using Thread 1
Thread Sorting started for file temp_in_00061.txt using Thread 2
Thread Sorting started for file temp_in_00062.txt using Thread 3
Thread Sorting started for file temp_in_00063.txt using Thread 0
Thread Sorting started for file temp_in_00064.txt using Thread 1
Thread Sorting started for file temp_in_00065.txt using Thread 2
Thread Sorting started for file temp_in_00066.txt using Thread 3
Thread Sorting started for file temp_in_00067.txt using Thread 0
Thread Sorting started for file temp_in_00068.txt using Thread 1
Thread Sorting started for file temp_in_00069.txt using Thread 2
Thread Sorting started for file temp_in_00070.txt using Thread 3
Thread Sorting started for file temp_in_00071.txt using Thread 1
Thread Sorting started for file temp_in_00072.txt using Thread 0
Thread Sorting started for file temp_in_00073.txt using Thread 2
Thread Sorting started for file temp_in_00074.txt using Thread 3
Thread Sorting started for file temp_in_00075.txt using Thread 1
Thread Sorting started for file temp_in_00076.txt using Thread 0

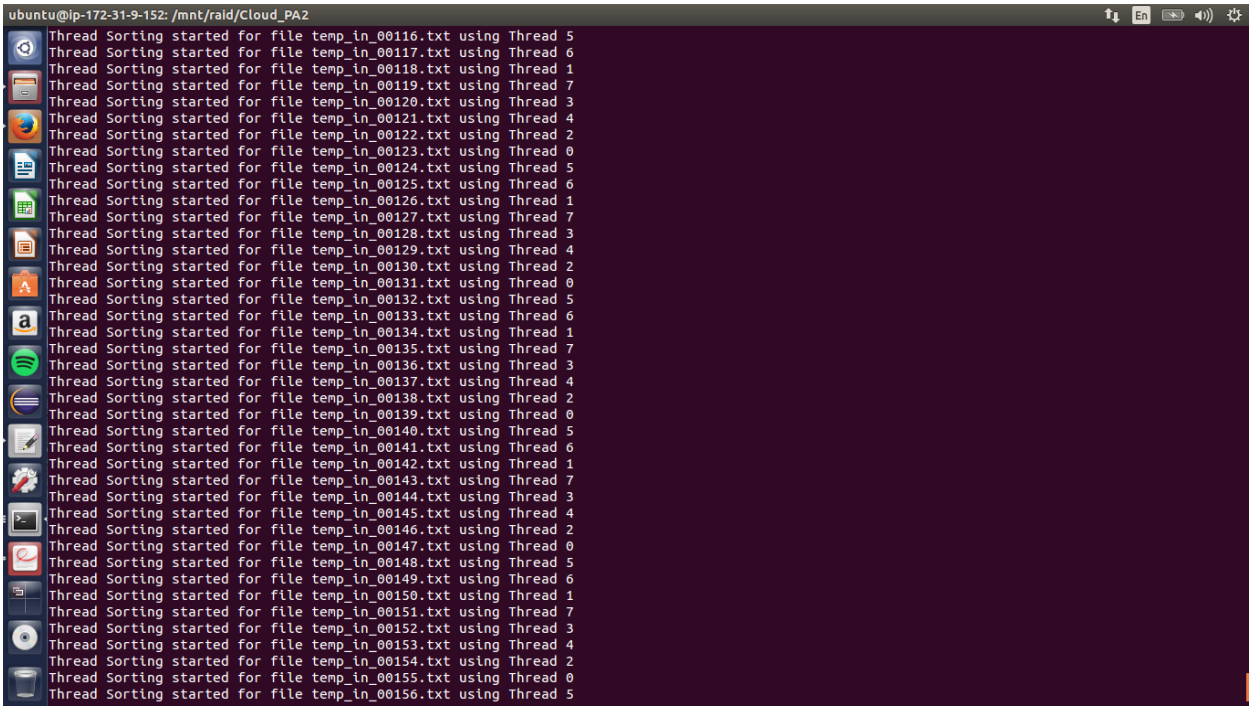

ubuntu@ip-172-31-9-152: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00072.txt using Thread 0
Thread Sorting started for file temp_in_00073.txt using Thread 2
Thread Sorting started for file temp_in_00074.txt using Thread 3
Thread Sorting started for file temp_in_00075.txt using Thread 1
Thread Sorting started for file temp_in_00076.txt using Thread 0
Thread Sorting started for file temp_in_00077.txt using Thread 2
Thread Sorting started for file temp_in_00078.txt using Thread 3
Thread Sorting started for file temp_in_00079.txt using Thread 1
Final phase of merging all the sorted output files started
Final phase of merging all the sorted output files completed
Removing all the intermediate files
The final sorted output is available in file ./input/Test10GB_output.txt
./input/Test10GB.txt data sorted in 2073.21 seconds
ubuntu@ip-172-31-9-152:/mnt/raid/Cloud_PA2$ ./valsort ./input/Test10GB_output.txt
Records: 100000000
Checksum: 2faf0ab746e89a8
Duplicate keys: 0
SUCCESS - all records are in order
ubuntu@ip-172-31-9-152:/mnt/raid/Cloud_PA2$ head -10 ./input/Test10GB_output.txt
"O!uve 00000000000000000000000001228D4 77778888000022224444DDDDDDDDEEEE00000000CCCC7777DDDD
PMd32= 00000000000000000000000003440CC1 FFFFEEEE6666CCCCBBBB9999333335555DDDDDDDD7777888866666
^3CO], 0000000000000000000000000158C5C5 5555AAAA9999EEEE888822229999CCCCDDDD6666555544442222
!&S3/]] 000000000000000000000000002145D78 8888BBBBDDDD1111CCCC55556666BBBB1111EEEEDDDD22229999
!,=U#,9 00000000000000000000000019072E3 33332222FFFFBBBB0000FFFFAAAA6666555533333DDDD3333CCCC
!0f[ITd 0000000000000000000000003CAAB4B 9999FFFF5555333377777CCCC4444BBBB7777EEEEBBBBDDDD4444
!f6Suy2 0000000000000000000000003ABFD84 EEEE555555556666AAAA5555BBBBDDDD00001111666600000DDDD
#%NIpq. 00000000000000000000000003B36FB9 11110000333344441111666666666AAAAAAAA00001111CCCCEEEE
#'^cl'~ 000000000000000000000002EDC5C8 8888AAAA11114444FFFF77773333EEEE44440000FFFF99999999
$"-'Q)] 00000000000000000000000005F1265D CCCC6666EEEEE22220000DDDDAAAA88886666BBBB00006666AAAA
ubuntu@ip-172-31-15-26:/mnt/raid/Cloud_PA2$ tail -10 ./input/Test10GB_output.txt
~~~uq2k#=U 00000000000000000000000002C06745 99991111DDDD222211110000FFFFEEEEFFFF33337777CCCC2222
~~~v/0&Qnm 00000000000000000000000004709701 CCCC88883333FFFF000000000000999911111FFFF777744446666
~~~yKOl:gE 00000000000000000000000002048B4F CCCC1111444488882222666666BBBB888855557777EEEEBBBB0000
~~~yK^H.il 00000000000000000000000000463D004 44440000FFFF333399994444777700DDDDFFFFAAAA11118888DDDD
~~~yL;C'XE 000000000000000000000000005B0D211 2222EEEE333300002222111100CCCCFFFF5555777744448BBBB6666
~~~zbA_ Tt 00000000000000000000000000007F9F4F BBBBCCCC6666555599999FFFF8888AAAA11116666AAAABBBB0000
~~~zeO^FEg 000000000000000000000000001E06130 4444CCCCBBBB999922228888555588888CCCCFFFF000011111111
~~~}GxjWHI 00000000000000000000000000CA1345 77771111888888AAAAAAAA22221111BBBB00002222BBBBCCCC2222
~~~}P;]g0g 00000000000000000000000040DA3E4 4444FFFF444466663333EEEE88888888DDDDEEEE44442222DDDD
~~~}kU|K<p 000000000000000000000000005E4A0AA 0000666655551111BBBB88889999AAAA55550000333355557777
ubuntu@ip-172-31-9-152:/mnt/raid/Cloud_PA2$

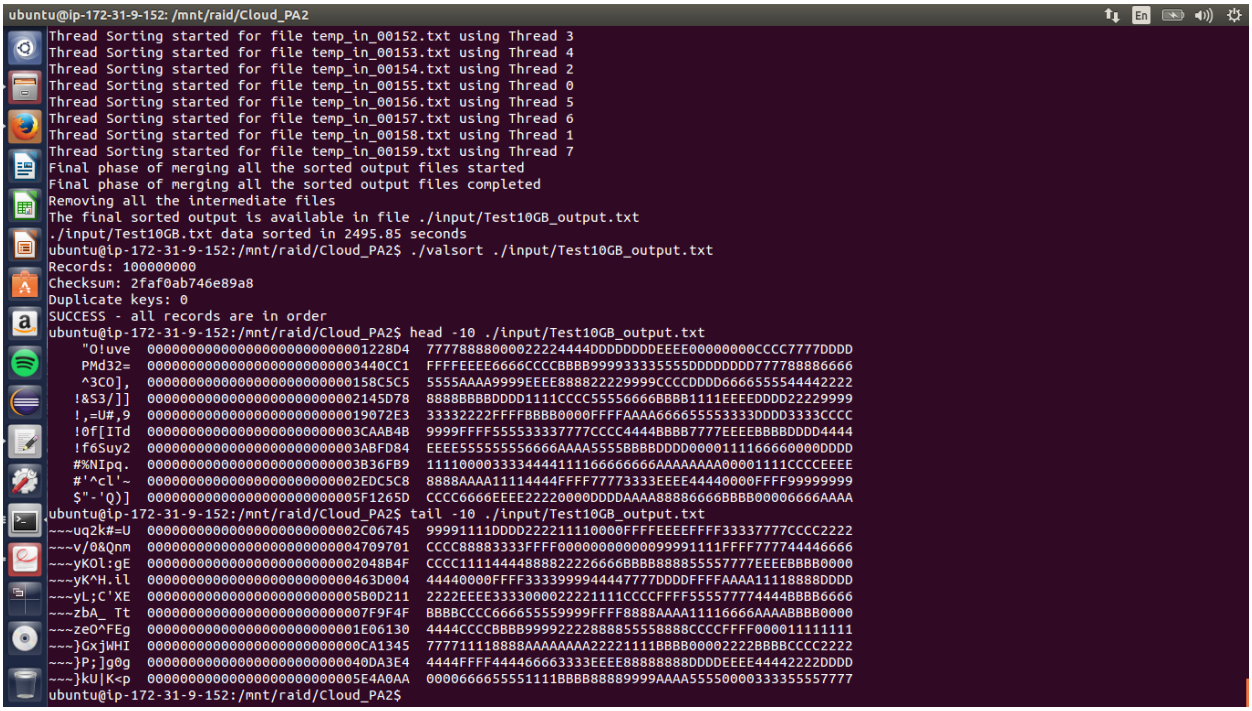
viii. Shared Memory Sort 10 GB 8 Threads



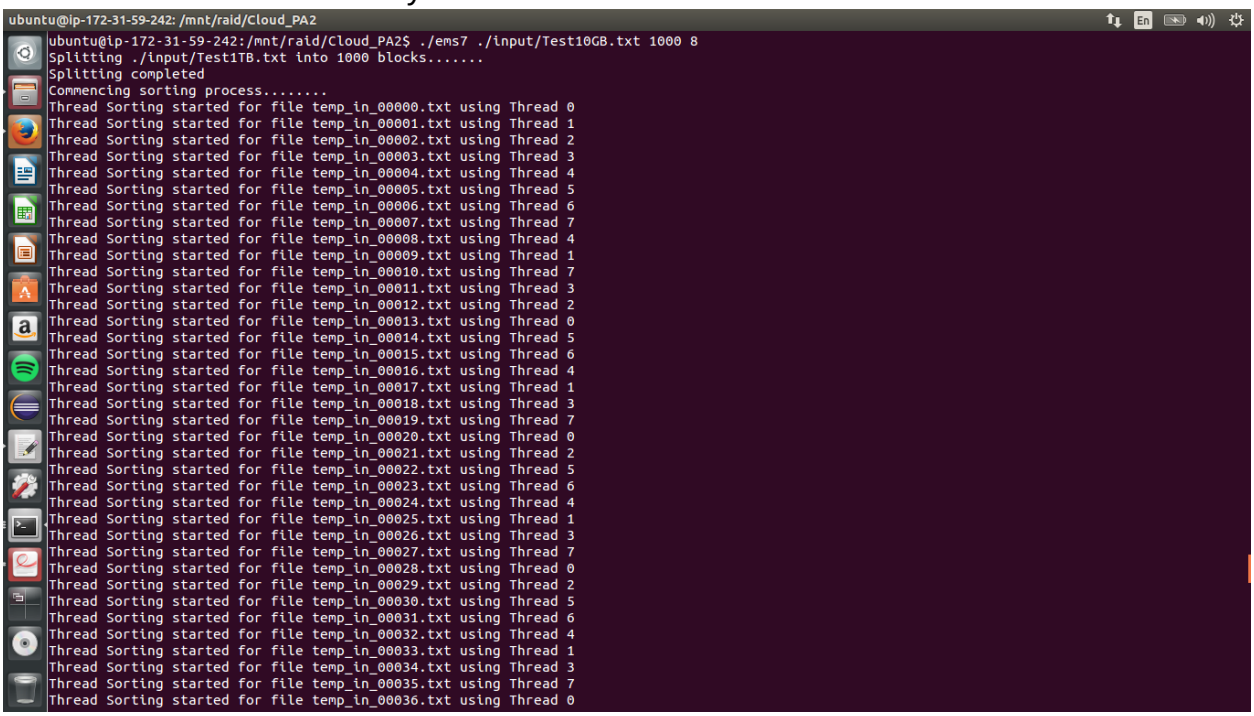


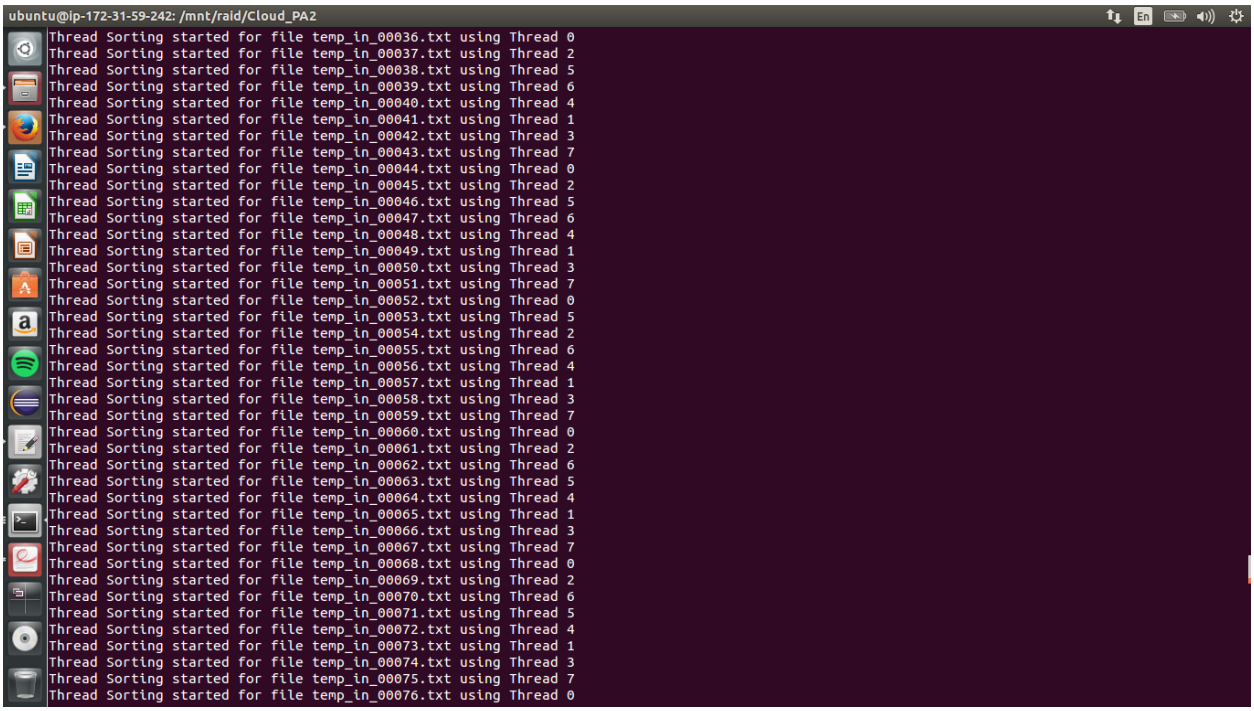

ubuntu@ip-172-31-9-152: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00076.txt using Thread 5
Thread Sorting started for file temp_in_00077.txt using Thread 6
Thread Sorting started for file temp_in_00078.txt using Thread 1
Thread Sorting started for file temp_in_00079.txt using Thread 7
Thread Sorting started for file temp_in_00080.txt using Thread 3
Thread Sorting started for file temp_in_00081.txt using Thread 4
Thread Sorting started for file temp_in_00082.txt using Thread 2
Thread Sorting started for file temp_in_00083.txt using Thread 0
Thread Sorting started for file temp_in_00084.txt using Thread 5
Thread Sorting started for file temp_in_00085.txt using Thread 6
Thread Sorting started for file temp_in_00086.txt using Thread 7
Thread Sorting started for file temp_in_00087.txt using Thread 1
Thread Sorting started for file temp_in_00088.txt using Thread 3
Thread Sorting started for file temp_in_00089.txt using Thread 4
Thread Sorting started for file temp_in_00090.txt using Thread 2
Thread Sorting started for file temp_in_00091.txt using Thread 0
Thread Sorting started for file temp_in_00092.txt using Thread 5
Thread Sorting started for file temp_in_00093.txt using Thread 6
Thread Sorting started for file temp_in_00094.txt using Thread 7
Thread Sorting started for file temp_in_00095.txt using Thread 1
Thread Sorting started for file temp_in_00096.txt using Thread 3
Thread Sorting started for file temp_in_00097.txt using Thread 4
Thread Sorting started for file temp_in_00098.txt using Thread 2
Thread Sorting started for file temp_in_00099.txt using Thread 0
Thread Sorting started for file temp_in_00100.txt using Thread 5
Thread Sorting started for file temp_in_00101.txt using Thread 6
Thread Sorting started for file temp_in_00102.txt using Thread 1
Thread Sorting started for file temp_in_00103.txt using Thread 7
Thread Sorting started for file temp_in_00104.txt using Thread 3
Thread Sorting started for file temp_in_00105.txt using Thread 4
Thread Sorting started for file temp_in_00106.txt using Thread 2
Thread Sorting started for file temp_in_00107.txt using Thread 0
Thread Sorting started for file temp_in_00108.txt using Thread 5
Thread Sorting started for file temp_in_00109.txt using Thread 6
Thread Sorting started for file temp_in_00110.txt using Thread 1
Thread Sorting started for file temp_in_00111.txt using Thread 7
Thread Sorting started for file temp_in_00112.txt using Thread 3
Thread Sorting started for file temp_in_00113.txt using Thread 4
Thread Sorting started for file temp_in_00114.txt using Thread 2
Thread Sorting started for file temp_in_00115.txt using Thread 0
Thread Sorting started for file temp_in_00116.txt using Thread 5

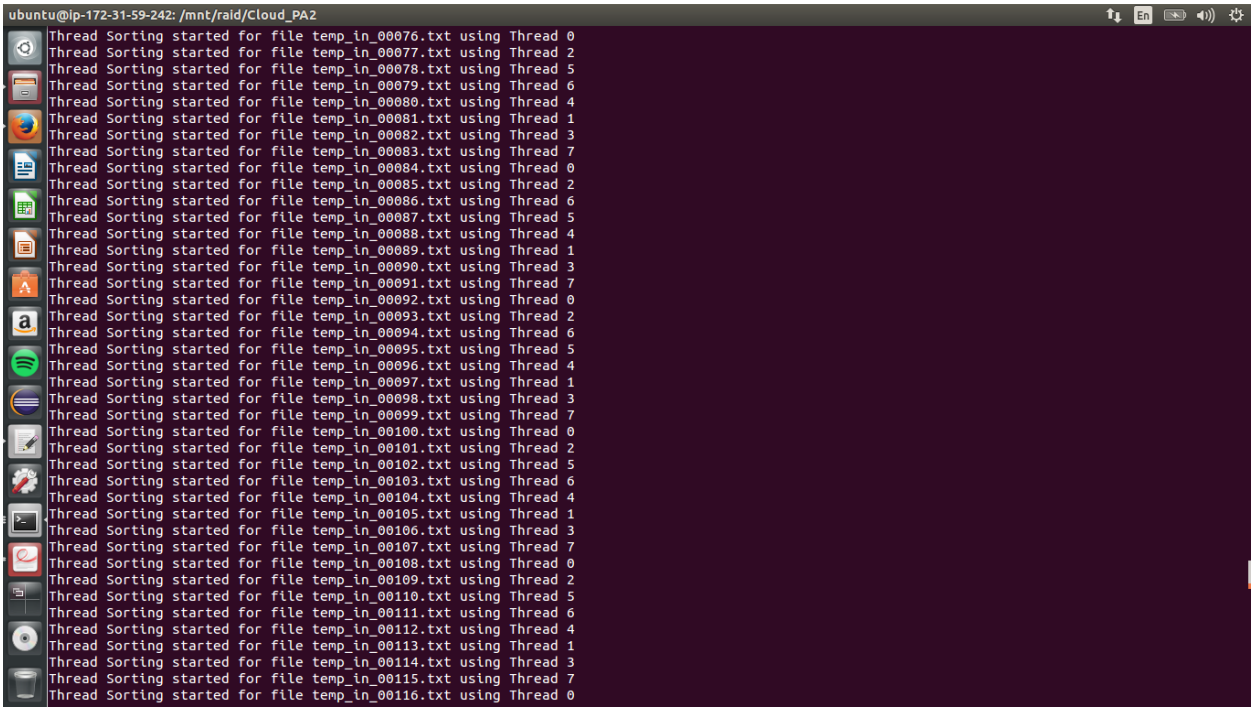

ubuntu@ip-172-31-9-152: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00116.txt using Thread 5
Thread Sorting started for file temp_in_00117.txt using Thread 6
Thread Sorting started for file temp_in_00118.txt using Thread 1
Thread Sorting started for file temp_in_00119.txt using Thread 7
Thread Sorting started for file temp_in_00120.txt using Thread 3
Thread Sorting started for file temp_in_00121.txt using Thread 4
Thread Sorting started for file temp_in_00122.txt using Thread 2
Thread Sorting started for file temp_in_00123.txt using Thread 0
Thread Sorting started for file temp_in_00124.txt using Thread 5
Thread Sorting started for file temp_in_00125.txt using Thread 6
Thread Sorting started for file temp_in_00126.txt using Thread 1
Thread Sorting started for file temp_in_00127.txt using Thread 7
Thread Sorting started for file temp_in_00128.txt using Thread 3
Thread Sorting started for file temp_in_00129.txt using Thread 4
Thread Sorting started for file temp_in_00130.txt using Thread 2
Thread Sorting started for file temp_in_00131.txt using Thread 0
Thread Sorting started for file temp_in_00132.txt using Thread 5
Thread Sorting started for file temp_in_00133.txt using Thread 6
Thread Sorting started for file temp_in_00134.txt using Thread 1
Thread Sorting started for file temp_in_00135.txt using Thread 7
Thread Sorting started for file temp_in_00136.txt using Thread 3
Thread Sorting started for file temp_in_00137.txt using Thread 4
Thread Sorting started for file temp_in_00138.txt using Thread 2
Thread Sorting started for file temp_in_00139.txt using Thread 0
Thread Sorting started for file temp_in_00140.txt using Thread 5
Thread Sorting started for file temp_in_00141.txt using Thread 6
Thread Sorting started for file temp_in_00142.txt using Thread 1
Thread Sorting started for file temp_in_00143.txt using Thread 7
Thread Sorting started for file temp_in_00144.txt using Thread 3
Thread Sorting started for file temp_in_00145.txt using Thread 4
Thread Sorting started for file temp_in_00146.txt using Thread 2
Thread Sorting started for file temp_in_00147.txt using Thread 0
Thread Sorting started for file temp_in_00148.txt using Thread 5
Thread Sorting started for file temp_in_00149.txt using Thread 6
Thread Sorting started for file temp_in_00150.txt using Thread 1
Thread Sorting started for file temp_in_00151.txt using Thread 7
Thread Sorting started for file temp_in_00152.txt using Thread 3
Thread Sorting started for file temp_in_00153.txt using Thread 4
Thread Sorting started for file temp_in_00154.txt using Thread 2
Thread Sorting started for file temp_in_00155.txt using Thread 0
Thread Sorting started for file temp_in_00156.txt using Thread 5

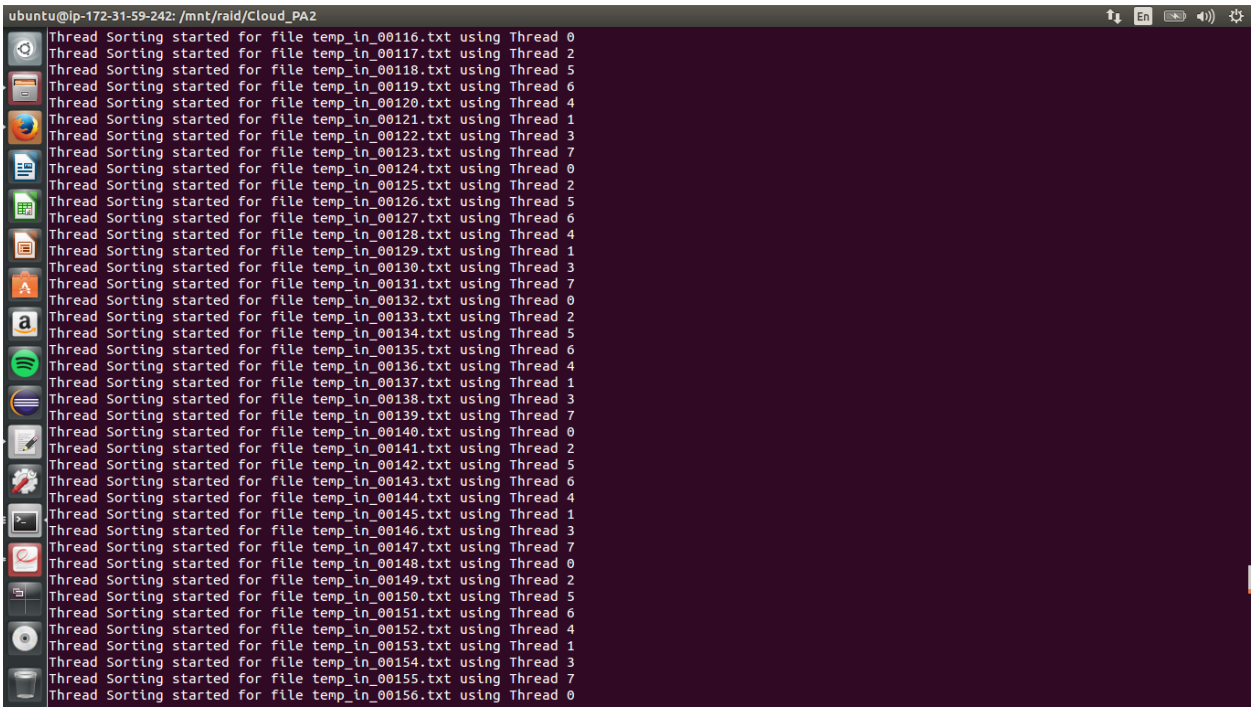


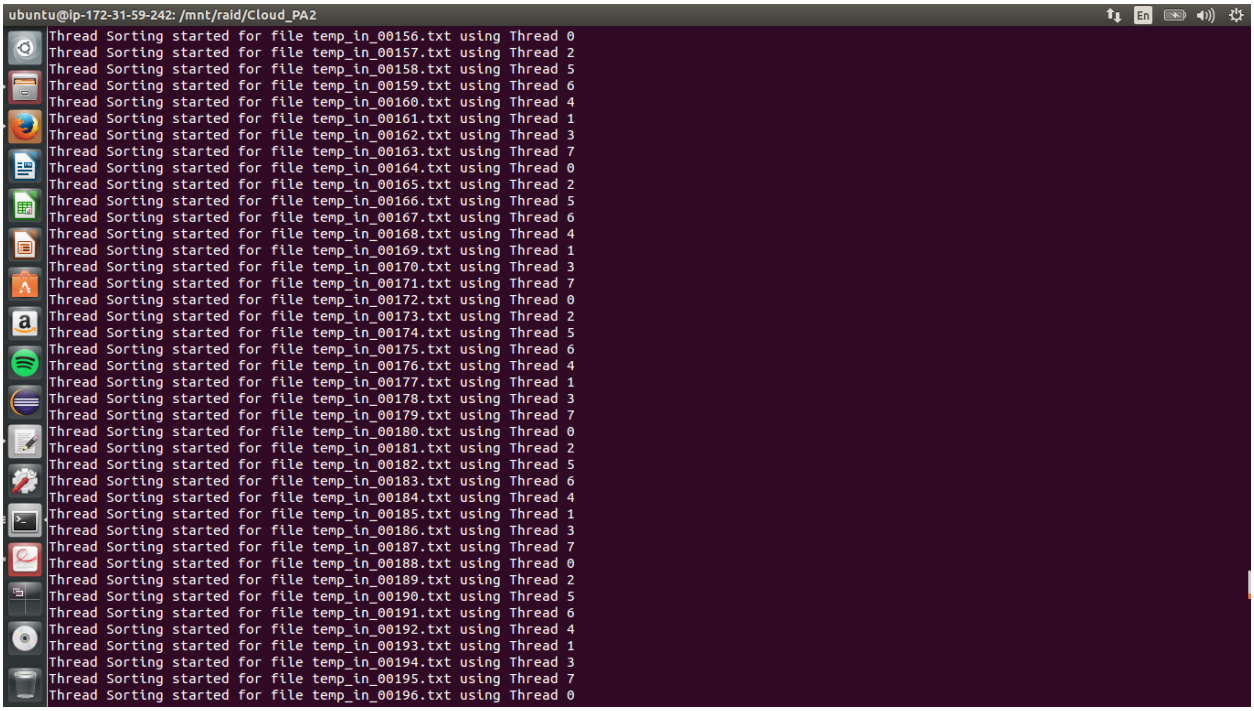
ix. Shared Memory Sort 1 TB 8 Threads

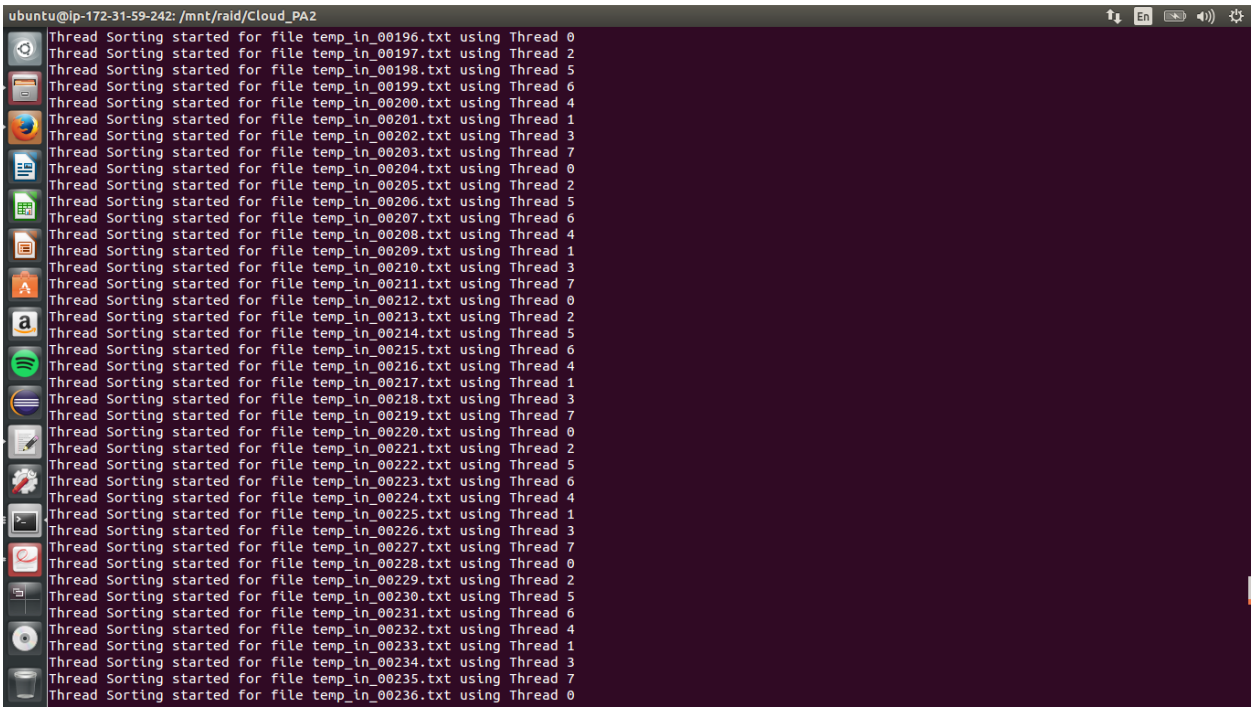


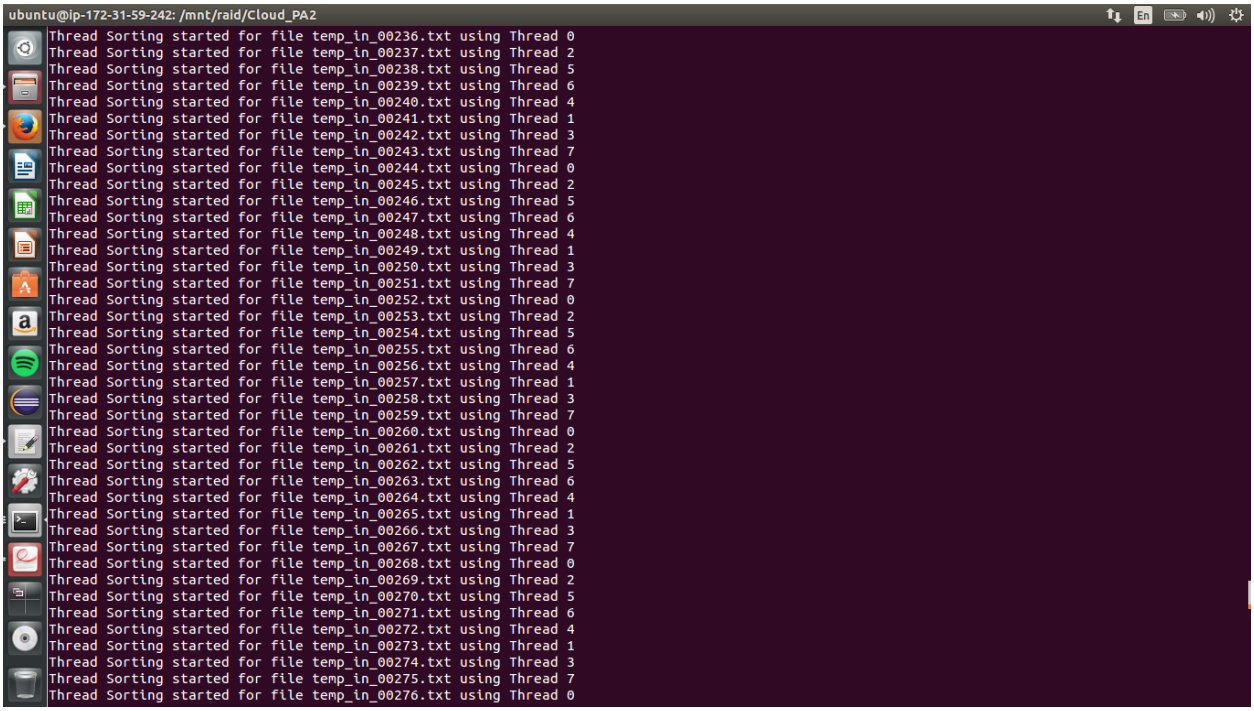

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00036.txt using Thread 0
Thread Sorting started for file temp_in_00037.txt using Thread 2
Thread Sorting started for file temp_in_00038.txt using Thread 5
Thread Sorting started for file temp_in_00039.txt using Thread 6
Thread Sorting started for file temp_in_00040.txt using Thread 4
Thread Sorting started for file temp_in_00041.txt using Thread 1
Thread Sorting started for file temp_in_00042.txt using Thread 3
Thread Sorting started for file temp_in_00043.txt using Thread 7
Thread Sorting started for file temp_in_00044.txt using Thread 0
Thread Sorting started for file temp_in_00045.txt using Thread 2
Thread Sorting started for file temp_in_00046.txt using Thread 5
Thread Sorting started for file temp_in_00047.txt using Thread 6
Thread Sorting started for file temp_in_00048.txt using Thread 4
Thread Sorting started for file temp_in_00049.txt using Thread 1
Thread Sorting started for file temp_in_00050.txt using Thread 3
Thread Sorting started for file temp_in_00051.txt using Thread 7
Thread Sorting started for file temp_in_00052.txt using Thread 0
Thread Sorting started for file temp_in_00053.txt using Thread 5
Thread Sorting started for file temp_in_00054.txt using Thread 2
Thread Sorting started for file temp_in_00055.txt using Thread 6
Thread Sorting started for file temp_in_00056.txt using Thread 4
Thread Sorting started for file temp_in_00057.txt using Thread 1
Thread Sorting started for file temp_in_00058.txt using Thread 3
Thread Sorting started for file temp_in_00059.txt using Thread 7
Thread Sorting started for file temp_in_00060.txt using Thread 0
Thread Sorting started for file temp_in_00061.txt using Thread 2
Thread Sorting started for file temp_in_00062.txt using Thread 6
Thread Sorting started for file temp_in_00063.txt using Thread 5
Thread Sorting started for file temp_in_00064.txt using Thread 4
Thread Sorting started for file temp_in_00065.txt using Thread 1
Thread Sorting started for file temp_in_00066.txt using Thread 3
Thread Sorting started for file temp_in_00067.txt using Thread 7
Thread Sorting started for file temp_in_00068.txt using Thread 0
Thread Sorting started for file temp_in_00069.txt using Thread 2
Thread Sorting started for file temp_in_00070.txt using Thread 6
Thread Sorting started for file temp_in_00071.txt using Thread 5
Thread Sorting started for file temp_in_00072.txt using Thread 4
Thread Sorting started for file temp_in_00073.txt using Thread 1
Thread Sorting started for file temp_in_00074.txt using Thread 3
Thread Sorting started for file temp_in_00075.txt using Thread 7
Thread Sorting started for file temp_in_00076.txt using Thread 0

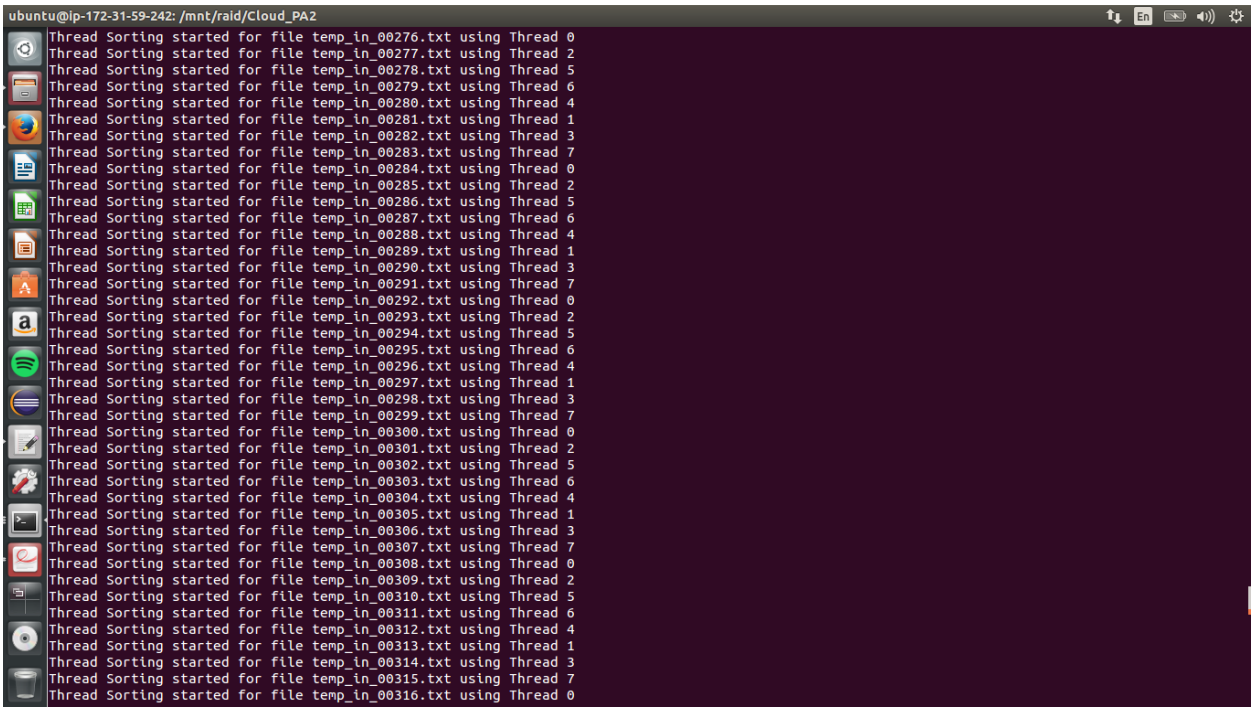

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00076.txt using Thread 0
Thread Sorting started for file temp_in_00077.txt using Thread 2
Thread Sorting started for file temp_in_00078.txt using Thread 5
Thread Sorting started for file temp_in_00079.txt using Thread 6
Thread Sorting started for file temp_in_00080.txt using Thread 4
Thread Sorting started for file temp_in_00081.txt using Thread 1
Thread Sorting started for file temp_in_00082.txt using Thread 3
Thread Sorting started for file temp_in_00083.txt using Thread 7
Thread Sorting started for file temp_in_00084.txt using Thread 0
Thread Sorting started for file temp_in_00085.txt using Thread 2
Thread Sorting started for file temp_in_00086.txt using Thread 6
Thread Sorting started for file temp_in_00087.txt using Thread 5
Thread Sorting started for file temp_in_00088.txt using Thread 4
Thread Sorting started for file temp_in_00089.txt using Thread 1
Thread Sorting started for file temp_in_00090.txt using Thread 3
Thread Sorting started for file temp_in_00091.txt using Thread 7
Thread Sorting started for file temp_in_00092.txt using Thread 0
Thread Sorting started for file temp_in_00093.txt using Thread 2
Thread Sorting started for file temp_in_00094.txt using Thread 6
Thread Sorting started for file temp_in_00095.txt using Thread 5
Thread Sorting started for file temp_in_00096.txt using Thread 4
Thread Sorting started for file temp_in_00097.txt using Thread 1
Thread Sorting started for file temp_in_00098.txt using Thread 3
Thread Sorting started for file temp_in_00099.txt using Thread 7
Thread Sorting started for file temp_in_00100.txt using Thread 0
Thread Sorting started for file temp_in_00101.txt using Thread 2
Thread Sorting started for file temp_in_00102.txt using Thread 5
Thread Sorting started for file temp_in_00103.txt using Thread 6
Thread Sorting started for file temp_in_00104.txt using Thread 4
Thread Sorting started for file temp_in_00105.txt using Thread 1
Thread Sorting started for file temp_in_00106.txt using Thread 3
Thread Sorting started for file temp_in_00107.txt using Thread 7
Thread Sorting started for file temp_in_00108.txt using Thread 0
Thread Sorting started for file temp_in_00109.txt using Thread 2
Thread Sorting started for file temp_in_00110.txt using Thread 5
Thread Sorting started for file temp_in_00111.txt using Thread 6
Thread Sorting started for file temp_in_00112.txt using Thread 4
Thread Sorting started for file temp_in_00113.txt using Thread 1
Thread Sorting started for file temp_in_00114.txt using Thread 3
Thread Sorting started for file temp_in_00115.txt using Thread 7
Thread Sorting started for file temp_in_00116.txt using Thread 0

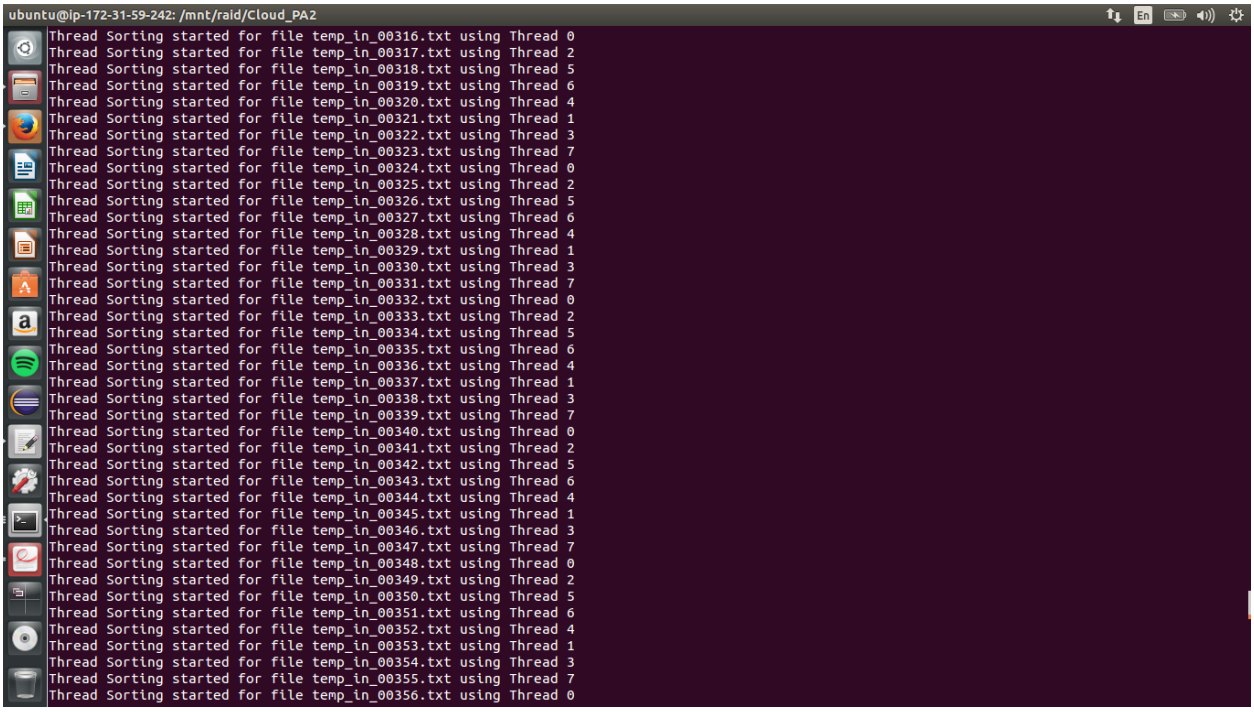

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00116.txt using Thread 0
Thread Sorting started for file temp_in_00117.txt using Thread 2
Thread Sorting started for file temp_in_00118.txt using Thread 5
Thread Sorting started for file temp_in_00119.txt using Thread 6
Thread Sorting started for file temp_in_00120.txt using Thread 4
Thread Sorting started for file temp_in_00121.txt using Thread 1
Thread Sorting started for file temp_in_00122.txt using Thread 3
Thread Sorting started for file temp_in_00123.txt using Thread 7
Thread Sorting started for file temp_in_00124.txt using Thread 0
Thread Sorting started for file temp_in_00125.txt using Thread 2
Thread Sorting started for file temp_in_00126.txt using Thread 5
Thread Sorting started for file temp_in_00127.txt using Thread 6
Thread Sorting started for file temp_in_00128.txt using Thread 4
Thread Sorting started for file temp_in_00129.txt using Thread 1
Thread Sorting started for file temp_in_00130.txt using Thread 3
Thread Sorting started for file temp_in_00131.txt using Thread 7
Thread Sorting started for file temp_in_00132.txt using Thread 0
Thread Sorting started for file temp_in_00133.txt using Thread 2
Thread Sorting started for file temp_in_00134.txt using Thread 5
Thread Sorting started for file temp_in_00135.txt using Thread 6
Thread Sorting started for file temp_in_00136.txt using Thread 4
Thread Sorting started for file temp_in_00137.txt using Thread 1
Thread Sorting started for file temp_in_00138.txt using Thread 3
Thread Sorting started for file temp_in_00139.txt using Thread 7
Thread Sorting started for file temp_in_00140.txt using Thread 0
Thread Sorting started for file temp_in_00141.txt using Thread 2
Thread Sorting started for file temp_in_00142.txt using Thread 5
Thread Sorting started for file temp_in_00143.txt using Thread 6
Thread Sorting started for file temp_in_00144.txt using Thread 4
Thread Sorting started for file temp_in_00145.txt using Thread 1
Thread Sorting started for file temp_in_00146.txt using Thread 3
Thread Sorting started for file temp_in_00147.txt using Thread 7
Thread Sorting started for file temp_in_00148.txt using Thread 0
Thread Sorting started for file temp_in_00149.txt using Thread 2
Thread Sorting started for file temp_in_00150.txt using Thread 5
Thread Sorting started for file temp_in_00151.txt using Thread 6
Thread Sorting started for file temp_in_00152.txt using Thread 4
Thread Sorting started for file temp_in_00153.txt using Thread 1
Thread Sorting started for file temp_in_00154.txt using Thread 3
Thread Sorting started for file temp_in_00155.txt using Thread 7
Thread Sorting started for file temp_in_00156.txt using Thread 0

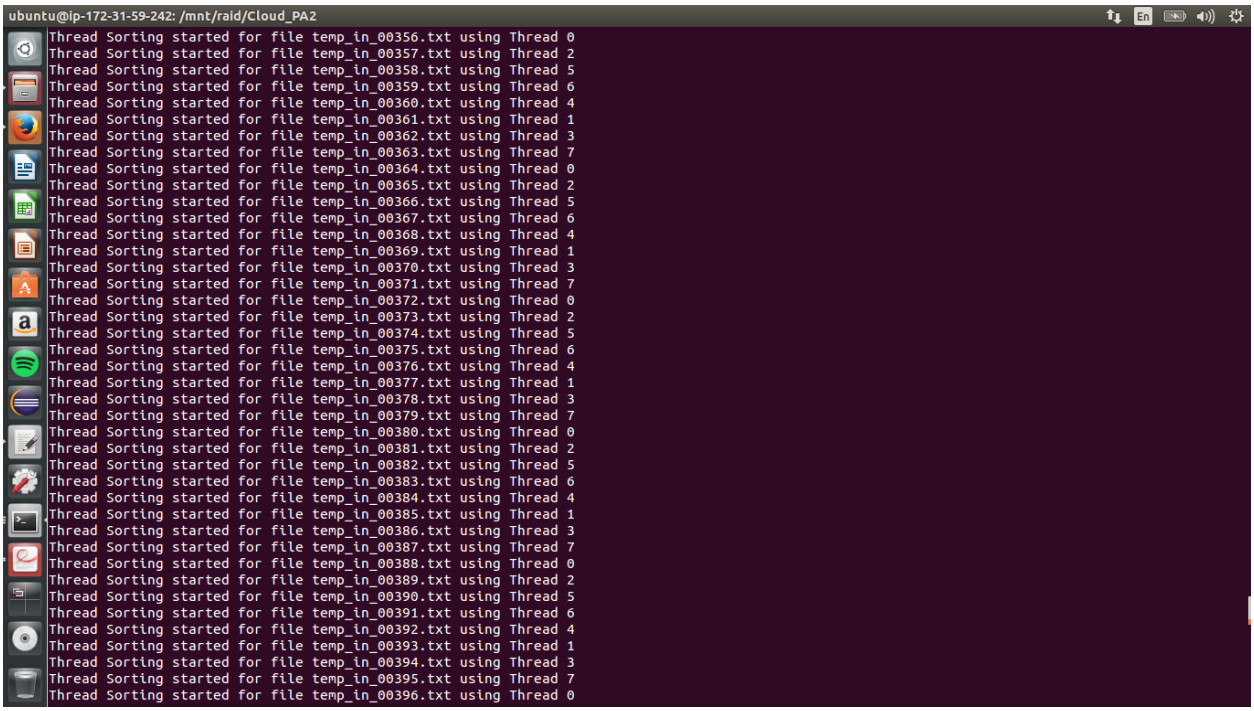

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00156.txt using Thread 0
Thread Sorting started for file temp_in_00157.txt using Thread 2
Thread Sorting started for file temp_in_00158.txt using Thread 5
Thread Sorting started for file temp_in_00159.txt using Thread 6
Thread Sorting started for file temp_in_00160.txt using Thread 4
Thread Sorting started for file temp_in_00161.txt using Thread 1
Thread Sorting started for file temp_in_00162.txt using Thread 3
Thread Sorting started for file temp_in_00163.txt using Thread 7
Thread Sorting started for file temp_in_00164.txt using Thread 0
Thread Sorting started for file temp_in_00165.txt using Thread 2
Thread Sorting started for file temp_in_00166.txt using Thread 5
Thread Sorting started for file temp_in_00167.txt using Thread 6
Thread Sorting started for file temp_in_00168.txt using Thread 4
Thread Sorting started for file temp_in_00169.txt using Thread 1
Thread Sorting started for file temp_in_00170.txt using Thread 3
Thread Sorting started for file temp_in_00171.txt using Thread 7
Thread Sorting started for file temp_in_00172.txt using Thread 0
Thread Sorting started for file temp_in_00173.txt using Thread 2
Thread Sorting started for file temp_in_00174.txt using Thread 5
Thread Sorting started for file temp_in_00175.txt using Thread 6
Thread Sorting started for file temp_in_00176.txt using Thread 4
Thread Sorting started for file temp_in_00177.txt using Thread 1
Thread Sorting started for file temp_in_00178.txt using Thread 3
Thread Sorting started for file temp_in_00179.txt using Thread 7
Thread Sorting started for file temp_in_00180.txt using Thread 0
Thread Sorting started for file temp_in_00181.txt using Thread 2
Thread Sorting started for file temp_in_00182.txt using Thread 5
Thread Sorting started for file temp_in_00183.txt using Thread 6
Thread Sorting started for file temp_in_00184.txt using Thread 4
Thread Sorting started for file temp_in_00185.txt using Thread 1
Thread Sorting started for file temp_in_00186.txt using Thread 3
Thread Sorting started for file temp_in_00187.txt using Thread 7
Thread Sorting started for file temp_in_00188.txt using Thread 0
Thread Sorting started for file temp_in_00189.txt using Thread 2
Thread Sorting started for file temp_in_00190.txt using Thread 5
Thread Sorting started for file temp_in_00191.txt using Thread 6
Thread Sorting started for file temp_in_00192.txt using Thread 4
Thread Sorting started for file temp_in_00193.txt using Thread 1
Thread Sorting started for file temp_in_00194.txt using Thread 3
Thread Sorting started for file temp_in_00195.txt using Thread 7
Thread Sorting started for file temp_in_00196.txt using Thread 0

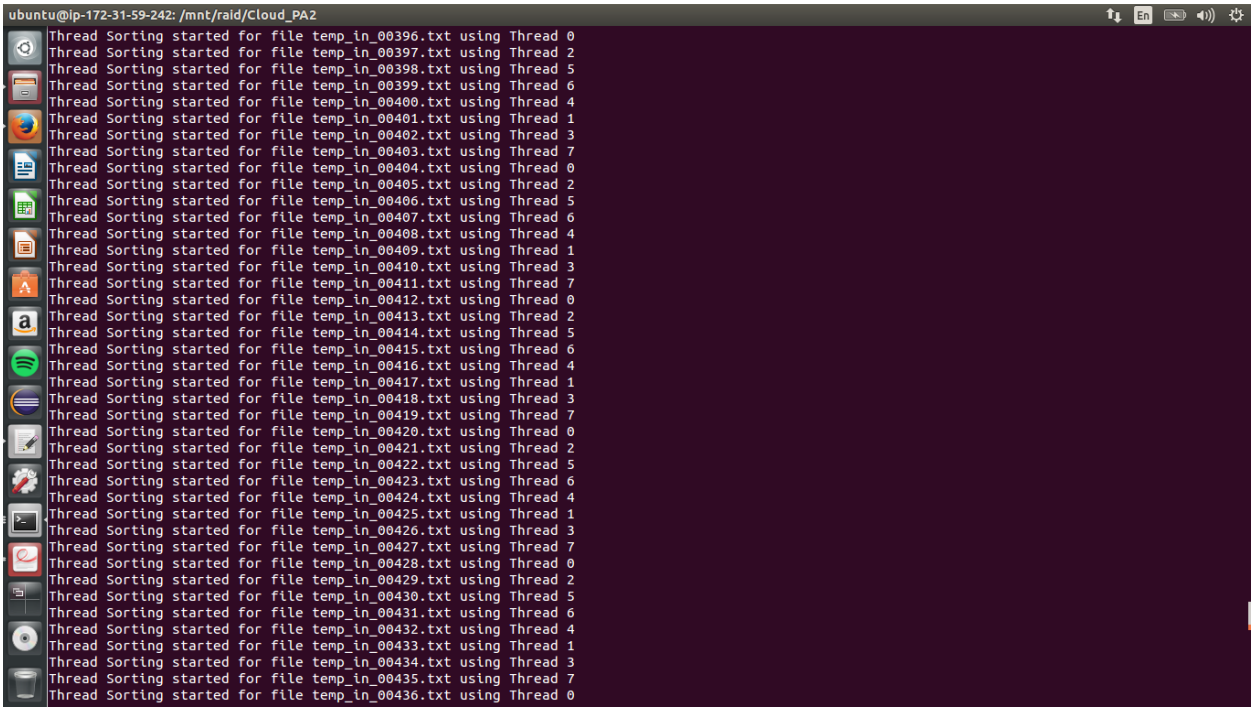

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00196.txt using Thread 0
Thread Sorting started for file temp_in_00197.txt using Thread 2
Thread Sorting started for file temp_in_00198.txt using Thread 5
Thread Sorting started for file temp_in_00199.txt using Thread 6
Thread Sorting started for file temp_in_00200.txt using Thread 4
Thread Sorting started for file temp_in_00201.txt using Thread 1
Thread Sorting started for file temp_in_00202.txt using Thread 3
Thread Sorting started for file temp_in_00203.txt using Thread 7
Thread Sorting started for file temp_in_00204.txt using Thread 0
Thread Sorting started for file temp_in_00205.txt using Thread 2
Thread Sorting started for file temp_in_00206.txt using Thread 5
Thread Sorting started for file temp_in_00207.txt using Thread 6
Thread Sorting started for file temp_in_00208.txt using Thread 4
Thread Sorting started for file temp_in_00209.txt using Thread 1
Thread Sorting started for file temp_in_00210.txt using Thread 3
Thread Sorting started for file temp_in_00211.txt using Thread 7
Thread Sorting started for file temp_in_00212.txt using Thread 0
Thread Sorting started for file temp_in_00213.txt using Thread 2
Thread Sorting started for file temp_in_00214.txt using Thread 5
Thread Sorting started for file temp_in_00215.txt using Thread 6
Thread Sorting started for file temp_in_00216.txt using Thread 4
Thread Sorting started for file temp_in_00217.txt using Thread 1
Thread Sorting started for file temp_in_00218.txt using Thread 3
Thread Sorting started for file temp_in_00219.txt using Thread 7
Thread Sorting started for file temp_in_00220.txt using Thread 0
Thread Sorting started for file temp_in_00221.txt using Thread 2
Thread Sorting started for file temp_in_00222.txt using Thread 5
Thread Sorting started for file temp_in_00223.txt using Thread 6
Thread Sorting started for file temp_in_00224.txt using Thread 4
Thread Sorting started for file temp_in_00225.txt using Thread 1
Thread Sorting started for file temp_in_00226.txt using Thread 3
Thread Sorting started for file temp_in_00227.txt using Thread 7
Thread Sorting started for file temp_in_00228.txt using Thread 0
Thread Sorting started for file temp_in_00229.txt using Thread 2
Thread Sorting started for file temp_in_00230.txt using Thread 5
Thread Sorting started for file temp_in_00231.txt using Thread 6
Thread Sorting started for file temp_in_00232.txt using Thread 4
Thread Sorting started for file temp_in_00233.txt using Thread 1
Thread Sorting started for file temp_in_00234.txt using Thread 3
Thread Sorting started for file temp_in_00235.txt using Thread 7
Thread Sorting started for file temp_in_00236.txt using Thread 0

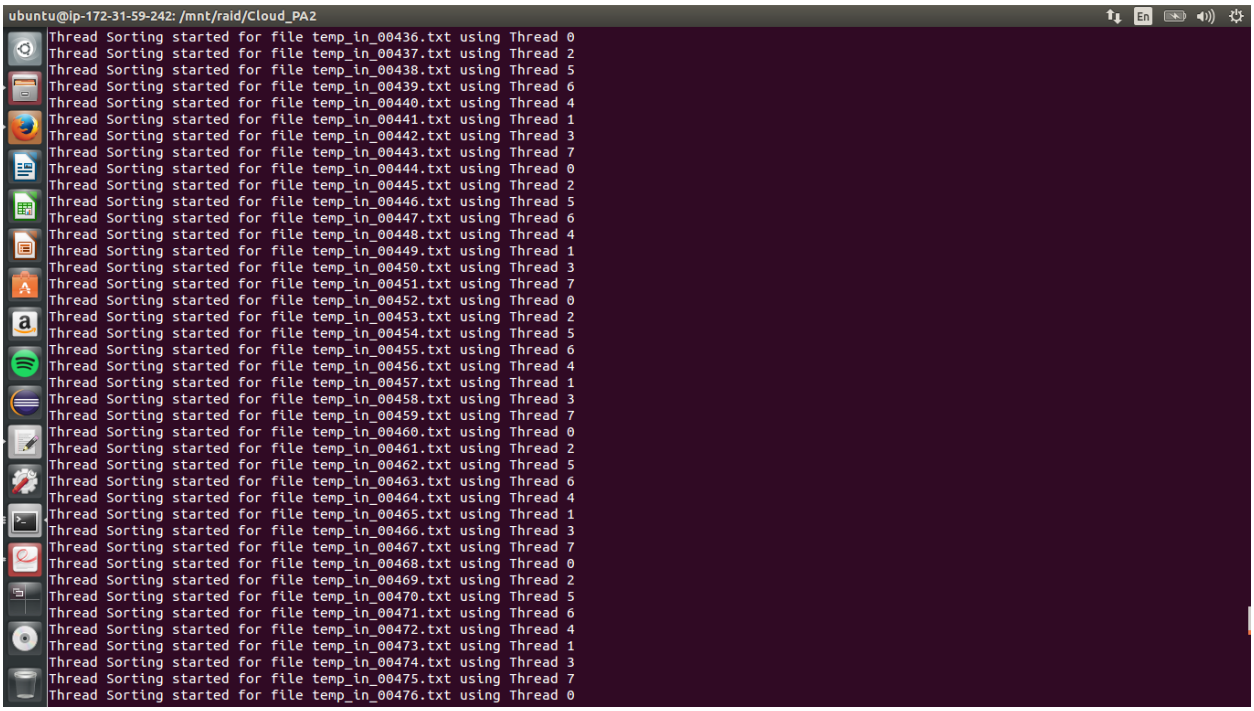

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00236.txt using Thread 0
Thread Sorting started for file temp_in_00237.txt using Thread 2
Thread Sorting started for file temp_in_00238.txt using Thread 5
Thread Sorting started for file temp_in_00239.txt using Thread 6
Thread Sorting started for file temp_in_00240.txt using Thread 4
Thread Sorting started for file temp_in_00241.txt using Thread 1
Thread Sorting started for file temp_in_00242.txt using Thread 3
Thread Sorting started for file temp_in_00243.txt using Thread 7
Thread Sorting started for file temp_in_00244.txt using Thread 0
Thread Sorting started for file temp_in_00245.txt using Thread 2
Thread Sorting started for file temp_in_00246.txt using Thread 5
Thread Sorting started for file temp_in_00247.txt using Thread 6
Thread Sorting started for file temp_in_00248.txt using Thread 4
Thread Sorting started for file temp_in_00249.txt using Thread 1
Thread Sorting started for file temp_in_00250.txt using Thread 3
Thread Sorting started for file temp_in_00251.txt using Thread 7
Thread Sorting started for file temp_in_00252.txt using Thread 0
Thread Sorting started for file temp_in_00253.txt using Thread 2
Thread Sorting started for file temp_in_00254.txt using Thread 5
Thread Sorting started for file temp_in_00255.txt using Thread 6
Thread Sorting started for file temp_in_00256.txt using Thread 4
Thread Sorting started for file temp_in_00257.txt using Thread 1
Thread Sorting started for file temp_in_00258.txt using Thread 3
Thread Sorting started for file temp_in_00259.txt using Thread 7
Thread Sorting started for file temp_in_00260.txt using Thread 0
Thread Sorting started for file temp_in_00261.txt using Thread 2
Thread Sorting started for file temp_in_00262.txt using Thread 5
Thread Sorting started for file temp_in_00263.txt using Thread 6
Thread Sorting started for file temp_in_00264.txt using Thread 4
Thread Sorting started for file temp_in_00265.txt using Thread 1
Thread Sorting started for file temp_in_00266.txt using Thread 3
Thread Sorting started for file temp_in_00267.txt using Thread 7
Thread Sorting started for file temp_in_00268.txt using Thread 0
Thread Sorting started for file temp_in_00269.txt using Thread 2
Thread Sorting started for file temp_in_00270.txt using Thread 5
Thread Sorting started for file temp_in_00271.txt using Thread 6
Thread Sorting started for file temp_in_00272.txt using Thread 4
Thread Sorting started for file temp_in_00273.txt using Thread 1
Thread Sorting started for file temp_in_00274.txt using Thread 3
Thread Sorting started for file temp_in_00275.txt using Thread 7
Thread Sorting started for file temp_in_00276.txt using Thread 0

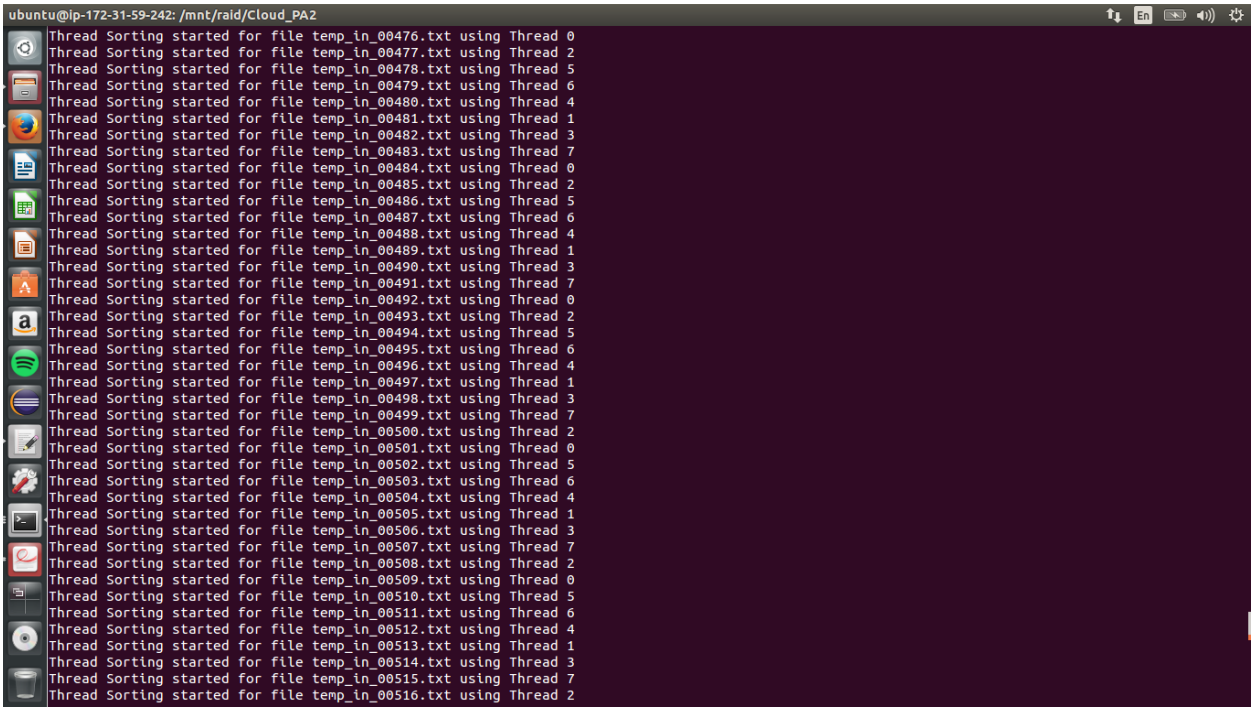

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00276.txt using Thread 0
Thread Sorting started for file temp_in_00277.txt using Thread 2
Thread Sorting started for file temp_in_00278.txt using Thread 5
Thread Sorting started for file temp_in_00279.txt using Thread 6
Thread Sorting started for file temp_in_00280.txt using Thread 4
Thread Sorting started for file temp_in_00281.txt using Thread 1
Thread Sorting started for file temp_in_00282.txt using Thread 3
Thread Sorting started for file temp_in_00283.txt using Thread 7
Thread Sorting started for file temp_in_00284.txt using Thread 0
Thread Sorting started for file temp_in_00285.txt using Thread 2
Thread Sorting started for file temp_in_00286.txt using Thread 5
Thread Sorting started for file temp_in_00287.txt using Thread 6
Thread Sorting started for file temp_in_00288.txt using Thread 4
Thread Sorting started for file temp_in_00289.txt using Thread 1
Thread Sorting started for file temp_in_00290.txt using Thread 3
Thread Sorting started for file temp_in_00291.txt using Thread 7
Thread Sorting started for file temp_in_00292.txt using Thread 0
Thread Sorting started for file temp_in_00293.txt using Thread 2
Thread Sorting started for file temp_in_00294.txt using Thread 5
Thread Sorting started for file temp_in_00295.txt using Thread 6
Thread Sorting started for file temp_in_00296.txt using Thread 4
Thread Sorting started for file temp_in_00297.txt using Thread 1
Thread Sorting started for file temp_in_00298.txt using Thread 3
Thread Sorting started for file temp_in_00299.txt using Thread 7
Thread Sorting started for file temp_in_00300.txt using Thread 0
Thread Sorting started for file temp_in_00301.txt using Thread 2
Thread Sorting started for file temp_in_00302.txt using Thread 5
Thread Sorting started for file temp_in_00303.txt using Thread 6
Thread Sorting started for file temp_in_00304.txt using Thread 4
Thread Sorting started for file temp_in_00305.txt using Thread 1
Thread Sorting started for file temp_in_00306.txt using Thread 3
Thread Sorting started for file temp_in_00307.txt using Thread 7
Thread Sorting started for file temp_in_00308.txt using Thread 0
Thread Sorting started for file temp_in_00309.txt using Thread 2
Thread Sorting started for file temp_in_00310.txt using Thread 5
Thread Sorting started for file temp_in_00311.txt using Thread 6
Thread Sorting started for file temp_in_00312.txt using Thread 4
Thread Sorting started for file temp_in_00313.txt using Thread 1
Thread Sorting started for file temp_in_00314.txt using Thread 3
Thread Sorting started for file temp_in_00315.txt using Thread 7
Thread Sorting started for file temp_in_00316.txt using Thread 0

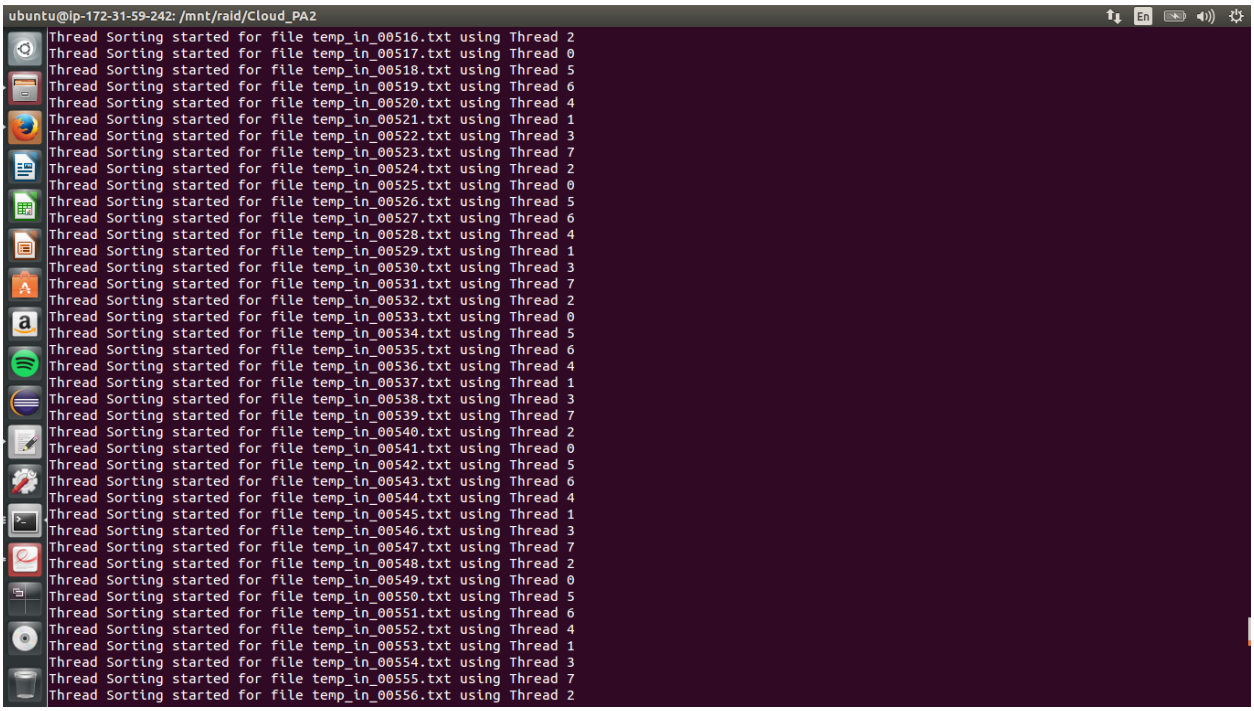

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00316.txt using Thread 0
Thread Sorting started for file temp_in_00317.txt using Thread 2
Thread Sorting started for file temp_in_00318.txt using Thread 5
Thread Sorting started for file temp_in_00319.txt using Thread 6
Thread Sorting started for file temp_in_00320.txt using Thread 4
Thread Sorting started for file temp_in_00321.txt using Thread 1
Thread Sorting started for file temp_in_00322.txt using Thread 3
Thread Sorting started for file temp_in_00323.txt using Thread 7
Thread Sorting started for file temp_in_00324.txt using Thread 0
Thread Sorting started for file temp_in_00325.txt using Thread 2
Thread Sorting started for file temp_in_00326.txt using Thread 5
Thread Sorting started for file temp_in_00327.txt using Thread 6
Thread Sorting started for file temp_in_00328.txt using Thread 4
Thread Sorting started for file temp_in_00329.txt using Thread 1
Thread Sorting started for file temp_in_00330.txt using Thread 3
Thread Sorting started for file temp_in_00331.txt using Thread 7
Thread Sorting started for file temp_in_00332.txt using Thread 0
Thread Sorting started for file temp_in_00333.txt using Thread 2
Thread Sorting started for file temp_in_00334.txt using Thread 5
Thread Sorting started for file temp_in_00335.txt using Thread 6
Thread Sorting started for file temp_in_00336.txt using Thread 4
Thread Sorting started for file temp_in_00337.txt using Thread 1
Thread Sorting started for file temp_in_00338.txt using Thread 3
Thread Sorting started for file temp_in_00339.txt using Thread 7
Thread Sorting started for file temp_in_00340.txt using Thread 0
Thread Sorting started for file temp_in_00341.txt using Thread 2
Thread Sorting started for file temp_in_00342.txt using Thread 5
Thread Sorting started for file temp_in_00343.txt using Thread 6
Thread Sorting started for file temp_in_00344.txt using Thread 4
Thread Sorting started for file temp_in_00345.txt using Thread 1
Thread Sorting started for file temp_in_00346.txt using Thread 3
Thread Sorting started for file temp_in_00347.txt using Thread 7
Thread Sorting started for file temp_in_00348.txt using Thread 0
Thread Sorting started for file temp_in_00349.txt using Thread 2
Thread Sorting started for file temp_in_00350.txt using Thread 5
Thread Sorting started for file temp_in_00351.txt using Thread 6
Thread Sorting started for file temp_in_00352.txt using Thread 4
Thread Sorting started for file temp_in_00353.txt using Thread 1
Thread Sorting started for file temp_in_00354.txt using Thread 3
Thread Sorting started for file temp_in_00355.txt using Thread 7
Thread Sorting started for file temp_in_00356.txt using Thread 0

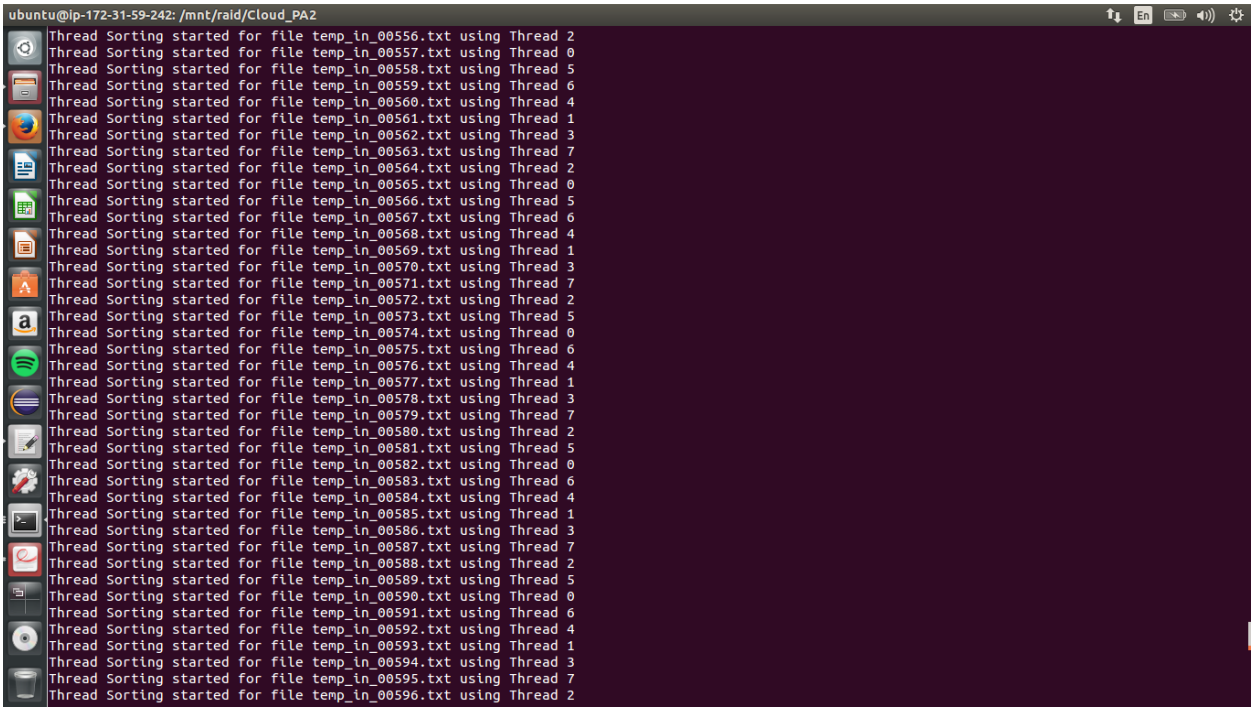

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00356.txt using Thread 0
Thread Sorting started for file temp_in_00357.txt using Thread 2
Thread Sorting started for file temp_in_00358.txt using Thread 5
Thread Sorting started for file temp_in_00359.txt using Thread 6
Thread Sorting started for file temp_in_00360.txt using Thread 4
Thread Sorting started for file temp_in_00361.txt using Thread 1
Thread Sorting started for file temp_in_00362.txt using Thread 3
Thread Sorting started for file temp_in_00363.txt using Thread 7
Thread Sorting started for file temp_in_00364.txt using Thread 0
Thread Sorting started for file temp_in_00365.txt using Thread 2
Thread Sorting started for file temp_in_00366.txt using Thread 5
Thread Sorting started for file temp_in_00367.txt using Thread 6
Thread Sorting started for file temp_in_00368.txt using Thread 4
Thread Sorting started for file temp_in_00369.txt using Thread 1
Thread Sorting started for file temp_in_00370.txt using Thread 3
Thread Sorting started for file temp_in_00371.txt using Thread 7
Thread Sorting started for file temp_in_00372.txt using Thread 0
Thread Sorting started for file temp_in_00373.txt using Thread 2
Thread Sorting started for file temp_in_00374.txt using Thread 5
Thread Sorting started for file temp_in_00375.txt using Thread 6
Thread Sorting started for file temp_in_00376.txt using Thread 4
Thread Sorting started for file temp_in_00377.txt using Thread 1
Thread Sorting started for file temp_in_00378.txt using Thread 3
Thread Sorting started for file temp_in_00379.txt using Thread 7
Thread Sorting started for file temp_in_00380.txt using Thread 0
Thread Sorting started for file temp_in_00381.txt using Thread 2
Thread Sorting started for file temp_in_00382.txt using Thread 5
Thread Sorting started for file temp_in_00383.txt using Thread 6
Thread Sorting started for file temp_in_00384.txt using Thread 4
Thread Sorting started for file temp_in_00385.txt using Thread 1
Thread Sorting started for file temp_in_00386.txt using Thread 3
Thread Sorting started for file temp_in_00387.txt using Thread 7
Thread Sorting started for file temp_in_00388.txt using Thread 0
Thread Sorting started for file temp_in_00389.txt using Thread 2
Thread Sorting started for file temp_in_00390.txt using Thread 5
Thread Sorting started for file temp_in_00391.txt using Thread 6
Thread Sorting started for file temp_in_00392.txt using Thread 4
Thread Sorting started for file temp_in_00393.txt using Thread 1
Thread Sorting started for file temp_in_00394.txt using Thread 3
Thread Sorting started for file temp_in_00395.txt using Thread 7
Thread Sorting started for file temp_in_00396.txt using Thread 0

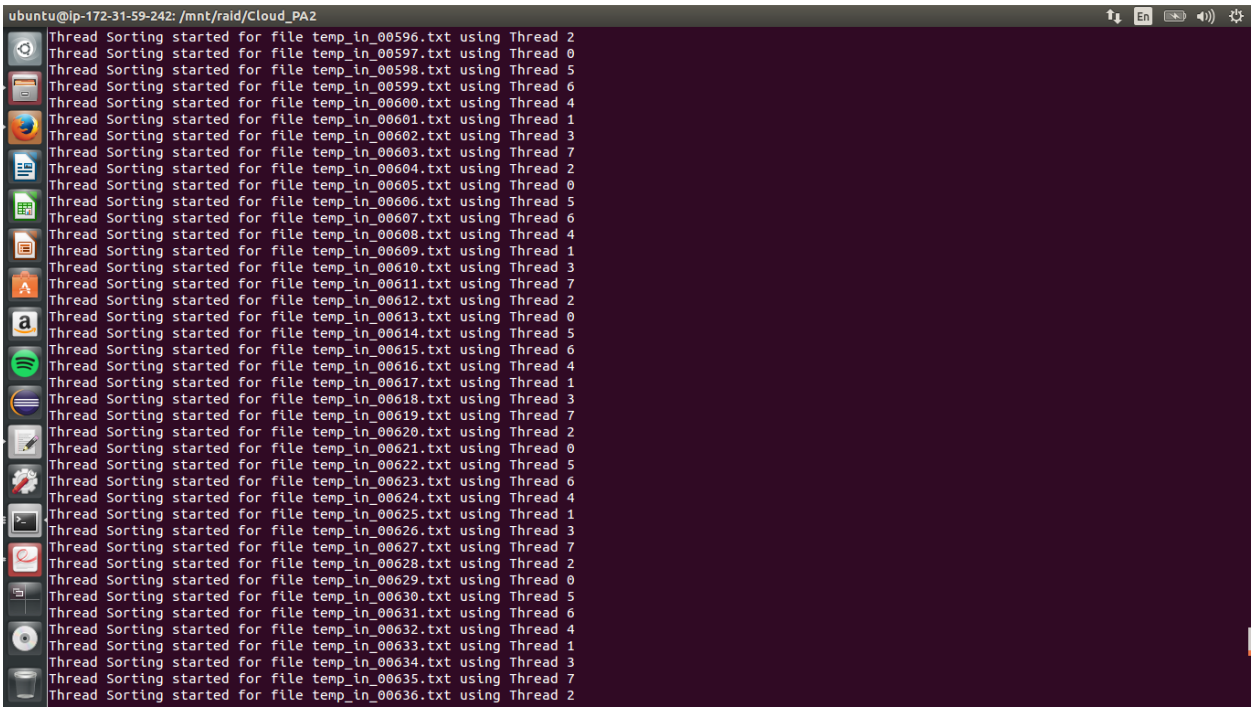

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00396.txt using Thread 0
Thread Sorting started for file temp_in_00397.txt using Thread 2
Thread Sorting started for file temp_in_00398.txt using Thread 5
Thread Sorting started for file temp_in_00399.txt using Thread 6
Thread Sorting started for file temp_in_00400.txt using Thread 4
Thread Sorting started for file temp_in_00401.txt using Thread 1
Thread Sorting started for file temp_in_00402.txt using Thread 3
Thread Sorting started for file temp_in_00403.txt using Thread 7
Thread Sorting started for file temp_in_00404.txt using Thread 0
Thread Sorting started for file temp_in_00405.txt using Thread 2
Thread Sorting started for file temp_in_00406.txt using Thread 5
Thread Sorting started for file temp_in_00407.txt using Thread 6
Thread Sorting started for file temp_in_00408.txt using Thread 4
Thread Sorting started for file temp_in_00409.txt using Thread 1
Thread Sorting started for file temp_in_00410.txt using Thread 3
Thread Sorting started for file temp_in_00411.txt using Thread 7
Thread Sorting started for file temp_in_00412.txt using Thread 0
Thread Sorting started for file temp_in_00413.txt using Thread 2
Thread Sorting started for file temp_in_00414.txt using Thread 5
Thread Sorting started for file temp_in_00415.txt using Thread 6
Thread Sorting started for file temp_in_00416.txt using Thread 4
Thread Sorting started for file temp_in_00417.txt using Thread 1
Thread Sorting started for file temp_in_00418.txt using Thread 3
Thread Sorting started for file temp_in_00419.txt using Thread 7
Thread Sorting started for file temp_in_00420.txt using Thread 0
Thread Sorting started for file temp_in_00421.txt using Thread 2
Thread Sorting started for file temp_in_00422.txt using Thread 5
Thread Sorting started for file temp_in_00423.txt using Thread 6
Thread Sorting started for file temp_in_00424.txt using Thread 4
Thread Sorting started for file temp_in_00425.txt using Thread 1
Thread Sorting started for file temp_in_00426.txt using Thread 3
Thread Sorting started for file temp_in_00427.txt using Thread 7
Thread Sorting started for file temp_in_00428.txt using Thread 0
Thread Sorting started for file temp_in_00429.txt using Thread 2
Thread Sorting started for file temp_in_00430.txt using Thread 5
Thread Sorting started for file temp_in_00431.txt using Thread 6
Thread Sorting started for file temp_in_00432.txt using Thread 4
Thread Sorting started for file temp_in_00433.txt using Thread 1
Thread Sorting started for file temp_in_00434.txt using Thread 3
Thread Sorting started for file temp_in_00435.txt using Thread 7
Thread Sorting started for file temp_in_00436.txt using Thread 0

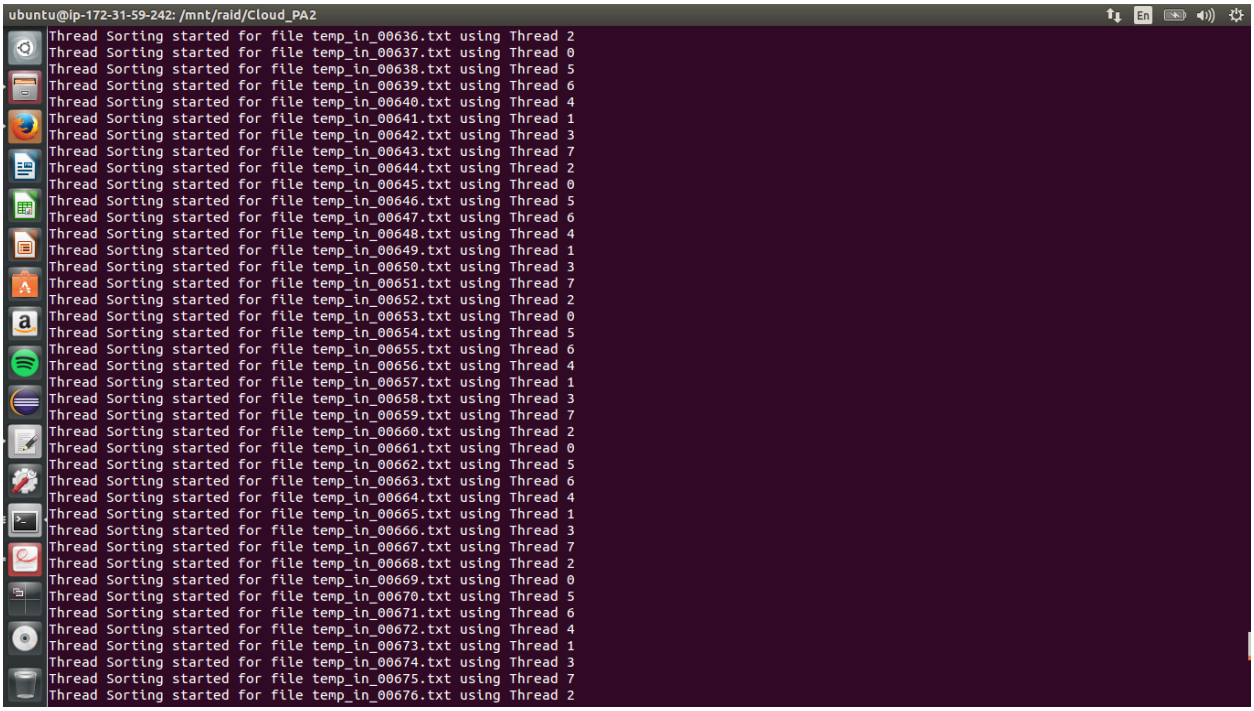

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00436.txt using Thread 0
Thread Sorting started for file temp_in_00437.txt using Thread 2
Thread Sorting started for file temp_in_00438.txt using Thread 5
Thread Sorting started for file temp_in_00439.txt using Thread 6
Thread Sorting started for file temp_in_00440.txt using Thread 4
Thread Sorting started for file temp_in_00441.txt using Thread 1
Thread Sorting started for file temp_in_00442.txt using Thread 3
Thread Sorting started for file temp_in_00443.txt using Thread 7
Thread Sorting started for file temp_in_00444.txt using Thread 0
Thread Sorting started for file temp_in_00445.txt using Thread 2
Thread Sorting started for file temp_in_00446.txt using Thread 5
Thread Sorting started for file temp_in_00447.txt using Thread 6
Thread Sorting started for file temp_in_00448.txt using Thread 4
Thread Sorting started for file temp_in_00449.txt using Thread 1
Thread Sorting started for file temp_in_00450.txt using Thread 3
Thread Sorting started for file temp_in_00451.txt using Thread 7
Thread Sorting started for file temp_in_00452.txt using Thread 0
Thread Sorting started for file temp_in_00453.txt using Thread 2
Thread Sorting started for file temp_in_00454.txt using Thread 5
Thread Sorting started for file temp_in_00455.txt using Thread 6
Thread Sorting started for file temp_in_00456.txt using Thread 4
Thread Sorting started for file temp_in_00457.txt using Thread 1
Thread Sorting started for file temp_in_00458.txt using Thread 3
Thread Sorting started for file temp_in_00459.txt using Thread 7
Thread Sorting started for file temp_in_00460.txt using Thread 0
Thread Sorting started for file temp_in_00461.txt using Thread 2
Thread Sorting started for file temp_in_00462.txt using Thread 5
Thread Sorting started for file temp_in_00463.txt using Thread 6
Thread Sorting started for file temp_in_00464.txt using Thread 4
Thread Sorting started for file temp_in_00465.txt using Thread 1
Thread Sorting started for file temp_in_00466.txt using Thread 3
Thread Sorting started for file temp_in_00467.txt using Thread 7
Thread Sorting started for file temp_in_00468.txt using Thread 0
Thread Sorting started for file temp_in_00469.txt using Thread 2
Thread Sorting started for file temp_in_00470.txt using Thread 5
Thread Sorting started for file temp_in_00471.txt using Thread 6
Thread Sorting started for file temp_in_00472.txt using Thread 4
Thread Sorting started for file temp_in_00473.txt using Thread 1
Thread Sorting started for file temp_in_00474.txt using Thread 3
Thread Sorting started for file temp_in_00475.txt using Thread 7
Thread Sorting started for file temp_in_00476.txt using Thread 0

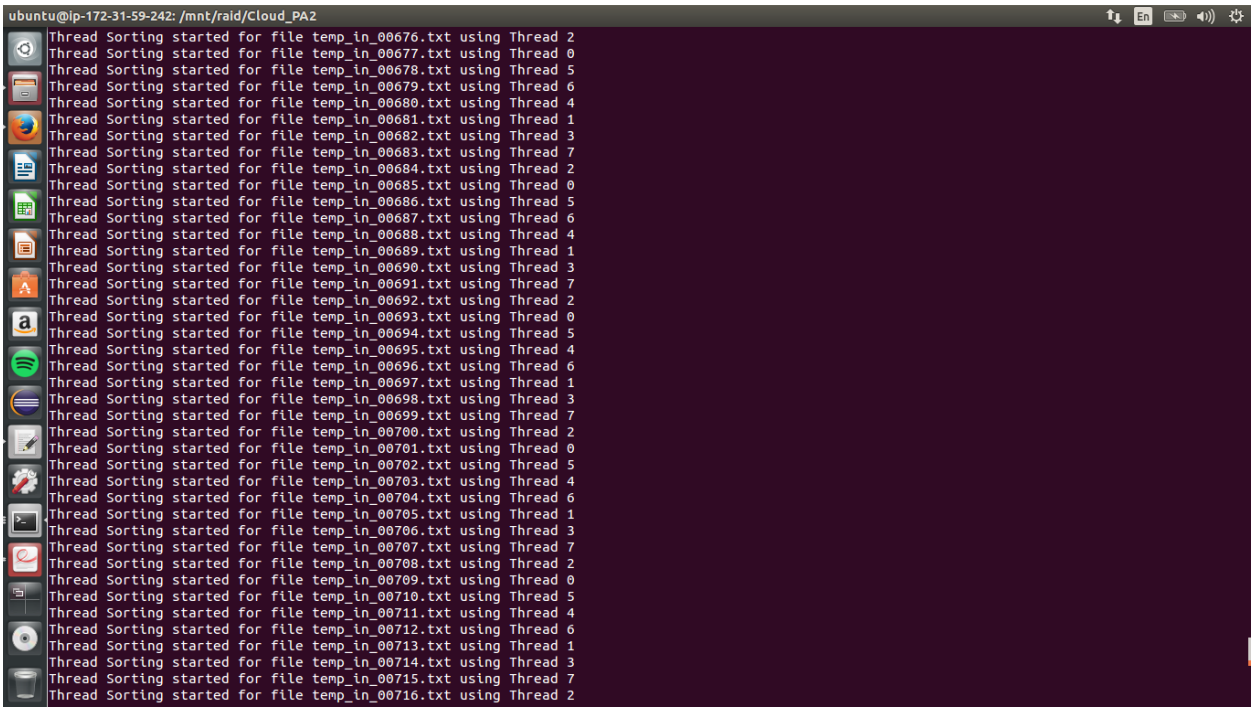

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00476.txt using Thread 0
Thread Sorting started for file temp_in_00477.txt using Thread 2
Thread Sorting started for file temp_in_00478.txt using Thread 5
Thread Sorting started for file temp_in_00479.txt using Thread 6
Thread Sorting started for file temp_in_00480.txt using Thread 4
Thread Sorting started for file temp_in_00481.txt using Thread 1
Thread Sorting started for file temp_in_00482.txt using Thread 3
Thread Sorting started for file temp_in_00483.txt using Thread 7
Thread Sorting started for file temp_in_00484.txt using Thread 0
Thread Sorting started for file temp_in_00485.txt using Thread 2
Thread Sorting started for file temp_in_00486.txt using Thread 5
Thread Sorting started for file temp_in_00487.txt using Thread 6
Thread Sorting started for file temp_in_00488.txt using Thread 4
Thread Sorting started for file temp_in_00489.txt using Thread 1
Thread Sorting started for file temp_in_00490.txt using Thread 3
Thread Sorting started for file temp_in_00491.txt using Thread 7
Thread Sorting started for file temp_in_00492.txt using Thread 0
Thread Sorting started for file temp_in_00493.txt using Thread 2
Thread Sorting started for file temp_in_00494.txt using Thread 5
Thread Sorting started for file temp_in_00495.txt using Thread 6
Thread Sorting started for file temp_in_00496.txt using Thread 4
Thread Sorting started for file temp_in_00497.txt using Thread 1
Thread Sorting started for file temp_in_00498.txt using Thread 3
Thread Sorting started for file temp_in_00499.txt using Thread 7
Thread Sorting started for file temp_in_00500.txt using Thread 2
Thread Sorting started for file temp_in_00501.txt using Thread 0
Thread Sorting started for file temp_in_00502.txt using Thread 5
Thread Sorting started for file temp_in_00503.txt using Thread 6
Thread Sorting started for file temp_in_00504.txt using Thread 4
Thread Sorting started for file temp_in_00505.txt using Thread 1
Thread Sorting started for file temp_in_00506.txt using Thread 3
Thread Sorting started for file temp_in_00507.txt using Thread 7
Thread Sorting started for file temp_in_00508.txt using Thread 2
Thread Sorting started for file temp_in_00509.txt using Thread 0
Thread Sorting started for file temp_in_00510.txt using Thread 5
Thread Sorting started for file temp_in_00511.txt using Thread 6
Thread Sorting started for file temp_in_00512.txt using Thread 4
Thread Sorting started for file temp_in_00513.txt using Thread 1
Thread Sorting started for file temp_in_00514.txt using Thread 3
Thread Sorting started for file temp_in_00515.txt using Thread 7
Thread Sorting started for file temp_in_00516.txt using Thread 2

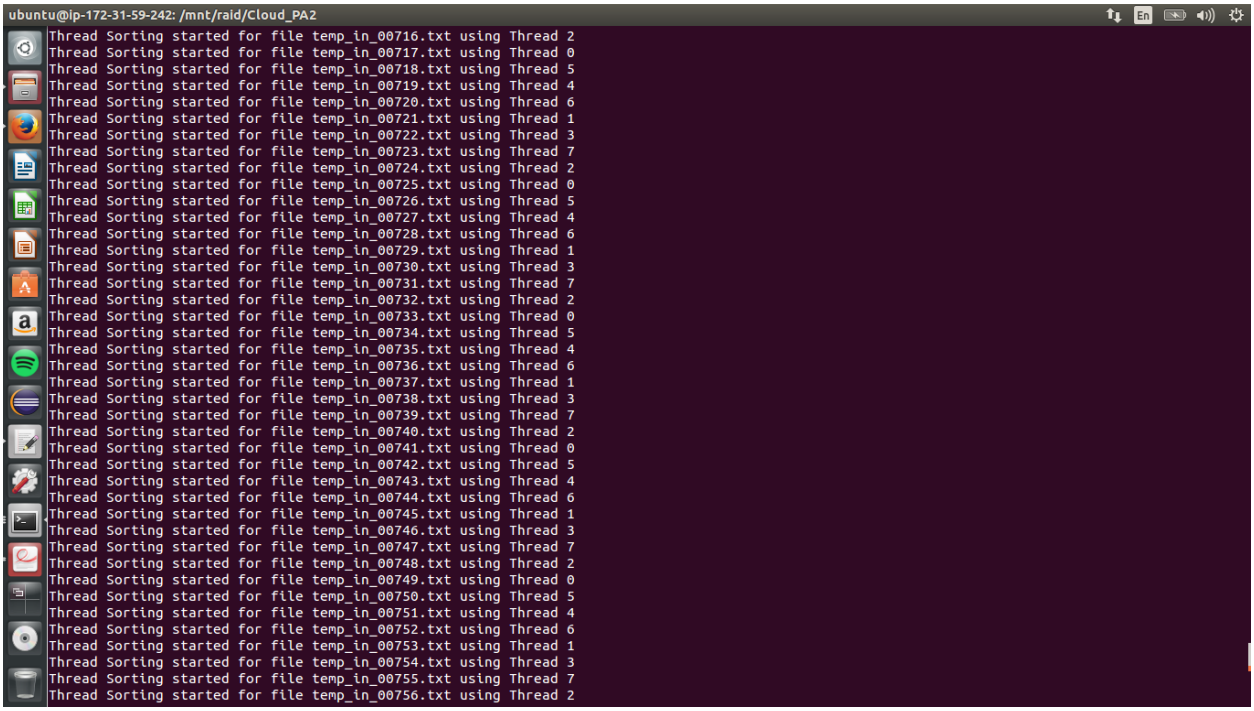

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00516.txt using Thread 2
Thread Sorting started for file temp_in_00517.txt using Thread 0
Thread Sorting started for file temp_in_00518.txt using Thread 5
Thread Sorting started for file temp_in_00519.txt using Thread 6
Thread Sorting started for file temp_in_00520.txt using Thread 4
Thread Sorting started for file temp_in_00521.txt using Thread 1
Thread Sorting started for file temp_in_00522.txt using Thread 3
Thread Sorting started for file temp_in_00523.txt using Thread 7
Thread Sorting started for file temp_in_00524.txt using Thread 2
Thread Sorting started for file temp_in_00525.txt using Thread 0
Thread Sorting started for file temp_in_00526.txt using Thread 5
Thread Sorting started for file temp_in_00527.txt using Thread 6
Thread Sorting started for file temp_in_00528.txt using Thread 4
Thread Sorting started for file temp_in_00529.txt using Thread 1
Thread Sorting started for file temp_in_00530.txt using Thread 3
Thread Sorting started for file temp_in_00531.txt using Thread 7
Thread Sorting started for file temp_in_00532.txt using Thread 2
Thread Sorting started for file temp_in_00533.txt using Thread 0
Thread Sorting started for file temp_in_00534.txt using Thread 5
Thread Sorting started for file temp_in_00535.txt using Thread 6
Thread Sorting started for file temp_in_00536.txt using Thread 4
Thread Sorting started for file temp_in_00537.txt using Thread 1
Thread Sorting started for file temp_in_00538.txt using Thread 3
Thread Sorting started for file temp_in_00539.txt using Thread 7
Thread Sorting started for file temp_in_00540.txt using Thread 2
Thread Sorting started for file temp_in_00541.txt using Thread 0
Thread Sorting started for file temp_in_00542.txt using Thread 5
Thread Sorting started for file temp_in_00543.txt using Thread 6
Thread Sorting started for file temp_in_00544.txt using Thread 4
Thread Sorting started for file temp_in_00545.txt using Thread 1
Thread Sorting started for file temp_in_00546.txt using Thread 3
Thread Sorting started for file temp_in_00547.txt using Thread 7
Thread Sorting started for file temp_in_00548.txt using Thread 2
Thread Sorting started for file temp_in_00549.txt using Thread 0
Thread Sorting started for file temp_in_00550.txt using Thread 5
Thread Sorting started for file temp_in_00551.txt using Thread 6
Thread Sorting started for file temp_in_00552.txt using Thread 4
Thread Sorting started for file temp_in_00553.txt using Thread 1
Thread Sorting started for file temp_in_00554.txt using Thread 3
Thread Sorting started for file temp_in_00555.txt using Thread 7
Thread Sorting started for file temp_in_00556.txt using Thread 2

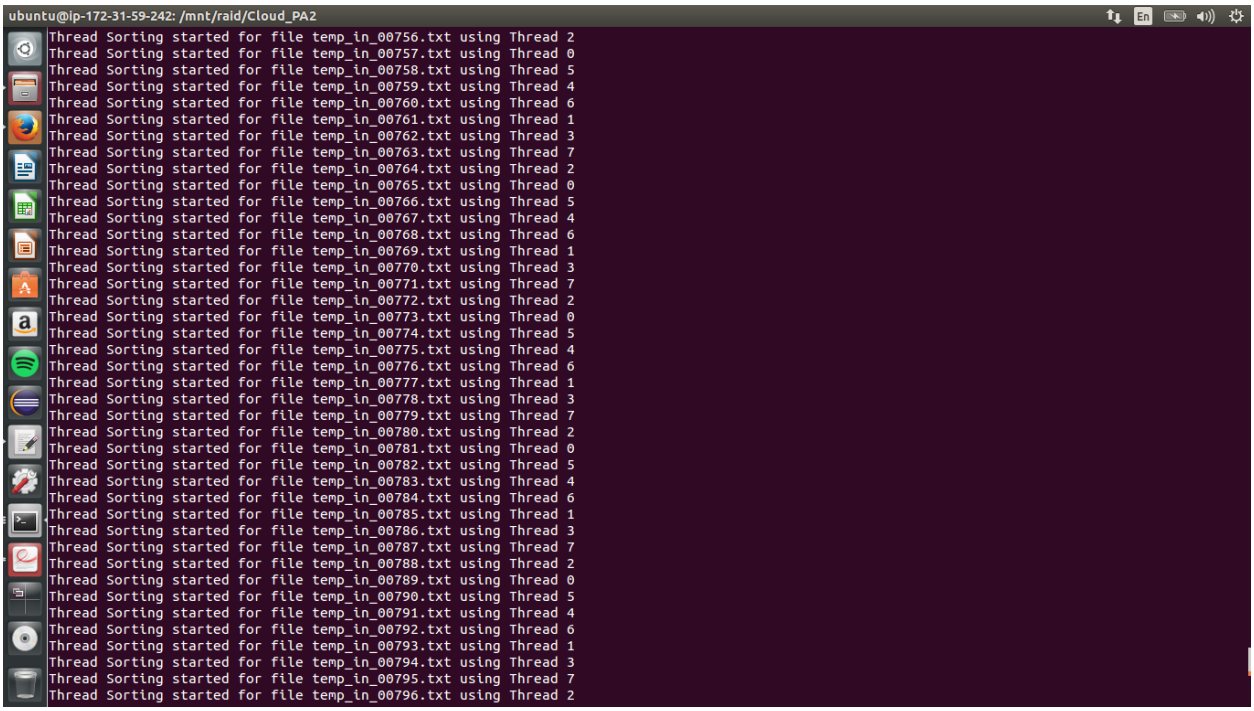

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00556.txt using Thread 2
Thread Sorting started for file temp_in_00557.txt using Thread 0
Thread Sorting started for file temp_in_00558.txt using Thread 5
Thread Sorting started for file temp_in_00559.txt using Thread 6
Thread Sorting started for file temp_in_00560.txt using Thread 4
Thread Sorting started for file temp_in_00561.txt using Thread 1
Thread Sorting started for file temp_in_00562.txt using Thread 3
Thread Sorting started for file temp_in_00563.txt using Thread 7
Thread Sorting started for file temp_in_00564.txt using Thread 2
Thread Sorting started for file temp_in_00565.txt using Thread 0
Thread Sorting started for file temp_in_00566.txt using Thread 5
Thread Sorting started for file temp_in_00567.txt using Thread 6
Thread Sorting started for file temp_in_00568.txt using Thread 4
Thread Sorting started for file temp_in_00569.txt using Thread 1
Thread Sorting started for file temp_in_00570.txt using Thread 3
Thread Sorting started for file temp_in_00571.txt using Thread 7
Thread Sorting started for file temp_in_00572.txt using Thread 2
Thread Sorting started for file temp_in_00573.txt using Thread 5
Thread Sorting started for file temp_in_00574.txt using Thread 0
Thread Sorting started for file temp_in_00575.txt using Thread 6
Thread Sorting started for file temp_in_00576.txt using Thread 4
Thread Sorting started for file temp_in_00577.txt using Thread 1
Thread Sorting started for file temp_in_00578.txt using Thread 3
Thread Sorting started for file temp_in_00579.txt using Thread 7
Thread Sorting started for file temp_in_00580.txt using Thread 2
Thread Sorting started for file temp_in_00581.txt using Thread 5
Thread Sorting started for file temp_in_00582.txt using Thread 0
Thread Sorting started for file temp_in_00583.txt using Thread 6
Thread Sorting started for file temp_in_00584.txt using Thread 4
Thread Sorting started for file temp_in_00585.txt using Thread 1
Thread Sorting started for file temp_in_00586.txt using Thread 3
Thread Sorting started for file temp_in_00587.txt using Thread 7
Thread Sorting started for file temp_in_00588.txt using Thread 2
Thread Sorting started for file temp_in_00589.txt using Thread 5
Thread Sorting started for file temp_in_00590.txt using Thread 0
Thread Sorting started for file temp_in_00591.txt using Thread 6
Thread Sorting started for file temp_in_00592.txt using Thread 4
Thread Sorting started for file temp_in_00593.txt using Thread 1
Thread Sorting started for file temp_in_00594.txt using Thread 3
Thread Sorting started for file temp_in_00595.txt using Thread 7
Thread Sorting started for file temp_in_00596.txt using Thread 2

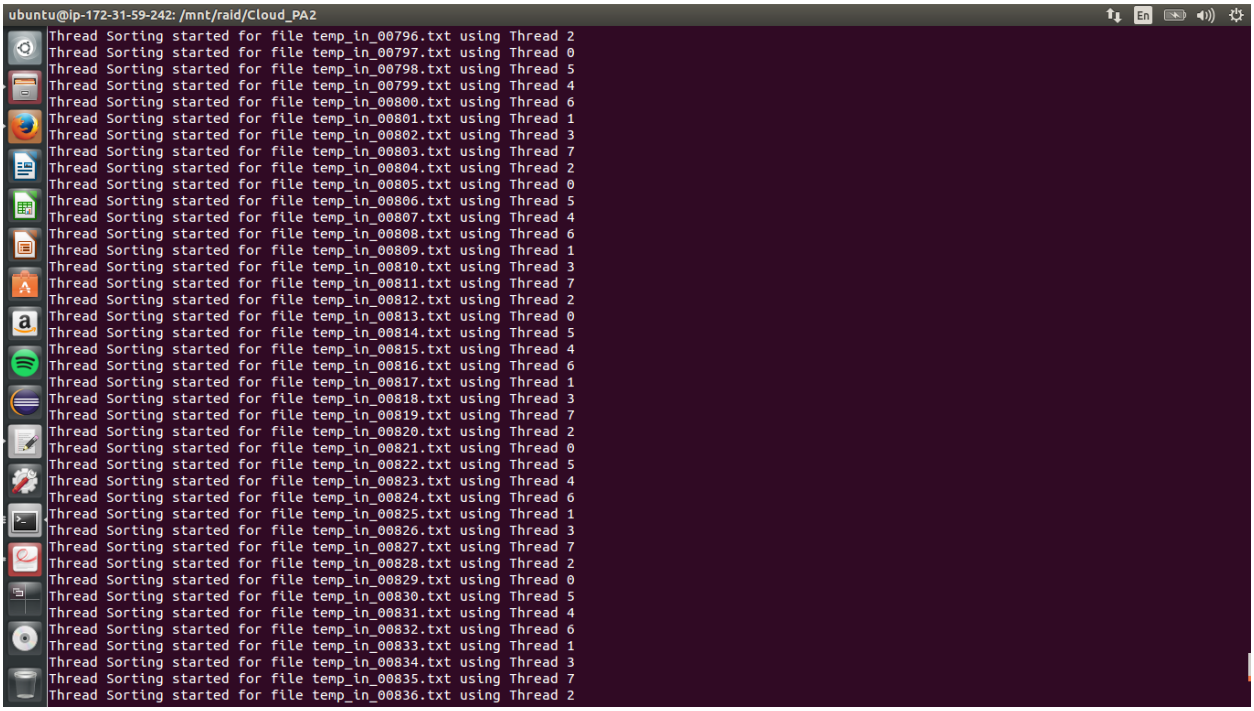

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00596.txt using Thread 2
Thread Sorting started for file temp_in_00597.txt using Thread 0
Thread Sorting started for file temp_in_00598.txt using Thread 5
Thread Sorting started for file temp_in_00599.txt using Thread 6
Thread Sorting started for file temp_in_00600.txt using Thread 4
Thread Sorting started for file temp_in_00601.txt using Thread 1
Thread Sorting started for file temp_in_00602.txt using Thread 3
Thread Sorting started for file temp_in_00603.txt using Thread 7
Thread Sorting started for file temp_in_00604.txt using Thread 2
Thread Sorting started for file temp_in_00605.txt using Thread 0
Thread Sorting started for file temp_in_00606.txt using Thread 5
Thread Sorting started for file temp_in_00607.txt using Thread 6
Thread Sorting started for file temp_in_00608.txt using Thread 4
Thread Sorting started for file temp_in_00609.txt using Thread 1
Thread Sorting started for file temp_in_00610.txt using Thread 3
Thread Sorting started for file temp_in_00611.txt using Thread 7
Thread Sorting started for file temp_in_00612.txt using Thread 2
Thread Sorting started for file temp_in_00613.txt using Thread 0
Thread Sorting started for file temp_in_00614.txt using Thread 5
Thread Sorting started for file temp_in_00615.txt using Thread 6
Thread Sorting started for file temp_in_00616.txt using Thread 4
Thread Sorting started for file temp_in_00617.txt using Thread 1
Thread Sorting started for file temp_in_00618.txt using Thread 3
Thread Sorting started for file temp_in_00619.txt using Thread 7
Thread Sorting started for file temp_in_00620.txt using Thread 2
Thread Sorting started for file temp_in_00621.txt using Thread 0
Thread Sorting started for file temp_in_00622.txt using Thread 5
Thread Sorting started for file temp_in_00623.txt using Thread 6
Thread Sorting started for file temp_in_00624.txt using Thread 4
Thread Sorting started for file temp_in_00625.txt using Thread 1
Thread Sorting started for file temp_in_00626.txt using Thread 3
Thread Sorting started for file temp_in_00627.txt using Thread 7
Thread Sorting started for file temp_in_00628.txt using Thread 2
Thread Sorting started for file temp_in_00629.txt using Thread 0
Thread Sorting started for file temp_in_00630.txt using Thread 5
Thread Sorting started for file temp_in_00631.txt using Thread 6
Thread Sorting started for file temp_in_00632.txt using Thread 4
Thread Sorting started for file temp_in_00633.txt using Thread 1
Thread Sorting started for file temp_in_00634.txt using Thread 3
Thread Sorting started for file temp_in_00635.txt using Thread 7
Thread Sorting started for file temp_in_00636.txt using Thread 2

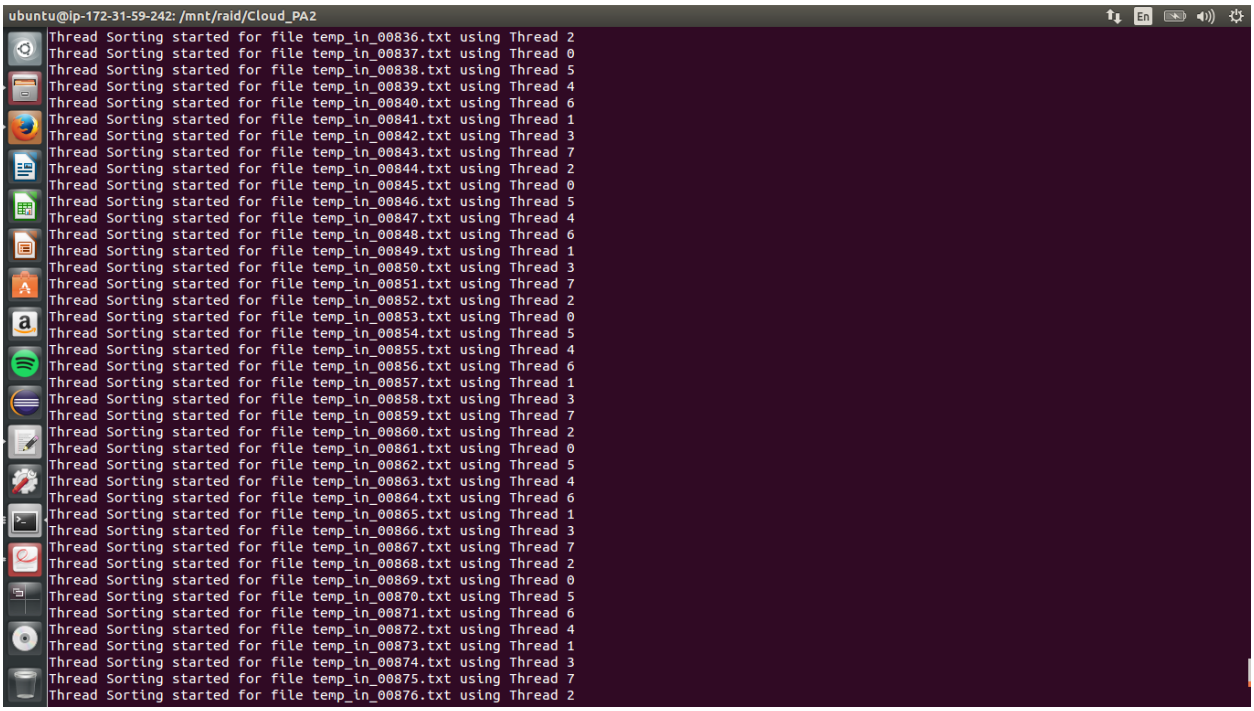

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00636.txt using Thread 2
Thread Sorting started for file temp_in_00637.txt using Thread 0
Thread Sorting started for file temp_in_00638.txt using Thread 5
Thread Sorting started for file temp_in_00639.txt using Thread 6
Thread Sorting started for file temp_in_00640.txt using Thread 4
Thread Sorting started for file temp_in_00641.txt using Thread 1
Thread Sorting started for file temp_in_00642.txt using Thread 3
Thread Sorting started for file temp_in_00643.txt using Thread 7
Thread Sorting started for file temp_in_00644.txt using Thread 2
Thread Sorting started for file temp_in_00645.txt using Thread 0
Thread Sorting started for file temp_in_00646.txt using Thread 5
Thread Sorting started for file temp_in_00647.txt using Thread 6
Thread Sorting started for file temp_in_00648.txt using Thread 4
Thread Sorting started for file temp_in_00649.txt using Thread 1
Thread Sorting started for file temp_in_00650.txt using Thread 3
Thread Sorting started for file temp_in_00651.txt using Thread 7
Thread Sorting started for file temp_in_00652.txt using Thread 2
Thread Sorting started for file temp_in_00653.txt using Thread 0
Thread Sorting started for file temp_in_00654.txt using Thread 5
Thread Sorting started for file temp_in_00655.txt using Thread 6
Thread Sorting started for file temp_in_00656.txt using Thread 4
Thread Sorting started for file temp_in_00657.txt using Thread 1
Thread Sorting started for file temp_in_00658.txt using Thread 3
Thread Sorting started for file temp_in_00659.txt using Thread 7
Thread Sorting started for file temp_in_00660.txt using Thread 2
Thread Sorting started for file temp_in_00661.txt using Thread 0
Thread Sorting started for file temp_in_00662.txt using Thread 5
Thread Sorting started for file temp_in_00663.txt using Thread 6
Thread Sorting started for file temp_in_00664.txt using Thread 4
Thread Sorting started for file temp_in_00665.txt using Thread 1
Thread Sorting started for file temp_in_00666.txt using Thread 3
Thread Sorting started for file temp_in_00667.txt using Thread 7
Thread Sorting started for file temp_in_00668.txt using Thread 2
Thread Sorting started for file temp_in_00669.txt using Thread 0
Thread Sorting started for file temp_in_00670.txt using Thread 5
Thread Sorting started for file temp_in_00671.txt using Thread 6
Thread Sorting started for file temp_in_00672.txt using Thread 4
Thread Sorting started for file temp_in_00673.txt using Thread 1
Thread Sorting started for file temp_in_00674.txt using Thread 3
Thread Sorting started for file temp_in_00675.txt using Thread 7
Thread Sorting started for file temp_in_00676.txt using Thread 2

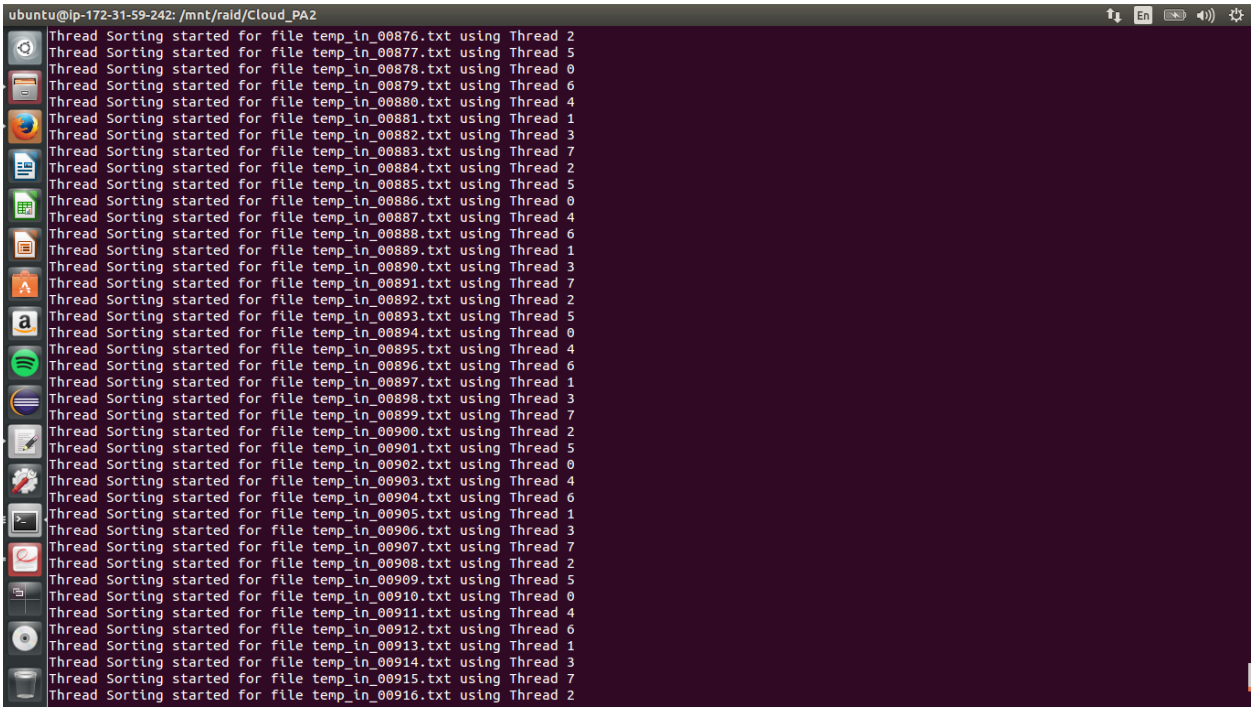

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00676.txt using Thread 2
Thread Sorting started for file temp_in_00677.txt using Thread 0
Thread Sorting started for file temp_in_00678.txt using Thread 5
Thread Sorting started for file temp_in_00679.txt using Thread 6
Thread Sorting started for file temp_in_00680.txt using Thread 4
Thread Sorting started for file temp_in_00681.txt using Thread 1
Thread Sorting started for file temp_in_00682.txt using Thread 3
Thread Sorting started for file temp_in_00683.txt using Thread 7
Thread Sorting started for file temp_in_00684.txt using Thread 2
Thread Sorting started for file temp_in_00685.txt using Thread 0
Thread Sorting started for file temp_in_00686.txt using Thread 5
Thread Sorting started for file temp_in_00687.txt using Thread 6
Thread Sorting started for file temp_in_00688.txt using Thread 4
Thread Sorting started for file temp_in_00689.txt using Thread 1
Thread Sorting started for file temp_in_00690.txt using Thread 3
Thread Sorting started for file temp_in_00691.txt using Thread 7
Thread Sorting started for file temp_in_00692.txt using Thread 2
Thread Sorting started for file temp_in_00693.txt using Thread 0
Thread Sorting started for file temp_in_00694.txt using Thread 5
Thread Sorting started for file temp_in_00695.txt using Thread 4
Thread Sorting started for file temp_in_00696.txt using Thread 6
Thread Sorting started for file temp_in_00697.txt using Thread 1
Thread Sorting started for file temp_in_00698.txt using Thread 3
Thread Sorting started for file temp_in_00699.txt using Thread 7
Thread Sorting started for file temp_in_00700.txt using Thread 2
Thread Sorting started for file temp_in_00701.txt using Thread 0
Thread Sorting started for file temp_in_00702.txt using Thread 5
Thread Sorting started for file temp_in_00703.txt using Thread 4
Thread Sorting started for file temp_in_00704.txt using Thread 6
Thread Sorting started for file temp_in_00705.txt using Thread 1
Thread Sorting started for file temp_in_00706.txt using Thread 3
Thread Sorting started for file temp_in_00707.txt using Thread 7
Thread Sorting started for file temp_in_00708.txt using Thread 2
Thread Sorting started for file temp_in_00709.txt using Thread 0
Thread Sorting started for file temp_in_00710.txt using Thread 5
Thread Sorting started for file temp_in_00711.txt using Thread 4
Thread Sorting started for file temp_in_00712.txt using Thread 6
Thread Sorting started for file temp_in_00713.txt using Thread 1
Thread Sorting started for file temp_in_00714.txt using Thread 3
Thread Sorting started for file temp_in_00715.txt using Thread 7
Thread Sorting started for file temp_in_00716.txt using Thread 2

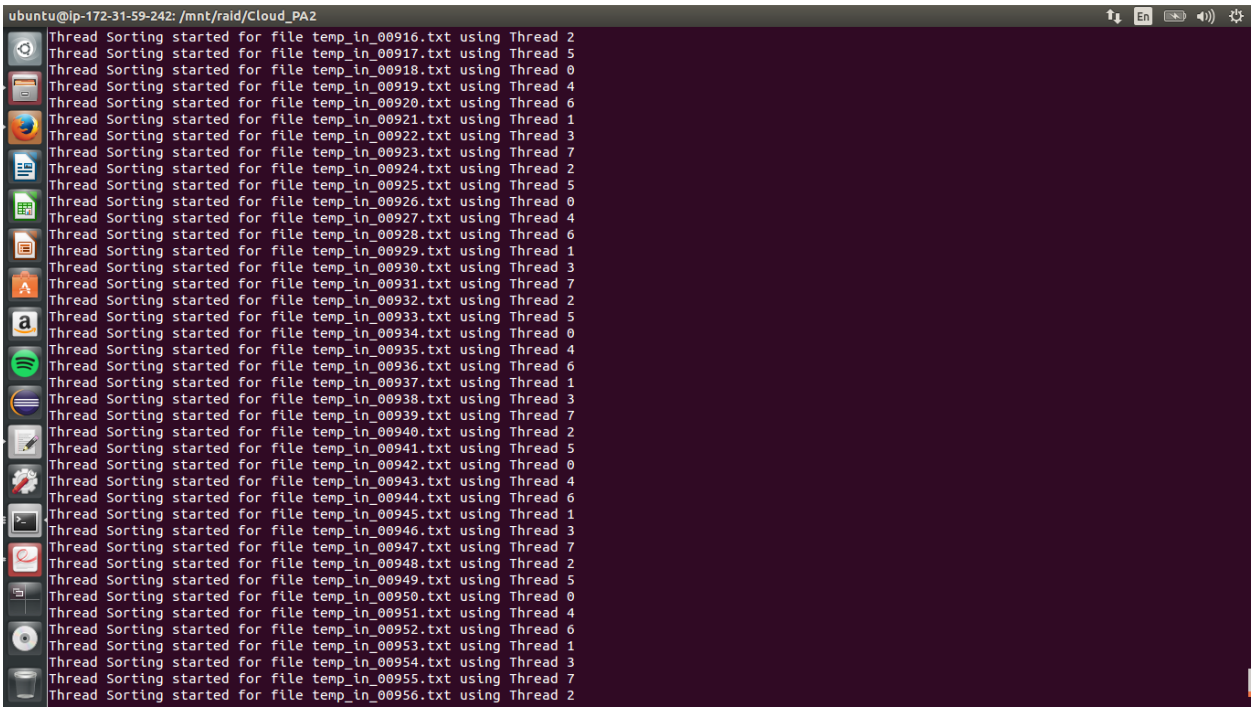

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00716.txt using Thread 2
Thread Sorting started for file temp_in_00717.txt using Thread 0
Thread Sorting started for file temp_in_00718.txt using Thread 5
Thread Sorting started for file temp_in_00719.txt using Thread 4
Thread Sorting started for file temp_in_00720.txt using Thread 6
Thread Sorting started for file temp_in_00721.txt using Thread 1
Thread Sorting started for file temp_in_00722.txt using Thread 3
Thread Sorting started for file temp_in_00723.txt using Thread 7
Thread Sorting started for file temp_in_00724.txt using Thread 2
Thread Sorting started for file temp_in_00725.txt using Thread 0
Thread Sorting started for file temp_in_00726.txt using Thread 5
Thread Sorting started for file temp_in_00727.txt using Thread 4
Thread Sorting started for file temp_in_00728.txt using Thread 6
Thread Sorting started for file temp_in_00729.txt using Thread 1
Thread Sorting started for file temp_in_00730.txt using Thread 3
Thread Sorting started for file temp_in_00731.txt using Thread 7
Thread Sorting started for file temp_in_00732.txt using Thread 2
Thread Sorting started for file temp_in_00733.txt using Thread 0
Thread Sorting started for file temp_in_00734.txt using Thread 5
Thread Sorting started for file temp_in_00735.txt using Thread 4
Thread Sorting started for file temp_in_00736.txt using Thread 6
Thread Sorting started for file temp_in_00737.txt using Thread 1
Thread Sorting started for file temp_in_00738.txt using Thread 3
Thread Sorting started for file temp_in_00739.txt using Thread 7
Thread Sorting started for file temp_in_00740.txt using Thread 2
Thread Sorting started for file temp_in_00741.txt using Thread 0
Thread Sorting started for file temp_in_00742.txt using Thread 5
Thread Sorting started for file temp_in_00743.txt using Thread 4
Thread Sorting started for file temp_in_00744.txt using Thread 6
Thread Sorting started for file temp_in_00745.txt using Thread 1
Thread Sorting started for file temp_in_00746.txt using Thread 3
Thread Sorting started for file temp_in_00747.txt using Thread 7
Thread Sorting started for file temp_in_00748.txt using Thread 2
Thread Sorting started for file temp_in_00749.txt using Thread 0
Thread Sorting started for file temp_in_00750.txt using Thread 5
Thread Sorting started for file temp_in_00751.txt using Thread 4
Thread Sorting started for file temp_in_00752.txt using Thread 6
Thread Sorting started for file temp_in_00753.txt using Thread 1
Thread Sorting started for file temp_in_00754.txt using Thread 3
Thread Sorting started for file temp_in_00755.txt using Thread 7
Thread Sorting started for file temp_in_00756.txt using Thread 2

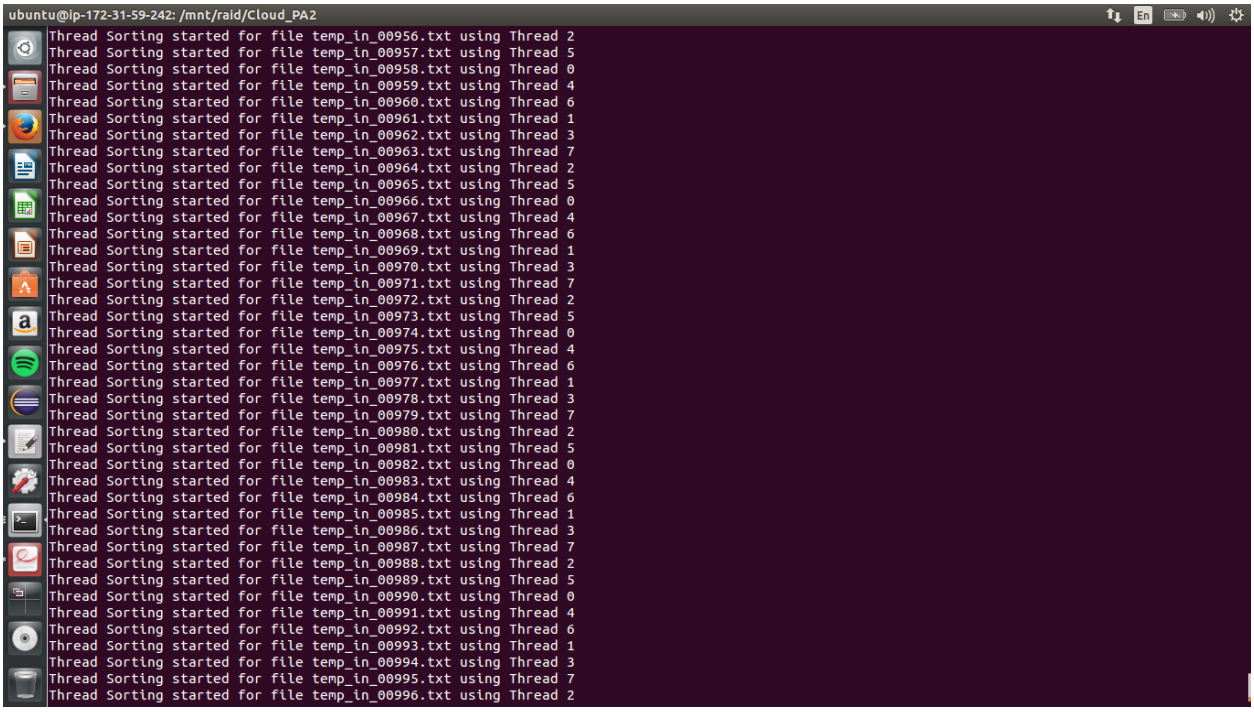

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00756.txt using Thread 2
Thread Sorting started for file temp_in_00757.txt using Thread 0
Thread Sorting started for file temp_in_00758.txt using Thread 5
Thread Sorting started for file temp_in_00759.txt using Thread 4
Thread Sorting started for file temp_in_00760.txt using Thread 6
Thread Sorting started for file temp_in_00761.txt using Thread 1
Thread Sorting started for file temp_in_00762.txt using Thread 3
Thread Sorting started for file temp_in_00763.txt using Thread 7
Thread Sorting started for file temp_in_00764.txt using Thread 2
Thread Sorting started for file temp_in_00765.txt using Thread 0
Thread Sorting started for file temp_in_00766.txt using Thread 5
Thread Sorting started for file temp_in_00767.txt using Thread 4
Thread Sorting started for file temp_in_00768.txt using Thread 6
Thread Sorting started for file temp_in_00769.txt using Thread 1
Thread Sorting started for file temp_in_00770.txt using Thread 3
Thread Sorting started for file temp_in_00771.txt using Thread 7
Thread Sorting started for file temp_in_00772.txt using Thread 2
Thread Sorting started for file temp_in_00773.txt using Thread 0
Thread Sorting started for file temp_in_00774.txt using Thread 5
Thread Sorting started for file temp_in_00775.txt using Thread 4
Thread Sorting started for file temp_in_00776.txt using Thread 6
Thread Sorting started for file temp_in_00777.txt using Thread 1
Thread Sorting started for file temp_in_00778.txt using Thread 3
Thread Sorting started for file temp_in_00779.txt using Thread 7
Thread Sorting started for file temp_in_00780.txt using Thread 2
Thread Sorting started for file temp_in_00781.txt using Thread 0
Thread Sorting started for file temp_in_00782.txt using Thread 5
Thread Sorting started for file temp_in_00783.txt using Thread 4
Thread Sorting started for file temp_in_00784.txt using Thread 6
Thread Sorting started for file temp_in_00785.txt using Thread 1
Thread Sorting started for file temp_in_00786.txt using Thread 3
Thread Sorting started for file temp_in_00787.txt using Thread 7
Thread Sorting started for file temp_in_00788.txt using Thread 2
Thread Sorting started for file temp_in_00789.txt using Thread 0
Thread Sorting started for file temp_in_00790.txt using Thread 5
Thread Sorting started for file temp_in_00791.txt using Thread 4
Thread Sorting started for file temp_in_00792.txt using Thread 6
Thread Sorting started for file temp_in_00793.txt using Thread 1
Thread Sorting started for file temp_in_00794.txt using Thread 3
Thread Sorting started for file temp_in_00795.txt using Thread 7
Thread Sorting started for file temp_in_00796.txt using Thread 2

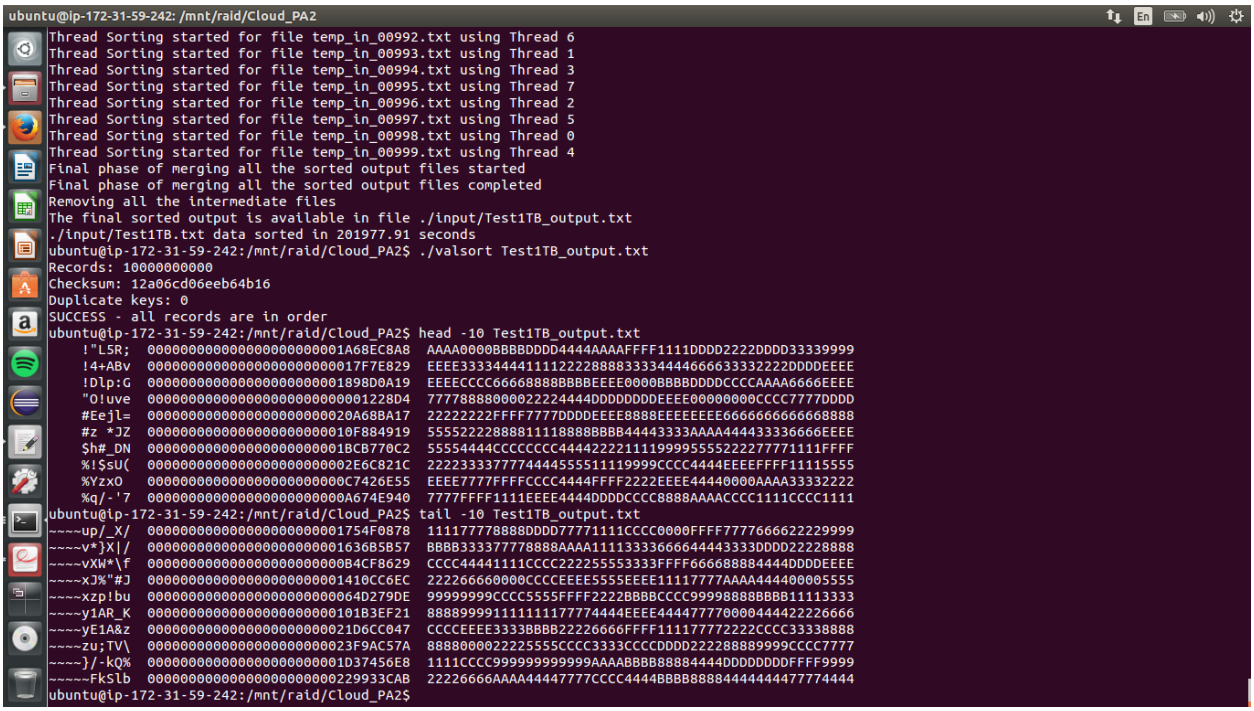

ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00796.txt using Thread 2
Thread Sorting started for file temp_in_00797.txt using Thread 0
Thread Sorting started for file temp_in_00798.txt using Thread 5
Thread Sorting started for file temp_in_00799.txt using Thread 4
Thread Sorting started for file temp_in_00800.txt using Thread 6
Thread Sorting started for file temp_in_00801.txt using Thread 1
Thread Sorting started for file temp_in_00802.txt using Thread 3
Thread Sorting started for file temp_in_00803.txt using Thread 7
Thread Sorting started for file temp_in_00804.txt using Thread 2
Thread Sorting started for file temp_in_00805.txt using Thread 0
Thread Sorting started for file temp_in_00806.txt using Thread 5
Thread Sorting started for file temp_in_00807.txt using Thread 4
Thread Sorting started for file temp_in_00808.txt using Thread 6
Thread Sorting started for file temp_in_00809.txt using Thread 1
Thread Sorting started for file temp_in_00810.txt using Thread 3
Thread Sorting started for file temp_in_00811.txt using Thread 7
Thread Sorting started for file temp_in_00812.txt using Thread 2
Thread Sorting started for file temp_in_00813.txt using Thread 0
Thread Sorting started for file temp_in_00814.txt using Thread 5
Thread Sorting started for file temp_in_00815.txt using Thread 4
Thread Sorting started for file temp_in_00816.txt using Thread 6
Thread Sorting started for file temp_in_00817.txt using Thread 1
Thread Sorting started for file temp_in_00818.txt using Thread 3
Thread Sorting started for file temp_in_00819.txt using Thread 7
Thread Sorting started for file temp_in_00820.txt using Thread 2
Thread Sorting started for file temp_in_00821.txt using Thread 0
Thread Sorting started for file temp_in_00822.txt using Thread 5
Thread Sorting started for file temp_in_00823.txt using Thread 4
Thread Sorting started for file temp_in_00824.txt using Thread 6
Thread Sorting started for file temp_in_00825.txt using Thread 1
Thread Sorting started for file temp_in_00826.txt using Thread 3
Thread Sorting started for file temp_in_00827.txt using Thread 7
Thread Sorting started for file temp_in_00828.txt using Thread 2
Thread Sorting started for file temp_in_00829.txt using Thread 0
Thread Sorting started for file temp_in_00830.txt using Thread 5
Thread Sorting started for file temp_in_00831.txt using Thread 4
Thread Sorting started for file temp_in_00832.txt using Thread 6
Thread Sorting started for file temp_in_00833.txt using Thread 1
Thread Sorting started for file temp_in_00834.txt using Thread 3
Thread Sorting started for file temp_in_00835.txt using Thread 7
Thread Sorting started for file temp_in_00836.txt using Thread 2


ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00836.txt using Thread 2
Thread Sorting started for file temp_in_00837.txt using Thread 0
Thread Sorting started for file temp_in_00838.txt using Thread 5
Thread Sorting started for file temp_in_00839.txt using Thread 4
Thread Sorting started for file temp_in_00840.txt using Thread 6
Thread Sorting started for file temp_in_00841.txt using Thread 1
Thread Sorting started for file temp_in_00842.txt using Thread 3
Thread Sorting started for file temp_in_00843.txt using Thread 7
Thread Sorting started for file temp_in_00844.txt using Thread 2
Thread Sorting started for file temp_in_00845.txt using Thread 0
Thread Sorting started for file temp_in_00846.txt using Thread 5
Thread Sorting started for file temp_in_00847.txt using Thread 4
Thread Sorting started for file temp_in_00848.txt using Thread 6
Thread Sorting started for file temp_in_00849.txt using Thread 1
Thread Sorting started for file temp_in_00850.txt using Thread 3
Thread Sorting started for file temp_in_00851.txt using Thread 7
Thread Sorting started for file temp_in_00852.txt using Thread 2
Thread Sorting started for file temp_in_00853.txt using Thread 0
Thread Sorting started for file temp_in_00854.txt using Thread 5
Thread Sorting started for file temp_in_00855.txt using Thread 4
Thread Sorting started for file temp_in_00856.txt using Thread 6
Thread Sorting started for file temp_in_00857.txt using Thread 1
Thread Sorting started for file temp_in_00858.txt using Thread 3
Thread Sorting started for file temp_in_00859.txt using Thread 7
Thread Sorting started for file temp_in_00860.txt using Thread 2
Thread Sorting started for file temp_in_00861.txt using Thread 0
Thread Sorting started for file temp_in_00862.txt using Thread 5
Thread Sorting started for file temp_in_00863.txt using Thread 4
Thread Sorting started for file temp_in_00864.txt using Thread 6
Thread Sorting started for file temp_in_00865.txt using Thread 1
Thread Sorting started for file temp_in_00866.txt using Thread 3
Thread Sorting started for file temp_in_00867.txt using Thread 7
Thread Sorting started for file temp_in_00868.txt using Thread 2
Thread Sorting started for file temp_in_00869.txt using Thread 0
Thread Sorting started for file temp_in_00870.txt using Thread 5
Thread Sorting started for file temp_in_00871.txt using Thread 6
Thread Sorting started for file temp_in_00872.txt using Thread 4
Thread Sorting started for file temp_in_00873.txt using Thread 1
Thread Sorting started for file temp_in_00874.txt using Thread 3
Thread Sorting started for file temp_in_00875.txt using Thread 7
Thread Sorting started for file temp_in_00876.txt using Thread 2


ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00876.txt using Thread 2
Thread Sorting started for file temp_in_00877.txt using Thread 5
Thread Sorting started for file temp_in_00878.txt using Thread 0
Thread Sorting started for file temp_in_00879.txt using Thread 6
Thread Sorting started for file temp_in_00880.txt using Thread 4
Thread Sorting started for file temp_in_00881.txt using Thread 1
Thread Sorting started for file temp_in_00882.txt using Thread 3
Thread Sorting started for file temp_in_00883.txt using Thread 7
Thread Sorting started for file temp_in_00884.txt using Thread 2
Thread Sorting started for file temp_in_00885.txt using Thread 5
Thread Sorting started for file temp_in_00886.txt using Thread 0
Thread Sorting started for file temp_in_00887.txt using Thread 4
Thread Sorting started for file temp_in_00888.txt using Thread 6
Thread Sorting started for file temp_in_00889.txt using Thread 1
Thread Sorting started for file temp_in_00890.txt using Thread 3
Thread Sorting started for file temp_in_00891.txt using Thread 7
Thread Sorting started for file temp_in_00892.txt using Thread 2
Thread Sorting started for file temp_in_00893.txt using Thread 5
Thread Sorting started for file temp_in_00894.txt using Thread 0
Thread Sorting started for file temp_in_00895.txt using Thread 4
Thread Sorting started for file temp_in_00896.txt using Thread 6
Thread Sorting started for file temp_in_00897.txt using Thread 1
Thread Sorting started for file temp_in_00898.txt using Thread 3
Thread Sorting started for file temp_in_00899.txt using Thread 7
Thread Sorting started for file temp_in_00900.txt using Thread 2
Thread Sorting started for file temp_in_00901.txt using Thread 5
Thread Sorting started for file temp_in_00902.txt using Thread 0
Thread Sorting started for file temp_in_00903.txt using Thread 4
Thread Sorting started for file temp_in_00904.txt using Thread 6
Thread Sorting started for file temp_in_00905.txt using Thread 1
Thread Sorting started for file temp_in_00906.txt using Thread 3
Thread Sorting started for file temp_in_00907.txt using Thread 7
Thread Sorting started for file temp_in_00908.txt using Thread 2
Thread Sorting started for file temp_in_00909.txt using Thread 5
Thread Sorting started for file temp_in_00910.txt using Thread 0
Thread Sorting started for file temp_in_00911.txt using Thread 4
Thread Sorting started for file temp_in_00912.txt using Thread 6
Thread Sorting started for file temp_in_00913.txt using Thread 1
Thread Sorting started for file temp_in_00914.txt using Thread 3
Thread Sorting started for file temp_in_00915.txt using Thread 7
Thread Sorting started for file temp_in_00916.txt using Thread 2


ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00916.txt using Thread 2
Thread Sorting started for file temp_in_00917.txt using Thread 5
Thread Sorting started for file temp_in_00918.txt using Thread 0
Thread Sorting started for file temp_in_00919.txt using Thread 4
Thread Sorting started for file temp_in_00920.txt using Thread 6
Thread Sorting started for file temp_in_00921.txt using Thread 1
Thread Sorting started for file temp_in_00922.txt using Thread 3
Thread Sorting started for file temp_in_00923.txt using Thread 7
Thread Sorting started for file temp_in_00924.txt using Thread 2
Thread Sorting started for file temp_in_00925.txt using Thread 5
Thread Sorting started for file temp_in_00926.txt using Thread 0
Thread Sorting started for file temp_in_00927.txt using Thread 4
Thread Sorting started for file temp_in_00928.txt using Thread 6
Thread Sorting started for file temp_in_00929.txt using Thread 1
Thread Sorting started for file temp_in_00930.txt using Thread 3
Thread Sorting started for file temp_in_00931.txt using Thread 7
Thread Sorting started for file temp_in_00932.txt using Thread 2
Thread Sorting started for file temp_in_00933.txt using Thread 5
Thread Sorting started for file temp_in_00934.txt using Thread 0
Thread Sorting started for file temp_in_00935.txt using Thread 4
Thread Sorting started for file temp_in_00936.txt using Thread 6
Thread Sorting started for file temp_in_00937.txt using Thread 1
Thread Sorting started for file temp_in_00938.txt using Thread 3
Thread Sorting started for file temp_in_00939.txt using Thread 7
Thread Sorting started for file temp_in_00940.txt using Thread 2
Thread Sorting started for file temp_in_00941.txt using Thread 5
Thread Sorting started for file temp_in_00942.txt using Thread 0
Thread Sorting started for file temp_in_00943.txt using Thread 4
Thread Sorting started for file temp_in_00944.txt using Thread 6
Thread Sorting started for file temp_in_00945.txt using Thread 1
Thread Sorting started for file temp_in_00946.txt using Thread 3
Thread Sorting started for file temp_in_00947.txt using Thread 7
Thread Sorting started for file temp_in_00948.txt using Thread 2
Thread Sorting started for file temp_in_00949.txt using Thread 5
Thread Sorting started for file temp_in_00950.txt using Thread 0
Thread Sorting started for file temp_in_00951.txt using Thread 4
Thread Sorting started for file temp_in_00952.txt using Thread 6
Thread Sorting started for file temp_in_00953.txt using Thread 1
Thread Sorting started for file temp_in_00954.txt using Thread 3
Thread Sorting started for file temp_in_00955.txt using Thread 7
Thread Sorting started for file temp_in_00956.txt using Thread 2


ubuntu@ip-172-31-59-242: /mnt/raid/Cloud_PA2
Thread Sorting started for file temp_in_00956.txt using Thread 2
Thread Sorting started for file temp_in_00957.txt using Thread 5
Thread Sorting started for file temp_in_00958.txt using Thread 0
Thread Sorting started for file temp_in_00959.txt using Thread 4
Thread Sorting started for file temp_in_00960.txt using Thread 6
Thread Sorting started for file temp_in_00961.txt using Thread 1
Thread Sorting started for file temp_in_00962.txt using Thread 3
Thread Sorting started for file temp_in_00963.txt using Thread 7
Thread Sorting started for file temp_in_00964.txt using Thread 2
Thread Sorting started for file temp_in_00965.txt using Thread 5
Thread Sorting started for file temp_in_00966.txt using Thread 0
Thread Sorting started for file temp_in_00967.txt using Thread 4
Thread Sorting started for file temp_in_00968.txt using Thread 6
Thread Sorting started for file temp_in_00969.txt using Thread 1
Thread Sorting started for file temp_in_00970.txt using Thread 3
Thread Sorting started for file temp_in_00971.txt using Thread 7
Thread Sorting started for file temp_in_00972.txt using Thread 2
Thread Sorting started for file temp_in_00973.txt using Thread 5
Thread Sorting started for file temp_in_00974.txt using Thread 0
Thread Sorting started for file temp_in_00975.txt using Thread 4
Thread Sorting started for file temp_in_00976.txt using Thread 6
Thread Sorting started for file temp_in_00977.txt using Thread 1
Thread Sorting started for file temp_in_00978.txt using Thread 3
Thread Sorting started for file temp_in_00979.txt using Thread 7
Thread Sorting started for file temp_in_00980.txt using Thread 2
Thread Sorting started for file temp_in_00981.txt using Thread 5
Thread Sorting started for file temp_in_00982.txt using Thread 0
Thread Sorting started for file temp_in_00983.txt using Thread 4
Thread Sorting started for file temp_in_00984.txt using Thread 6
Thread Sorting started for file temp_in_00985.txt using Thread 1
Thread Sorting started for file temp_in_00986.txt using Thread 3
Thread Sorting started for file temp_in_00987.txt using Thread 7
Thread Sorting started for file temp_in_00988.txt using Thread 2
Thread Sorting started for file temp_in_00989.txt using Thread 5
Thread Sorting started for file temp_in_00990.txt using Thread 0
Thread Sorting started for file temp_in_00991.txt using Thread 4
Thread Sorting started for file temp_in_00992.txt using Thread 6
Thread Sorting started for file temp_in_00993.txt using Thread 1
Thread Sorting started for file temp_in_00994.txt using Thread 3
Thread Sorting started for file temp_in_00995.txt using Thread 7
Thread Sorting started for file temp_in_00996.txt using Thread 2



g. Output Files

Output files “Sort-shared-memory-1GB.txt”, “Sort-shared-memory-10GB.txt” and “Sort-shared-memory-1TB.txt” are available in “\outputs\shared_memory” folder inside the main folder.

h. Tradeoffs made

- No log file generation, only the output can be redirected to a text file
- Final Merge Pass could have made faster using the already available inbuilt data structures like Priority Queue, Comparators, etc.
- Using priority queues or “divide and conquer” algorithms would had been beneficial
- As c3.large instance has only 2 cores and 3.75 GB RAM (RAM usable or available is still lesser) along with single Disk mounted as Raid 0, it becomes difficult to achieve performance gain even over multithreading environment

i. Difficulties faced

- Spot Instances were terminated automatically and the progress made was lost
- Coding and optimizing the Final Merge Pass was the most difficult part

j. Possible Improvements

- Using priority queues for final merge pass to achieve better performance
- Using categorical splitting (like Input Sampling in Hadoop) to split the original file in such a way that there will be no need of final merge pass
- Testing the program with better infrastructure (like d2.xlarge) instance
- More fine-tuning to achieve better and optimized final merge pass

2. Hadoop Sort

a. Problem

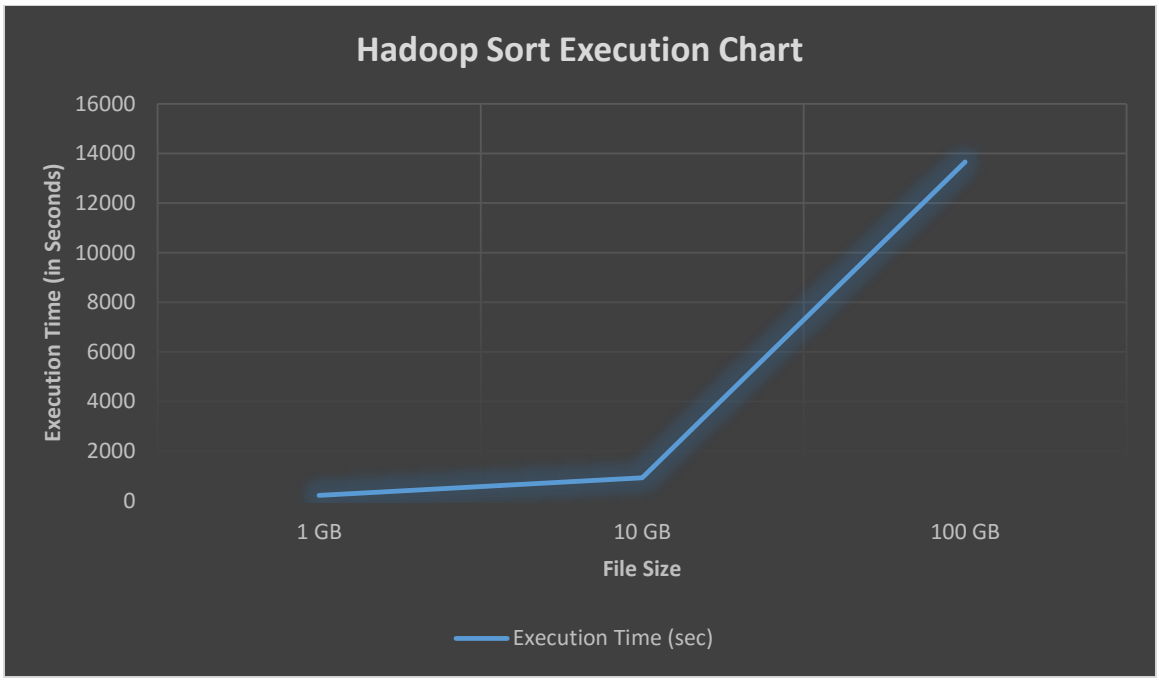
Install Hadoop on 17 c3.large instance nodes (including the HDFS distributed file system) and setup a virtual cluster; turn off replication in order to have lower storage requirement. Implement the Hadoop Sort application, and evaluate its performance on 1 node and 16 nodes. We must be doing strong scaling experiments (keep the dataset fixed) as we scale up from 1 node to 17 nodes. We can configure Hadoop to run the job tracker and filesystem metadata service on separate nodes, leaving 16 nodes available to run workers for map and reduce tasks. Answer the questions given in PA2 statement.

We were also supposed to measure the time to sort on the 1 GB & 10GB datasets on 1 node Hadoop and 1 GB & 100 GB datasets on 17 node Hadoop cluster. Save the first 10 lines of output from the Hadoop Sort application, as well as the last 10 lines of output, for each dataset, in Sort-Hadoop-10GB.txt & Sort-Hadoop-100GB.txt respectively.

b. Performance

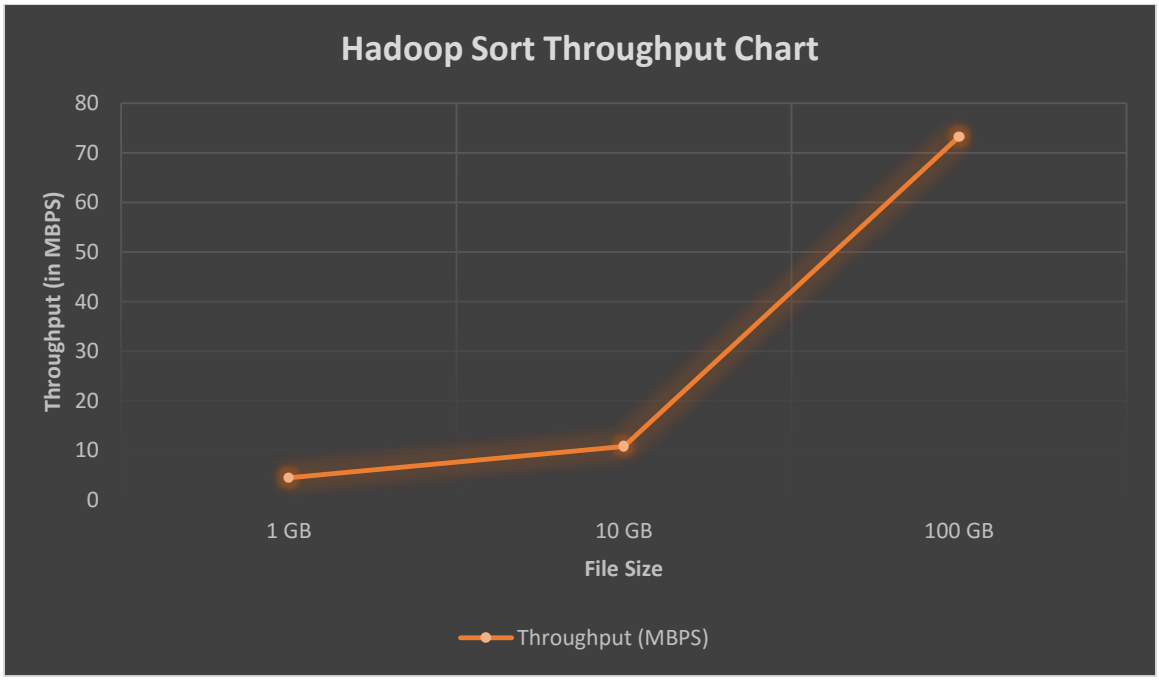
i. Hadoop Sort Execution

	1 GB	10 GB	100 GB
Execution Time (sec)	224	925	13661



ii. Hadoop Sort Throughput

	1 GB	10 GB	100 GB
Throughput (MBPS)	4.464286	10.81081	73.20108



(assuming 1,000,000 Bytes = 1MB)

c. Methodology

- Hadoop Sorting program is written in Java
- We must override Mapper and Reducer class in our program to use Hadoop's Map – Reduce framework to sort the provided data
- In Mapper Class (*MyMapper*), program tries to convert each individual line (consisting of 100 characters) into Key – Value pair (Key consisting of 10 characters and Value consisting of remaining characters)
- In Reducer Class (*MyReducer*), program tries to assign the results of desired functionality (e.g.: no. of occurrences in counting occurrences of key program) as Value for the respective Key
- Finally, output is combination of Key Value pairs available at the end of Reducer phase

d. Runtime Environment Settings

- Operating System
  Ubuntu 14.04.3 LTS, Release: 14.04, Codename: trusty
- Operating System Kernel
  Linux ip-172-31-60-162 3.13.0-74-generic #118-Ubuntu SMP Thu Dec 17 22:52:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
- Hadoop
  Hadoop 2.7.2
  Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r b165c4fe8a74265c792ce23f546c64604acf0e41
  Compiled by jenkins on 2016-01-26T00:08Z
  Compiled with protoc 2.5.0
  From source with checksum d0fda26633fa762bff87ec759ebe689c
  This command was run using /home/ubuntu/hadoop-2.7.2/share/hadoop/common/hadoop-common-2.7.2.jar
- Ant
  Apache Ant(TM) version 1.9.3 compiled on April 8 2014
- Java
  java version "1.7.0_95"
  OpenJDK Runtime Environment (IcedTea 2.6.4) (7u95-2.6.4-0ubuntu0.14.04.2)
  OpenJDK 64-Bit Server VM (build 24.95-b01, mixed mode)

e. Setup & Execution Process

i. Setting up Single Node Cluster for Hadoop

- Request either for additional storage or prepare RAID 0
  - sudo apt-get update
  - sudo apt-get install mdadm
  - echo "changing to raid 0"
  - lsblk
  - sudo mdadm --create --verbose /dev/md0 --level=0 --name=Cloud_PA2 --raid-devices=2 /dev/xvdb /dev/xvdc
  - sudo mkfs.ext4 -L Cloud_PA2 /dev/md0
  - sudo mkdir -p /mnt/raid
  - sudo mount LABEL=Cloud_PA2 /mnt/raid

- Install all prerequisites like Java, Ant, SSH, PSSH, etc.
  - sudo apt-get install default-jdk
  - sudo apt-get install ant
  - sudo apt-get install ssh
  - sudo apt-get install pssh

- Configure a dedicated Hadoop user
  - sudo addgroup hadoop
  - sudo adduser --ingroup hadoop hduser
  - sudo adduser hduser sudo

- Create and setup SSH certificates, test access to machine using public DNS ($public_name variable)
  - sudo ssh-keygen -t rsa -P ""
  - sudo cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
  - sudo chmod 0600 ~/.ssh/authorized_keys
  - sudo ssh $public_name

- Setup password-less access by copying the PEM file ($pem_name) to server
  - eval `ssh-agent -s`
  - sudo chmod 600 $pem_name
  - sudo ssh-add $pem_name

- Download, unzip and install Hadoop
  - sudo wget http://mirrors.sonic.net/apache/hadoop/common/hadoop-2.7.2/hadoop-2.7.2.tar.gz
  - sudo tar xvzf hadoop-2.7.2.tar.gz
  - sudo mkdir /usr/local/hadoop
  - sudo mv hadoop-2.7.2 /usr/local/hadoop/
  - sudo chown -R hduser:hadoop /usr/local/hadoop/hadoop-2.7.2
  - sudo cp /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/mapred-site.xml
  - sudo chmod 777 /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/mapred-site.xml

- update the *bashrc* file
  - echo "updating bashrc"
  - sudo cat << bashrc >> ~/.bashrc
  - #HADOOP VARIABLES START
  - export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
  - export HADOOP_INSTALL=/usr/local/hadoop/hadoop-2.7.2
  - export PATH=$PATH:$HADOOP_INSTALL/bin
  - export PATH=$PATH:$HADOOP_INSTALL/sbin
  - export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
  - export HADOOP_COMMON_HOME=$HADOOP_INSTALL
  - export HADOOP_HDFS_HOME=$HADOOP_INSTALL
  - export YARN_HOME=$HADOOP_INSTALL
  - export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
  - export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
  - export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
  - export CONF=/usr/local/hadoop/hadoop-2.7.2/etc/hadoop
  - # Some convenient aliases and functions for running Hadoop-related commands
  - unalias fs &> /dev/null
  - alias fs="hadoop fs"
  - unalias hls &> /dev/null
  - alias hls="fs -ls"
  - # If you have LZO compression enabled in your Hadoop cluster and
  - # compress job outputs with LZOP (not covered in this tutorial):
  - # Conveniently inspect an LZOP compressed file from the command
  - # line; run via:

- #
- # $ lzohead /hdfs/path/to/lzop/compressed/file.lzo
- #
- # Requires installed 'lzop' command.
- #
- lzohead () {
- hadoop fs -cat $1 | lzop -dc | head -1000 | less
- }
- #HADOOP VARIABLES END
- bashrc
- source ~/.bashrc

- update the */etc/hadoop/hadoop-env.sh* file
  - echo "updating hadoop-env.sh"
  - sudo cat << hadoop-env >> /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/hadoop-env.sh
  - export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
  - hadoop-env

- update the */etc/hadoop/core-site* file
  - sudo mkdir -p /app/hadoop/tmp
  - sudo chown hduser:hadoop /app/hadoop/tmp
  - sudo chmod 750 /app/hadoop/tmp
  - echo "updating core-site.xml"
  - sudo cat > /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/core-site.xml << core-site
  - <?xml version="1.0" encoding="UTF-8"?>
  - <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
  - <configuration>
  - <property>
  - <name>hadoop.tmp.dir</name>
  - <value>/app/hadoop/tmp</value>
  - </property>
  - <property>
  - <name>fs.default.name</name>
  - <value>hdfs://$public_name:54310</value>
  - <description>The name of the default file system. A URI whose
  - scheme and authority determine the FileSystem implementation. The

- uri's scheme determines the config property (fs.SCHEME.impl) naming
- the FileSystem implementation class. The uri's authority is used to
- determine the host, port, etc. for a filesystem.</description>
- </property>
- </configuration>
- core-site

- update the */etc/hadoop/hdfs-site* file
  - echo "updating hdfs-site.xml"
  - sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode
  - sudo mkdir -p /usr/local/hadoop_store/hdfs/datanode
  - sudo chown -R hduser:hadoop /usr/local/hadoop_store
  - sudo cat > /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/hdfs-site.xml << hdfs-site
  - <?xml version="1.0" encoding="UTF-8"?>
  - <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
  - <configuration>
  - <property>
  - <name>dfs.replication</name>
  - <value>1</value>
  - <description>Default block replication.
  - The actual number of replications can be specified when the file is created.
  - The default is used if replication is not specified in create time.
  - </description>
  - </property>
  - <property>
  - <name>dfs.permissions</name>
  - <value>false</value>
  - </property>
  - <property>
  - <name>dfs.namenode.name.dir</name>
  - <value>file:/usr/local/hadoop_store/hdfs/namenode</value>
  - </property>
  - <property>
  - <name>dfs.datanode.data.dir</name>
  - <value>file:/usr/local/hadoop_store/hdfs/datanode</value>
  - </property>
  - <property>

- <name>dfs.blocksize</name>
- <value>134217728</value>
- </property>
- </configuration>
- hdfs-site

- update the */etc/hadoop/yarn-site* file
  - sudo echo "updating yarn-site.xml"
  - sudo cat > /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/yarn-site.xml << yarn-site
  - <?xml version="1.0" encoding="UTF-8"?>
  - <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
  - <configuration>
  - <property>
  - <name>yarn.nodemanager.aux-services</name>
  - <value>mapreduce_shuffle</value>
  - </property>
  - <property>
  - <name>yarn.resourcemanager.scheduler.address</name>
  - <value>$public_name:54312</value>
  - </property>
  - <property>
  - <name>yarn.resourcemanager.address</name>
  - <value>$public_name:54313</value>
  - </property>
  - <property>
  - <name>yarn.resourcemanager.webapp.address</name>
  - <value>$public_name:54314</value>
  - </property>
  - <property>
  - <name>yarn.resourcemanager.resource-tracker.address</name>
  - <value>$public_name:54315</value>
  - </property>
  - <property>
  - <name>yarn.resourcemanager.admin.address</name>
  - <value>$public_name:54316</value>
  - </property>
  - <property>

- <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
- <value>org.apache.hadoop.mapred.ShuffleHandler</value>
- </property>
- </configuration>
- yarn-site

- update the */etc/hadoop/mapred-site* file
  - echo "updating mapred-site.xml"
  - sudo cat << mapred-site >> /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/mapred-site.xml
  - <?xml version="1.0"?>
  - <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
  - <configuration>
  - <property>
  - <name>mapred.job.tracker</name>
  - <value>$public_name:54311</value>
  - </property>
  - <property>
  - <name>mapred.framework.name</name>
  - <value>yarn</value>
  - </property>
  - <property>
  - <name>mapred.tasktracker.map.tasks.maximum</name>
  - <value>2</value>
  - </property>
  - <property>
  - <name>mapred.tasktracker.reduce.tasks.maximum</name>
  - <value>2</value>
  - </property>
  - </configuration>
  - mapred-site

- Format the namenode and start Hadoop
  - echo "Formatting HDFS"
  - sudo /usr/local/hadoop/hadoop-2.7.2/bin/hdfs namenode -format
  - echo "starting Hadoop"
  - sudo /usr/local/hadoop/hadoop-2.7.2/sbin/start-all.sh

- All the configuration files created using above steps will be optimized for further usage while continuing the experiments
- Completely automated shell scripts are available to perform above given steps in "HadoopPrereqInstallation.sh" and "HadoopInstallation.sh" files in "scripts" folder inside main folder

ii. Setting up 17 Node Cluster for Hadoop

- Create an AMI Image of the Single Node Hadoop cluster and launch 16 additional instances with the image
- Modify Master and Slaves files appropriately to configure the Master – Slave nodes in cluster. Sample *Master* and *Slave* files are available inside "config" folder inside main folder
- Change the *slaves* file in every slave node; remove the localhost and put the Private IP of the corresponding slave node.

iii. Execution Process

- Copy the *"HadoopSort.java"* in Hadoop directory
- Compile the files using *ant* mechanism
- Execute "*HadoopSort.jar"* with Input File Path and Output Path to start the sorting process by executing following commands on terminal keeping Hadoop installation directory as working directory
    - export HADOOP_CLASSPATH=/usr/lib/jvm/java-7-openjdk-amd64/lib/tools.jar
    - ./bin/hadoop com.sun.tools.javac.Main HadoopSort.java
    - jar cf mr.jar HadoopSort*.class
    - ./bin/hadoop jar HadoopSort.jar HadoopSort src dstn

iv. General Information about Hadoop

Hadoop

At the most basic level, Hadoop is an open-source software platform designed to store and process quantities of data that are too large for just one particular device or server. Hadoop's strength lies in its ability to scale across thousands of commodity servers that don't share memory or disk space.

Hadoop delegates tasks across these servers (called “worker nodes” or “slave nodes”), essentially harnessing the power of each device and running them together simultaneously. This is what allows massive amounts of data to be analyzed: splitting the tasks across different locations in this manner allows bigger jobs to be completed faster.

Hadoop can be thought of as an ecosystem—it’s comprised of many different components that all work together to create a single platform. There are two key functional components within this ecosystem: The storage of data (Hadoop Distributed File System, or HDFS) and the framework for running parallel computations on this data (MapReduce). Let’s take a closer look at each.

Hadoop Distributed File System (HDFS)

HDFS is the “secret sauce” that enables Hadoop to store huge files. It’s a scalable file system that distributes and stores data across all machines in a Hadoop cluster (a group of servers). Each HDFS cluster contains the following:

NameNode: Runs on a “master node” that tracks and directs the storage of the cluster.

DataNode: Runs on “slave nodes,” which make up the majority of the machines within a cluster. The NameNode instructs data files to be split into blocks, each of which are replicated three times and stored on machines across the cluster. These replicas ensure the entire system won’t go down if one server fails or is taken offline—known as “fault tolerance.”

Client machine: neither a NameNode or a DataNode, Client machines have Hadoop installed on them. They’re responsible for loading data into the cluster, submitting MapReduce jobs and viewing the results of the job once complete.

MapReduce

MapReduce is the system used to efficiently process the large amount of data Hadoop stores in HDFS. Originally created by Google, its strength lies in the ability to divide a single large data processing job into smaller tasks. All MapReduce jobs are written in Java, but other languages can be used via the Hadoop Streaming API, which is a utility that comes with Hadoop.

Once the tasks have been created, they’re spread across multiple nodes and run simultaneously (the “map” step). The “reduce” phase combines the results together.

Imagine, for example, that an entire MapReduce job is the equivalent of building a house. Each job is broken down into individual tasks (e.g. lay the foundation, put up drywall) and assigned to various workers, or “mappers” and “reducers.” Completing each task results in a single, combined output: the house is complete.

This delegation of tasks is handled by two “daemons,” the JobTracker and TaskTracker. The technical definition of a daemon is “a process that is long-lived.” In our house example, a daemon can be thought of as a foreman: the jobs may change (new houses must be built), workers will come and go, but the foreman is always there to oversee the job and delegate tasks.

JobTracker: The JobTracker oversees how MapReduce jobs are split up into tasks and divided among nodes within the cluster.

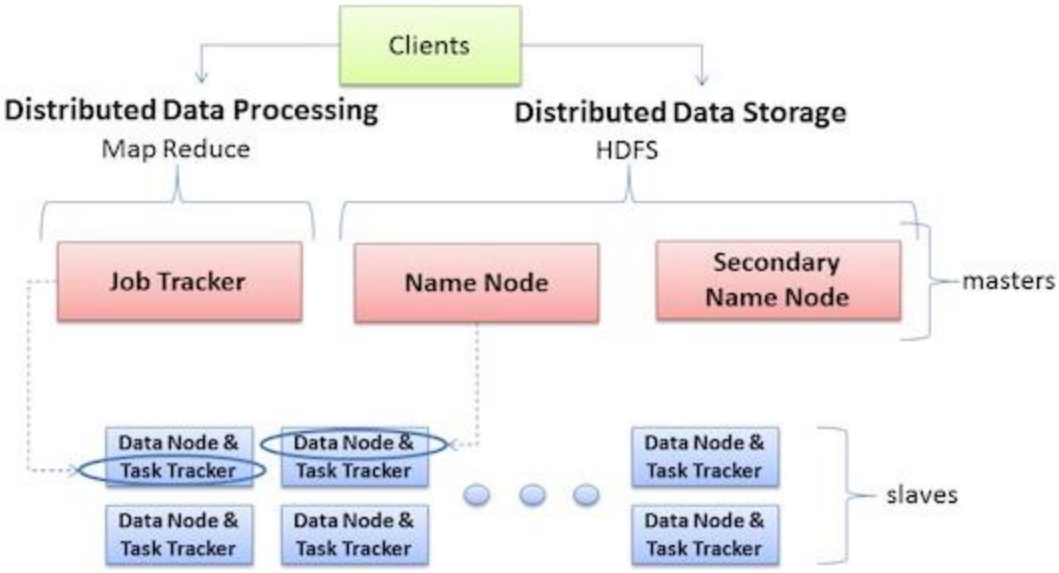
TaskTracker: The TaskTracker accepts tasks from the JobTracker, performs the work and alerts the JobTracker once it’s done. TaskTrackers and DataNodes are located on the same nodes to improve performance.

Data locality

An important concept with HDFS and MapReduce, data locality can best be described as “bringing the compute to the data.” In other words, whenever you use a MapReduce program on a particular part of HDFS data, you always want to run that program on the node, or machine, that actually stores this data in HDFS. Doing so allows processes to be run much faster, since it prevents you from having to move large amounts of data around.

When a MapReduce job is submitted, part of what the JobTracker does is look to see which machines the blocks required for the task are located on. This is why, when the NameNode splits data files into blocks, each one is replicated three times: the first is stored on the same machine as the block, while the second and third are each stored on separate machines.

Storing the data across three machines thus gives you a much higher chance of achieving data locality, since it’s likely that at least one of the machines will be freed up enough to process the data stored at that particular location.



v. Question and Answers

A. Write a description of the function for 1) conf/master, 2) conf/slaves, 3) conf/core-site.xml, 4) conf/hdfs-site.xml, 5) conf/mapred-site.xml. What modifications you had to make to go from 1 node to multiple nodes?

Answer:

master

This file contains a list of machine names / IPs (one per line) that each run a secondary namenode. The 'Masters' file at Master server contains a hostname, Secondary Name Node servers.

slaves

This file lists the hosts, one per line, where the Hadoop slave daemons (datanodes and tasktrackers) will run. By default this contains the single entry localhost

hadoop-env.sh

This file specifies environment variables that affect the JDK used by Hadoop Daemon (bin/hadoop).

As Hadoop framework is written in Java and uses Java Runtime environment, one of the important environment variables for Hadoop daemon is $JAVA_HOME in hadoop-env.sh. This variable directs Hadoop daemon to the Java path in the system.

This file is also used for setting another Hadoop daemon execution environment such as heap size (HADOOP_HEAP), hadoop home (HADOOP_HOME), log file location (HADOOP_LOG_DIR), etc.

core-site.xml

This file informs Hadoop daemon where NameNode runs in the cluster. It contains the configuration settings for Hadoop Core such as I/O settings that are common to HDFS and MapReduce. Hostname and port are the machine name or IP and port on which NameNode daemon runs and listens. It also informs the Name Node as to which IP and port it should bind. The commonly used port is 8020.

hdfs-site.xml

This file contains the configuration settings for HDFS daemons; the Name Node, the Secondary Name Node, and the data nodes. This file is also used to specify default block replication and permission checking on HDFS. The actual number of replications can also be specified when the file is created. The default is used if replication is not specified in create time.

The value “true” for property ‘dfs.permissions’ enables permission checking in HDFS and the value “false” turns off the permission checking. Switching from one parameter value to the other does not change the mode, owner or group of files or directories.

mapred-site.xml

This file contains the configuration settings for MapReduce daemons; the job tracker and the task-trackers. The mapred.job.tracker parameter is a hostname (or IP address) and port pair on which the Job Tracker listens for RPC communication. This parameter specifies the location of the Job Tracker to Task Trackers and MapReduce clients.

mapred.job.tracker	Host or IP and port of JobTracker.	host:port pair.
mapred.system.dir	Path on the HDFS where the MapReduce framework stores system files e.g. /hadoop/mapred/system/.	This is in the default filesystem (HDFS) and must be accessible from both the server and client machines.
mapred.local.dir	Comma-separated list of paths on the local filesystem where temporary MapReduce data is written.	Multiple paths help spread disk i/o.
mapred.tasktracker.{map|reduce}.tasks.maximum	The maximum number of MapReduce tasks, which are run simultaneously on a given TaskTracker, individually.	Defaults to 2 (2 maps and 2 reduces), but vary it depending on your hardware.
dfs.hosts/dfs.hosts.exclude	List of permitted/excluded DataNodes.	If necessary, use these files to control the list of allowable datanodes.
mapred.hosts/mapred.hosts.exclude	List of permitted/excluded TaskTrackers.	If necessary, use these files to control the list of allowable TaskTrackers.
mapred.queue.names	Comma separated list of queues to which jobs can be submitted.	The MapReduce system always supports atleast one queue with the name as *default*. Hence, this

		parameter's value should always contain the string *default*. Some job schedulers supported in Hadoop, like the Capacity Scheduler, support multiple queues. If such a scheduler is being used, the list of configured queue names must be specified here. Once queues are defined, users can submit jobs to a queue using the property name *mapred.job.queue.name* in the job configuration. There could be a separate configuration file for configuring properties of these queues that is managed by the scheduler. Refer to the documentation of the scheduler for information on the same.
mapred.acls.enabled	Boolean, specifying whether checks for queue ACLs and job ACLs are to be done for authorizing users for doing queue operations and job operations.	If *true*, queue ACLs are checked while submitting and administering jobs and job ACLs are checked for authorizing view and modification of jobs. Queue ACLs are specified using the configuration parameters of the form *mapred.queue.queue-name.acl-name*, defined below under mapred-queue-acls.xml. Job ACLs are described at Job Authorization

Following changes were made into various files while moving from 1 node to 17 node cluster

- Number of mappers and reducer tasks were adjusted in mapred.xml
- *slaves* files on individual slave instances was changed by replacing localhost entry with the Private IP of the AWS instance
- *Slaves* file on master instance was modified to include the Private IPs of all the slave instances

B. What is a Master node? What is a Slaves node?

Answer:

Master Node:

The master nodes in distributed Hadoop clusters host the various storage and processing management services, described in this list, for the entire Hadoop cluster. Redundancy is critical in avoiding single points of failure, so you see two switches and three master nodes.

- **NameNode:** Manages HDFS storage. To ensure high availability, you have both an active NameNode and a standby NameNode. Each runs on its own, dedicated master node.

- **Checkpoint node (or backup node):** Provides checkpointing services for the NameNode. This involves reading the NameNode's edit log for changes to files in HDFS (new, deleted, and appended files) since the last checkpoint, and applying them to the NameNode's master file that maps files to data blocks.

  In addition, the Backup Node keeps a copy of the file system namespace in memory and keeps it in sync with the state of the NameNode. For high availability deployments, do not use a checkpoint node or backup node — use a Standby NameNode instead. In addition to being an active standby for the NameNode, the Standby NameNode maintains the checkpointing services and keeps an up-to-date copy of the file system namespace in memory.

- **Resource Manager:** Oversees the scheduling of application tasks and management of the Hadoop cluster's resources. This service is the heart of YARN.

- **JobTracker:** For Hadoop 1 servers, handles cluster resource management and scheduling. With YARN, the JobTracker is obsolete and isn't used. A number of Hadoop deployments still haven't migrated to Hadoop 2 and YARN.

Slave Node:

In a Hadoop universe, slave nodes are where Hadoop data is stored and where data processing takes place. The following services enable slave nodes to store and process data:

- **NodeManager:** Coordinates the resources for an individual slave node and reports back to the Resource Manager.

- **ApplicationMaster:** Tracks the progress of all the tasks running on the Hadoop cluster for a specific application. For each client application, the Resource Manager deploys an instance of the ApplicationMaster service in a container on a slave node. (Remember that any node running the NodeManager service is visible to the Resource Manager.)

- **Container:** A collection of all the resources needed to run individual tasks for an application. When an application is running on the cluster, the Resource Manager schedules the tasks for the application to run as container services on the cluster's slave nodes.

- **TaskTracker:** Manages the individual map and reduce tasks executing on a slave node for Hadoop 1 clusters. In Hadoop 2, this service is obsolete and has been replaced by YARN services.

- **DataNode:** An HDFS service that enables the NameNode to store blocks on the slave node.

Each slave node is always running a DataNode instance (which enables HDFS to store and retrieve data blocks on the slave node) and a NodeManager instance (which enables the Resource Manager to assign application tasks to the slave node for processing). The container processes are individual tasks for applications that are running on the cluster.

Each running application has a dedicated ApplicationMaster task, which also runs in a container, and tracks the execution of all the tasks executing on the cluster until the application is finished.

Slave nodes have a fixed number of map slots and reduce slots for the execution of map and reduce tasks respectively.

C. 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?

Answer

Each port is assigned for specific functionality (e.g.: 54310 for HDFS functionality, 54312 for Scheduler functionality, 54313 for Resource Manager, 54315 for Resource Tracker, etc.). If we use same port number for all users or functionalities, then there will be collision and communication issues between Master(s) and Slave(s) Nodes.

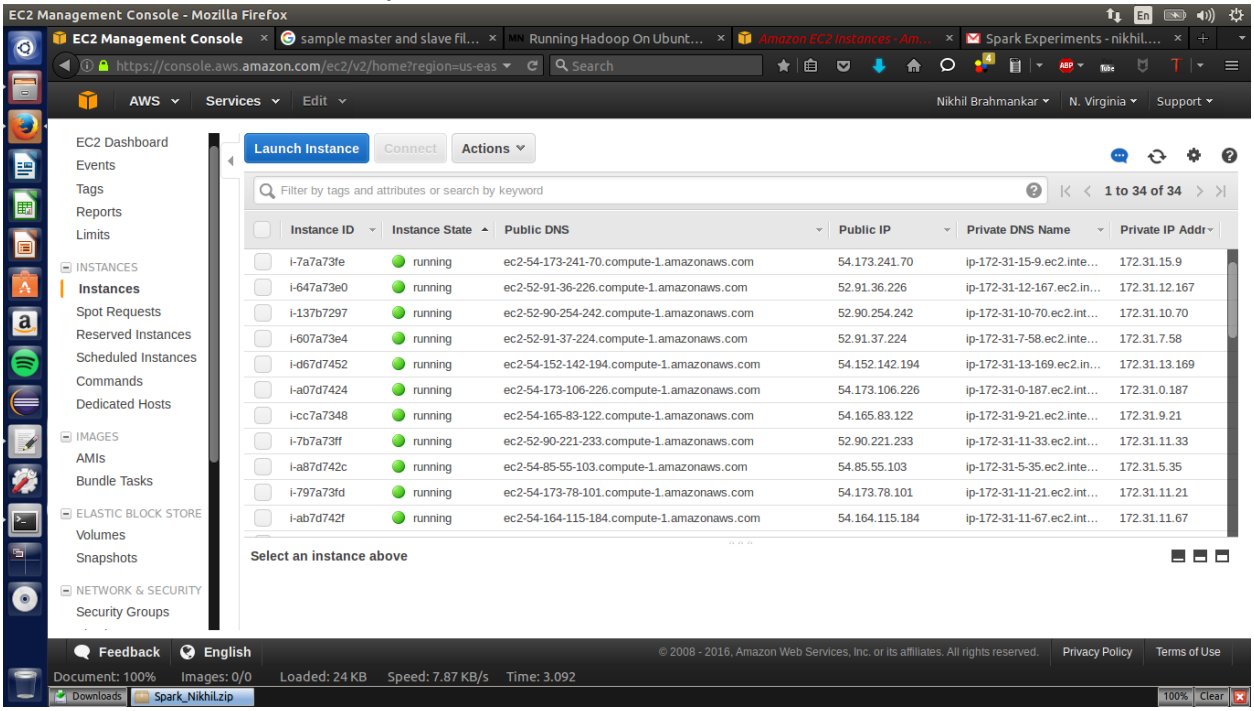
D. How can we change the number of mappers and reducers from the configuration file?

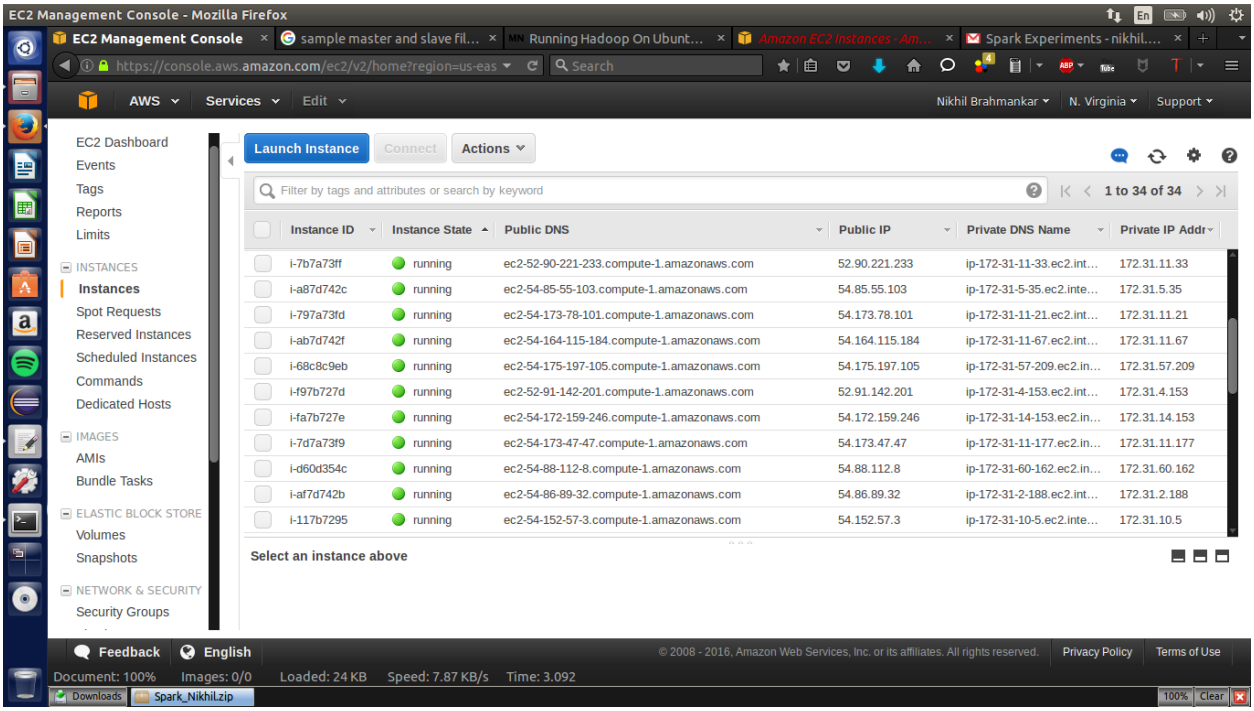
Answer:

Total number of mappers and reducers can be changed by editing "*mapred.tasktracker.map.tasks.maximum*" and "*mapred.tasktracker.reduce.tasks.maximum*" properties from the configuration file "mapred-site.xml" and rerun the multi-node shell.

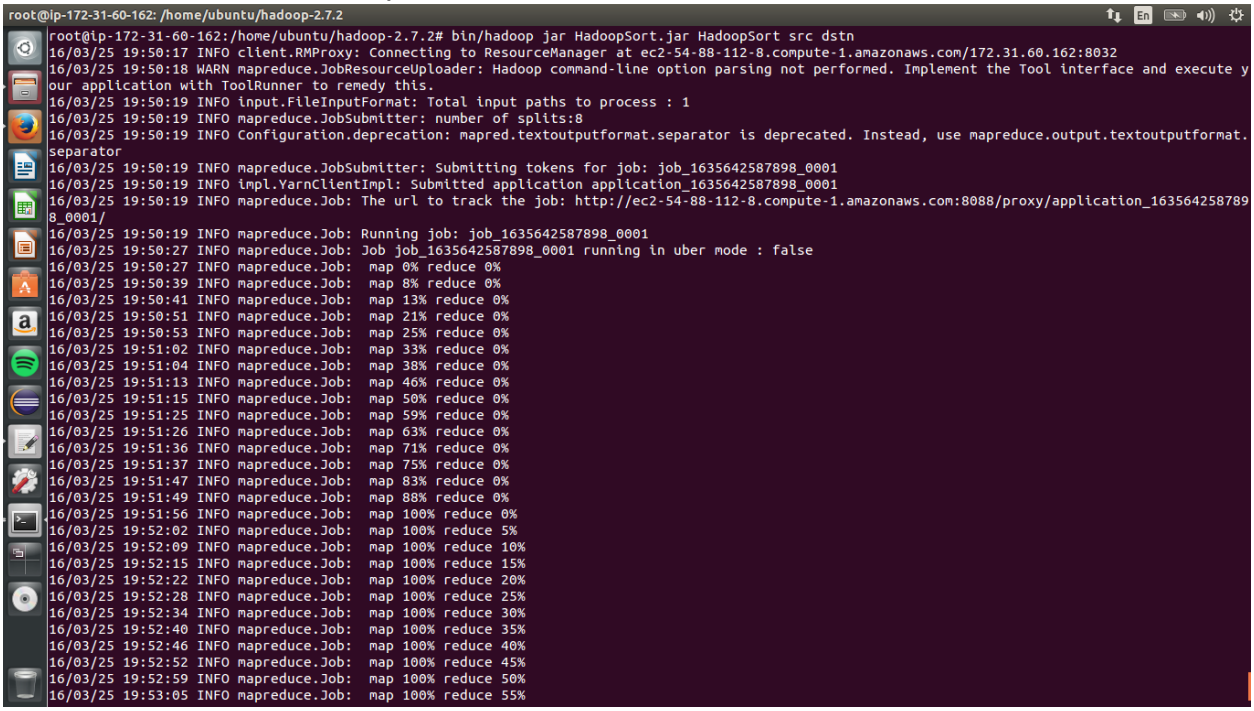
f. Output Screenshots

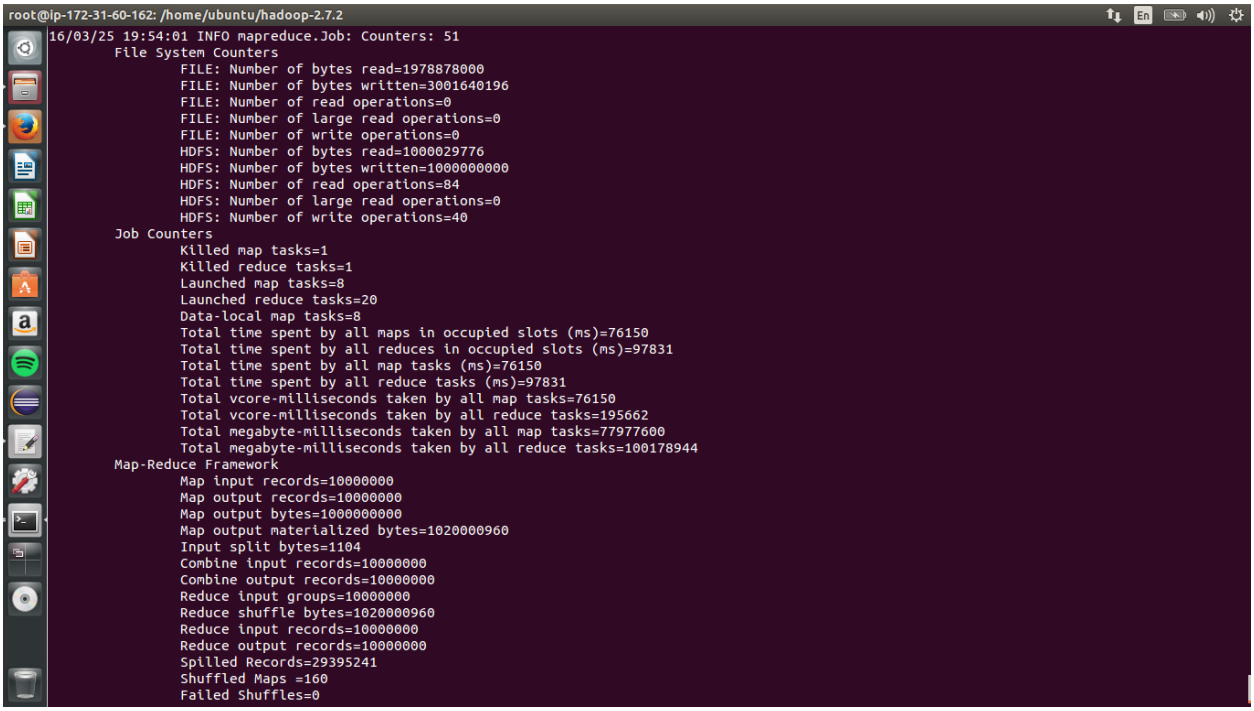
i. Hadoop AWS Instances Screenshots

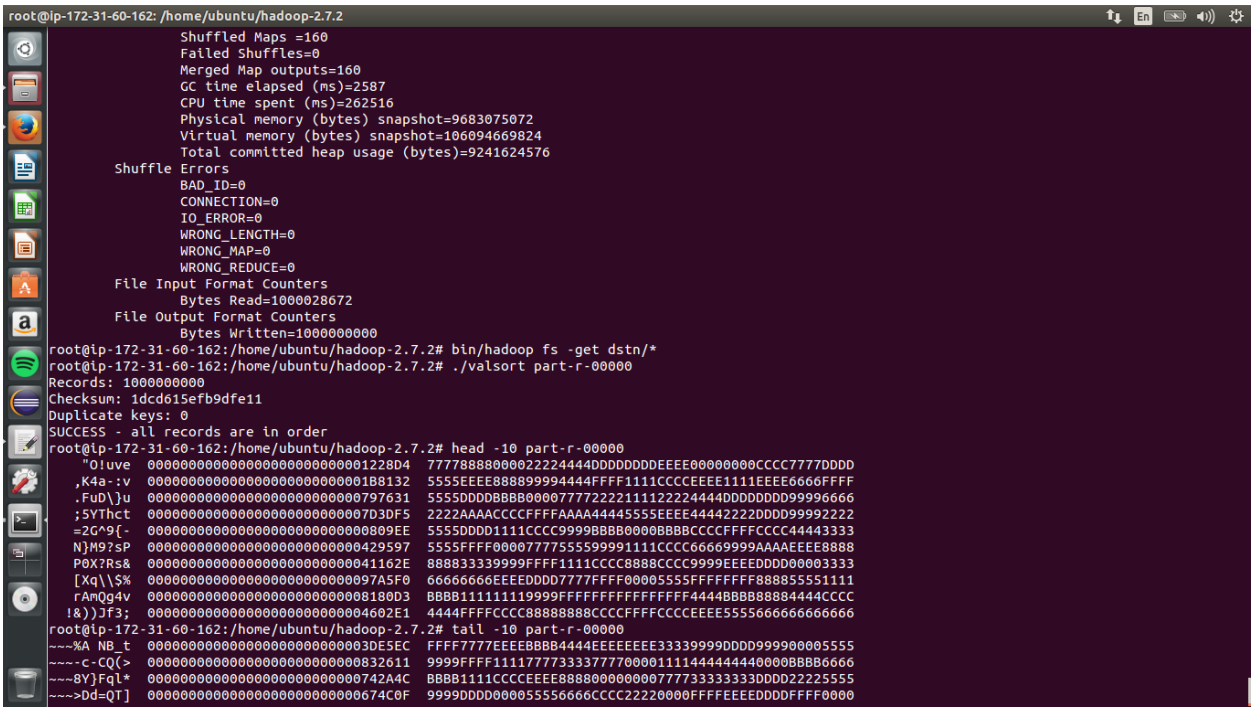


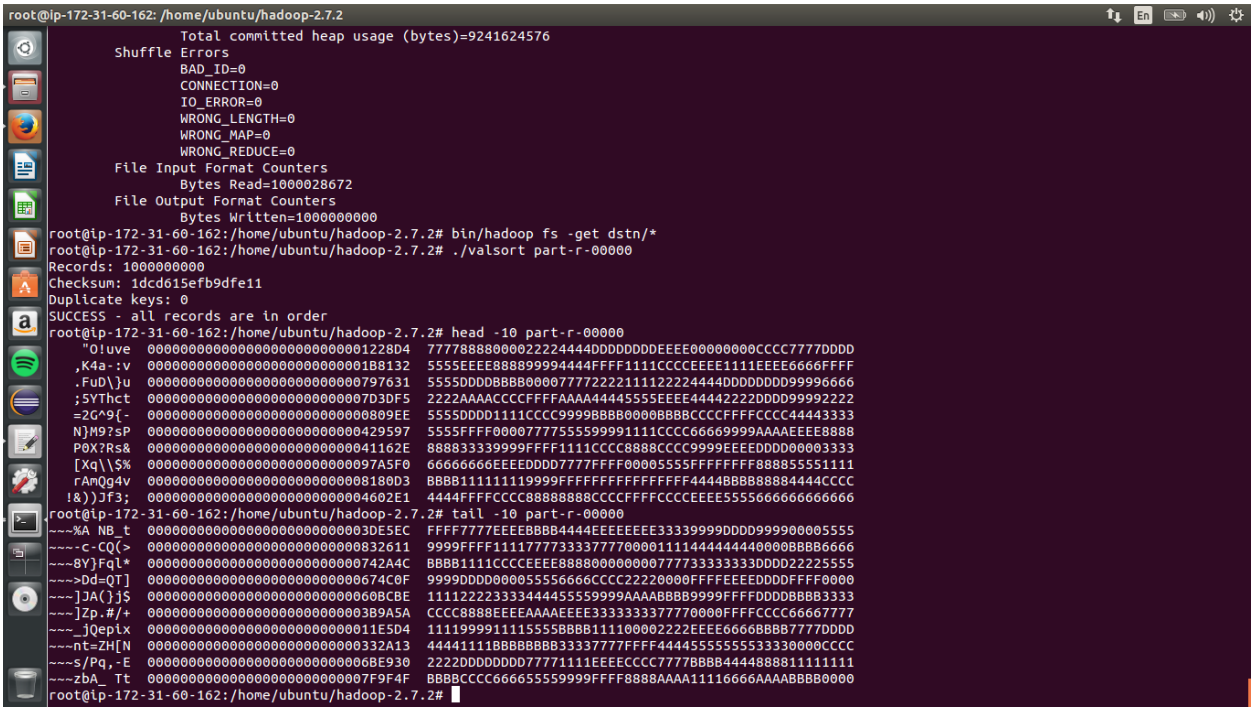


ii. Hadoop 1 GB 1 Node Sort

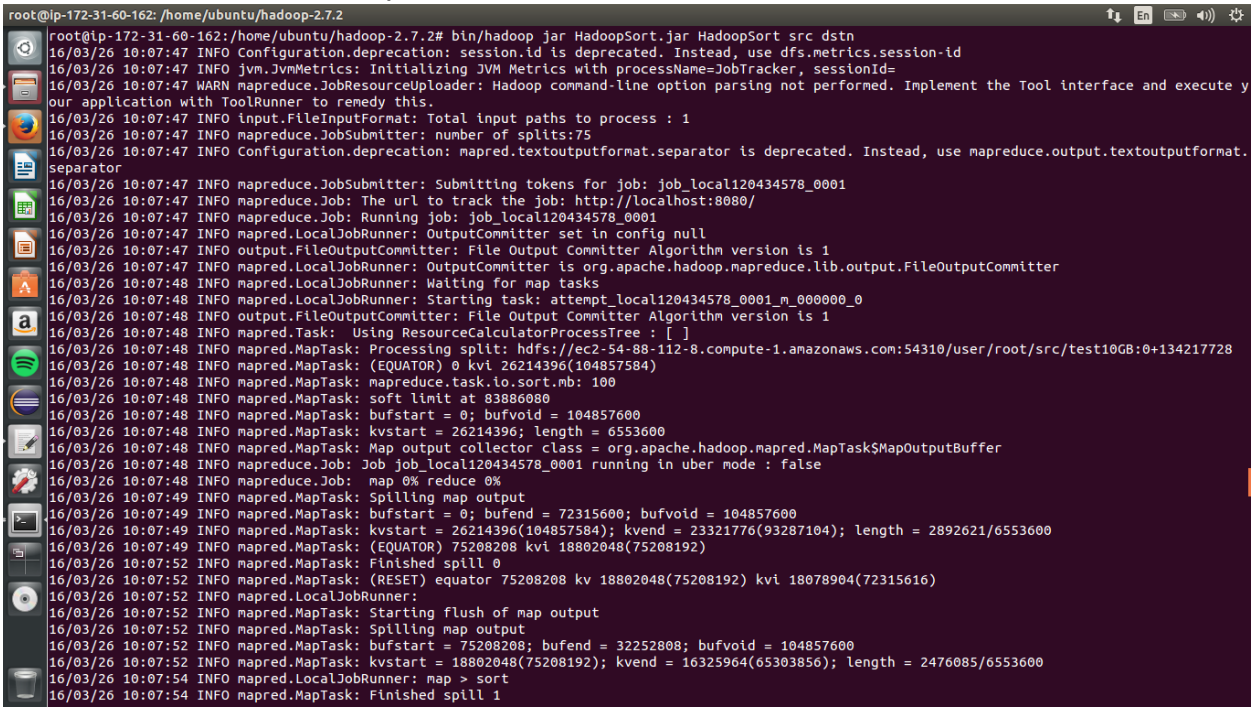


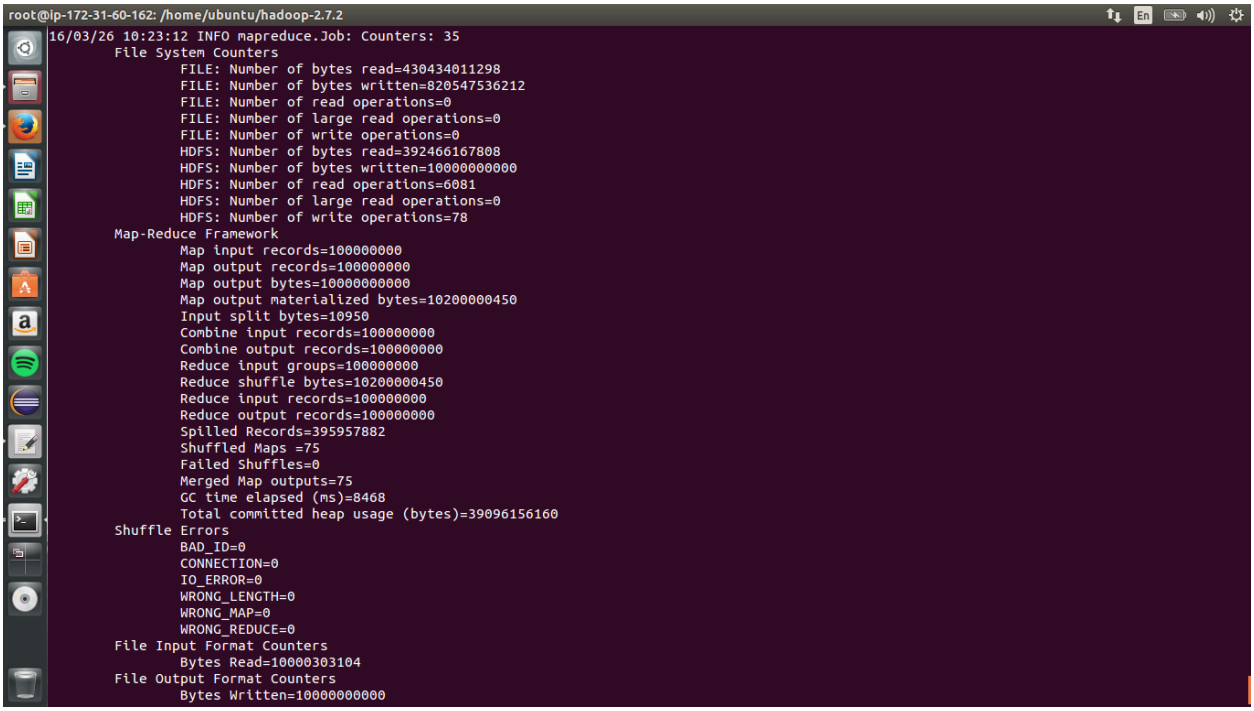


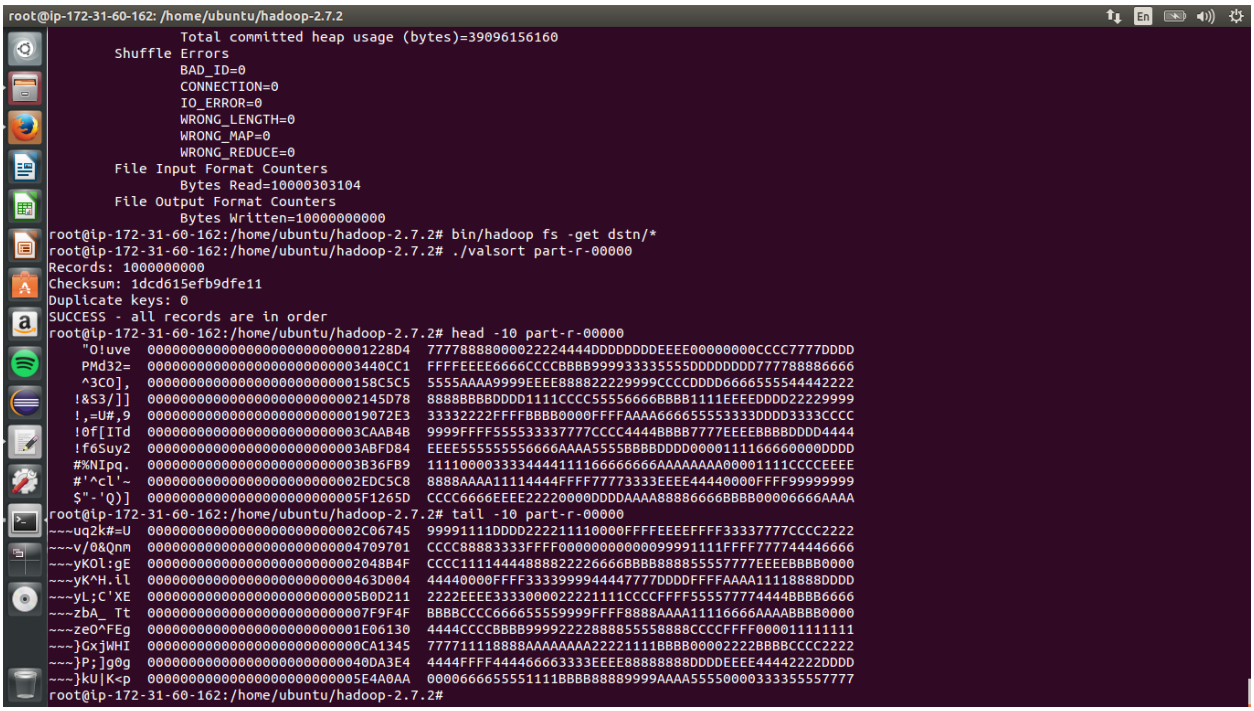

root@ip-172-31-60-162: /home/ubuntu/hadoop-2.7.2
Shuffled Maps =160
Failed Shuffles=0
Merged Map outputs=160
GC time elapsed (ms)=2587
CPU time spent (ms)=262516
Physical memory (bytes) snapshot=9683075072
Virtual memory (bytes) snapshot=106094669824
Total committed heap usage (bytes)=9241624576
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=1000028672
File Output Format Counters
Bytes Written=1000000000
root@ip-172-31-60-162:/home/ubuntu/hadoop-2.7.2# bin/hadoop fs -get dstn/*
root@ip-172-31-60-162:/home/ubuntu/hadoop-2.7.2# ./valsort part-r-00000
Records: 1000000000
Checksum: 1dcd615efb9dfe11
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-60-162:/home/ubuntu/hadoop-2.7.2# head -10 part-r-00000
"O!uve 00000000000000000000000001228D4 77778888000022224444DDDDDDDDEEEE00000000CCCC7777DDDD
,K4a-:v 00000000000000000000000001B8132 5555EEEE888899994444FFFF1111CCCCEEEE1111EEEE6666FFFF
.FuD\}u 00000000000000000000000000797631 5555DDDDBBBB00007777222211112222444DDDDDDDD99996666
;5YThct 00000000000000000000000007D3DF5 2222AAAACCCCFFFFAAAA44445555EEEE44442222DDDD99992222
=2G^9{- 00000000000000000000000000809EE 5555DDDD1111CCCC9999BBBB0000BBBBCCCCFFFFCCCC44443333
N}M9?sP 00000000000000000000000000429597 5555FFFF000077775555999911111CCCC66669999AAAAEEEE8888
P0X?Rs& 00000000000000000000000041162E 888833339999FFFF1111CCCC8888CCCC9999EEEEDDDD00003333
[Xq\\$% 00000000000000000000000097A5F0 66666666EEEEDDDD7777FFFF00005555FFFFFFFF888855551111
rAmQg4v 00000000000000000000000008180D3 BBBB111111119999FFFFFFFFFFFFFFFF4444BBBB88884444CCCC
!&))Jf3; 00000000000000000000000004602E1 4444FFFFCCCC88888888CCCCFFFFCCCCEEEE5555666666666666
root@ip-172-31-60-162:/home/ubuntu/hadoop-2.7.2# tail -10 part-r-00000
~~~%A NB_t 00000000000000000000000003DE5EC FFFF7777EEEEBBBB4444EEEEEEEE33339999DDDD999900005555
~~~-c-CQ(> 00000000000000000000000000832611 9999FFFF1111777733337777000011114444444440000BBBB6666
~~~8Y}Fql* 00000000000000000000000000742A4C BBBB1111CCCCEEEE88880000000077773333333333DDDD22225555
~~~>Dd=QT] 00000000000000000000000000674C0F 9999DDDD000055556666CCCC22220000FFFFEEEEDDDDFFFF0000


root@ip-172-31-60-162: /home/ubuntu/hadoop-2.7.2
Total committed heap usage (bytes)=9241624576
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=1000028672
File Output Format Counters
Bytes Written=1000000000
root@ip-172-31-60-162:/home/ubuntu/hadoop-2.7.2# bin/hadoop fs -get dstn/*
root@ip-172-31-60-162:/home/ubuntu/hadoop-2.7.2# ./valsort part-r-00000
Records: 1000000000
Checksum: 1dcd615efb9dfe11
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-60-162:/home/ubuntu/hadoop-2.7.2# head -10 part-r-00000
"O!uve 00000000000000000000000001228D4 77778888000022224444DDDDDDDDEEEE00000000CCCC7777DDDD
,K4a-:v 00000000000000000000000001B8132 5555EEEE888899994444FFFF1111CCCCEEEE1111EEEE6666FFFF
.FuD\}u 00000000000000000000000000797631 5555DDDDBBBB00007777222211112222444DDDDDDDD99996666
;5YThct 00000000000000000000000007D3DF5 2222AAAACCCCFFFFAAAA44445555EEEE44442222DDDD99992222
=2G^9{- 00000000000000000000000000809EE 5555DDDD1111CCCC9999BBBB0000BBBBCCCCFFFFCCCC44443333
N}M9?sP 00000000000000000000000000429597 5555FFFF000077775555999911111CCCC66669999AAAAEEEE8888
P0X?Rs& 00000000000000000000000041162E 888833339999FFFF1111CCCC8888CCCC9999EEEEDDDD00003333
[Xq\\$% 00000000000000000000000097A5F0 66666666EEEEDDDD7777FFFF00005555FFFFFFFF888855551111
rAmQg4v 00000000000000000000000008180D3 BBBB111111119999FFFFFFFFFFFFFFFF4444BBBB88884444CCCC
!&))Jf3; 00000000000000000000000004602E1 4444FFFFCCCC88888888CCCCFFFFCCCCEEEE5555666666666666
root@ip-172-31-60-162:/home/ubuntu/hadoop-2.7.2# tail -10 part-r-00000
~~~%A NB_t 00000000000000000000000003DE5EC FFFF7777EEEEBBBB4444EEEEEEEE33339999DDDD999900005555
~~~-c-CQ(> 00000000000000000000000000832611 9999FFFF1111777733337777000011114444444440000BBBB6666
~~~8Y}Fql* 00000000000000000000000000742A4C BBBB1111CCCCEEEE88880000000077773333333333DDDD22225555
~~~>Dd=QT] 00000000000000000000000000674C0F 9999DDDD000055556666CCCC22220000FFFFEEEEDDDDFFFF0000
~~~]JA(}j$ 00000000000000000000000000060BCBE 11112222333344445555999AAAABBBB9999FFFFDDDDBBBB3333
~~~]Zp.#/+ 00000000000000000000000003B9A5A CCCC8888EEEEEAAAAEEEE33333333777700000FFFFCCCC66667777
~~~_jQepix 00000000000000000000000011E5D4 11119999111155555BBBB111100002222EEEEE6666BBBB7777DDDD
~~~nt=ZH[N 00000000000000000000000000332A13 44441111BBBBBBBB33337777FFFF444455555555333300000CCCC
~~~S/Pq,-E 00000000000000000000000006BE930 2222DDDDDDDD77771111EEEECCCC7777BBBB44448888111111111
~~~zbA_ Tt 00000000000000000000000007F9F4F BBBBCCCC666655559999FFFF8888AAAA11116666AAAABBBB0000
root@ip-172-31-60-162:/home/ubuntu/hadoop-2.7.2#

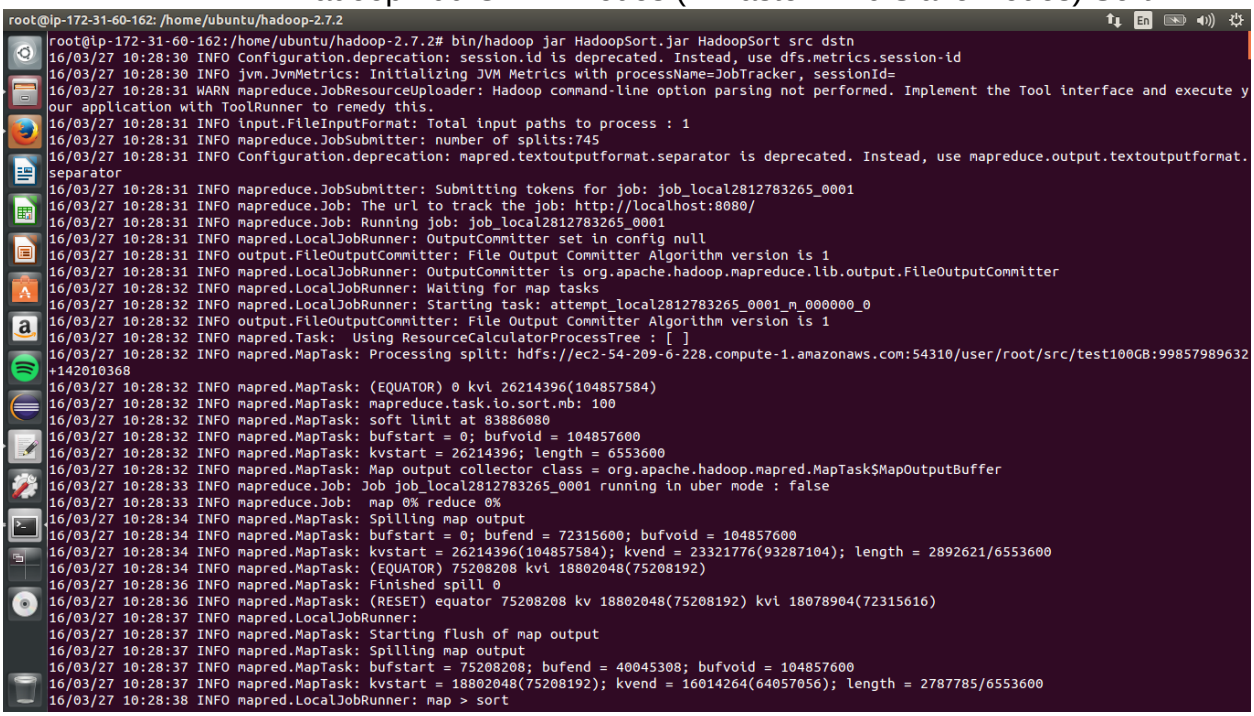
iii. Hadoop 10 GB 1 Node Sort

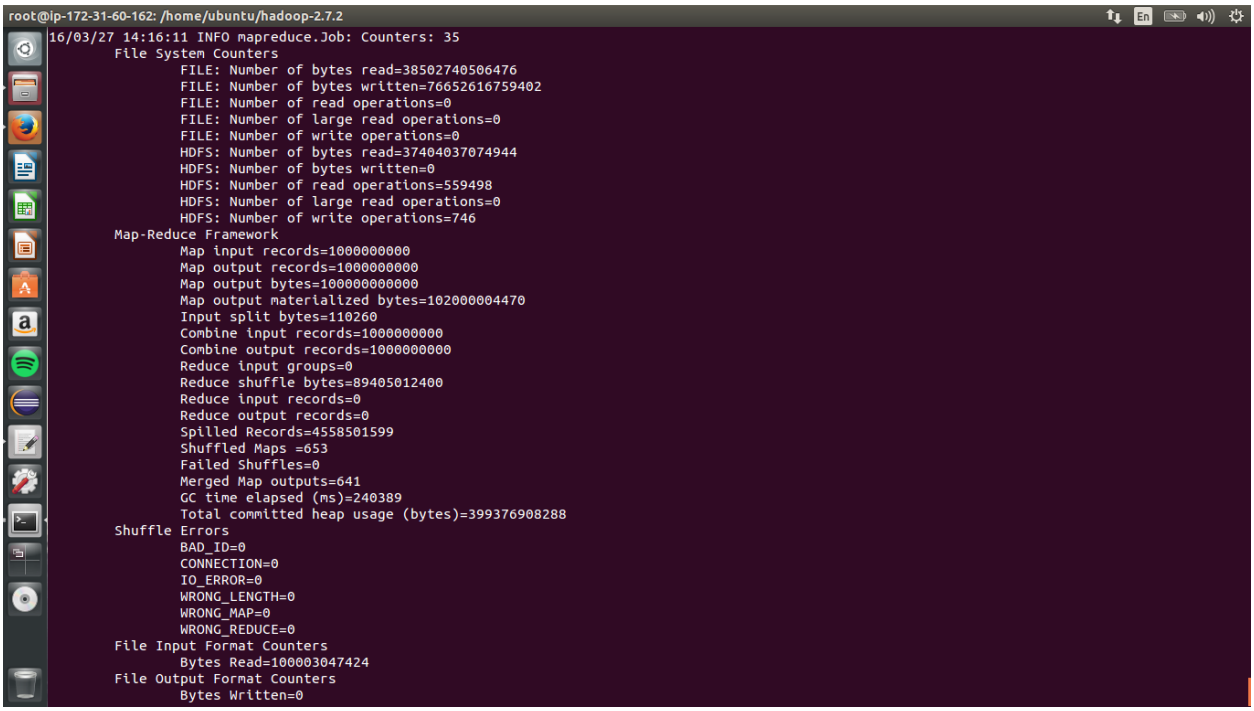


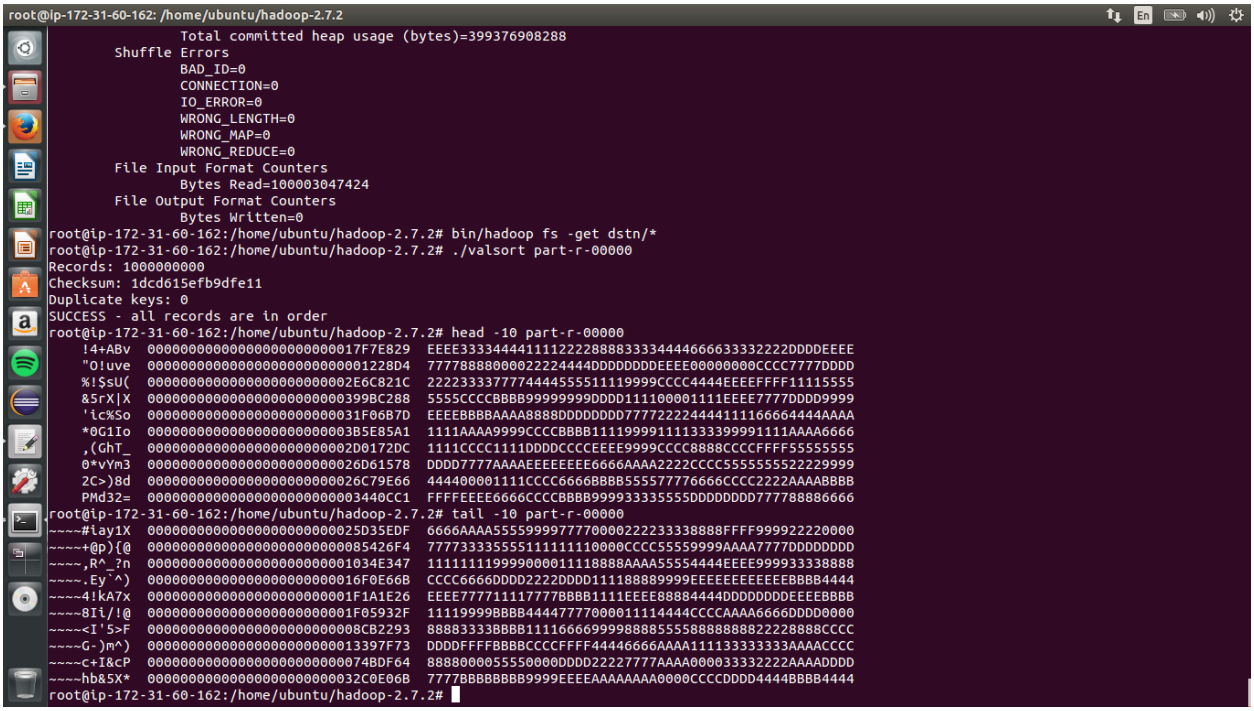




iv. Hadoop 100 GB 17 Nodes (1 Master + 16 Slave Nodes) Sort







g. Output Files

Output files “Sort-hadoop-1GB.txt”, “Sort-hadoop-10GB.txt” and “Sort-hadoop-100GB.txt” are available in “\outputs\hadoop” folder inside the main folder.

h. Tradeoffs made

- Hadoop was not utilized optimum because of the lack of knowledge
- Lack of knowledge about various settings that be utilized in Hadoop at the Job level instead of configuration file level results in making job specific configuration files

i. Difficulties faced

- Spot Instances were terminated automatically and the progress made was lost
- Understanding and removing the errors by trial and error method in absence of in-depth knowledge of Hadoop

j. Possible Improvements

- Need to understand the configuration files and their properties in a better way
- Need to understand the usage of TotalOrderPartitioner, InputSampler, RandomSampler classes which may help to optimize sorting performance in better way
- Dedicated instances would have helped in more learning that reconfiguring the instances

3. Spark Sort

a. Problem

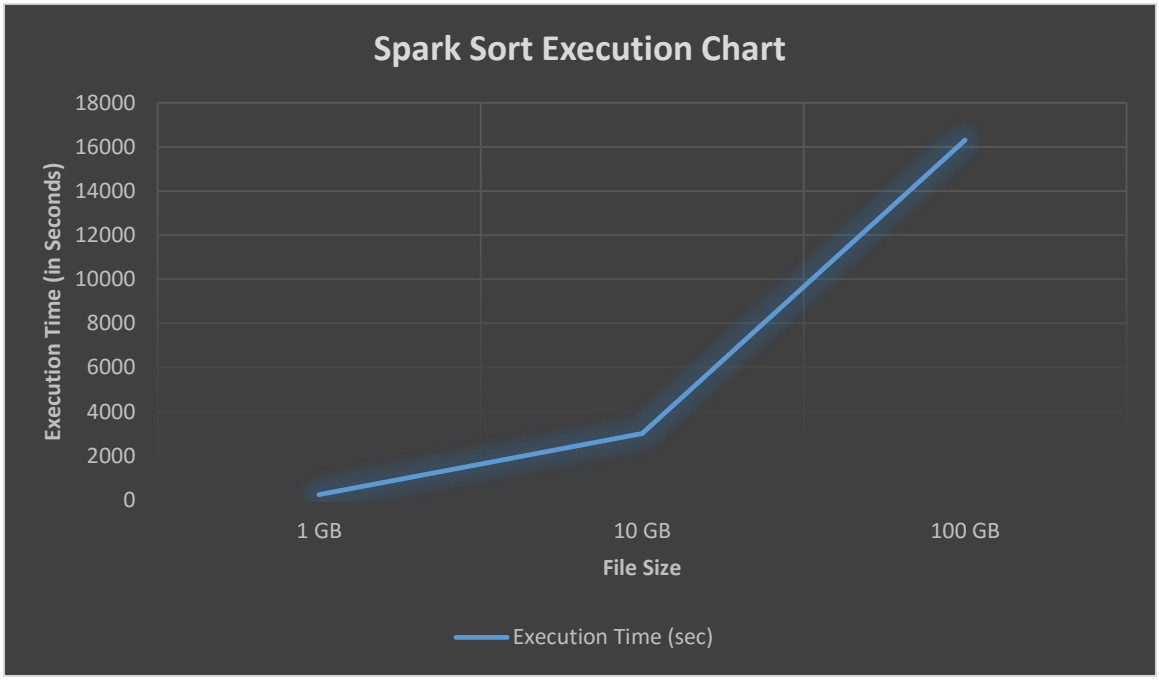
Install Spark on 17 c3.large instance nodes (including the HDFS distributed file system) and setup a virtual cluster; turn off replication in order to have lower storage requirement. Implement the Spark Sort application, and evaluate its performance on 1 node and 16 nodes. We must be doing strong scaling experiments (keep the dataset fixed) as we scale up from 1 node to 17 nodes.

We were also supposed to measure the time to sort on the 1 GB & 10GB datasets on 1 node Spark and 1 GB & 100 GB datasets on 17 node Spark cluster. Save the first 10 lines of output from the Spark Sort application, as well as the last 10 lines of output, for each dataset, in Sort-Spark-10GB.txt & Sort-Spark-100GB.txt respectively.

b. Performance

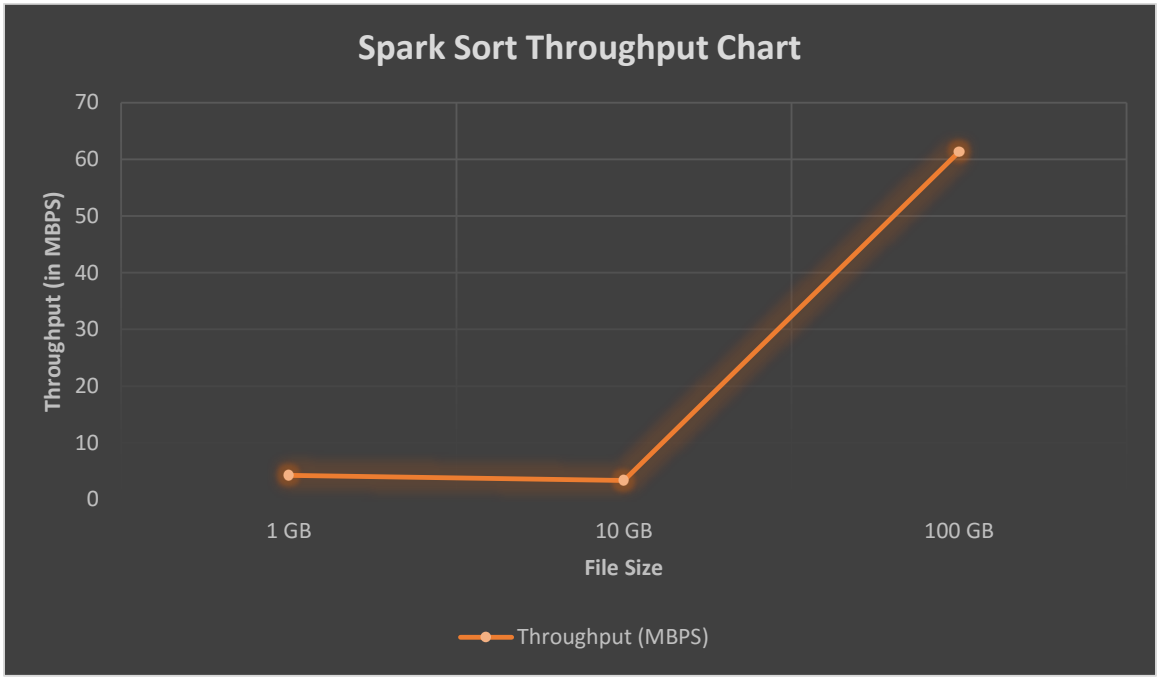
i. Spark Sort Execution

	1 GB	10 GB	100 GB
Execution Time (sec)	236	3000	236



ii. Spark Sort Throughput

	1 GB	10 GB	100 GB
Throughput (MBPS)	4.464286	10.81081	61.29705



(assuming 1,000,000 Bytes = 1MB)

c. Methodology

- Spark Sorting program is written in Python (1 GB & 10 GB) and in Scala (100 GB)
- The Map function converts each individual line (consisting of 100 characters) into Key – Value pair (Key consisting of 10 characters and Value consisting of remaining characters) and stores it into RDD
- The RDD from above step will be sorted on Key using *sortByKey* function
- Once the RDD is sorted based on the Key then RDD output can be combined and saved as text file as final output

d. Runtime Environment Settings

- Operating System
  Ubuntu 14.04.3 LTS, Release: 14.04, Codename: trusty
- Operating System Kernel

Linux ip-172-31-60-162 3.13.0-74-generic #118-Ubuntu SMP Thu Dec 17 22:52:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux

- Spark
  Spark version 1.6.0
  Using Scala version 2.10.5 (OpenJDK 64-Bit Server VM, Java 1.7.0_99)
- Java
  java version "1.7.0_95"
  OpenJDK Runtime Environment (IcedTea 2.6.4) (7u95-2.6.4-0ubuntu0.14.04.2)
  OpenJDK 64-Bit Server VM (build 24.95-b01, mixed mode)
- Python
  Python 2.7.6 (default, Jun 22 2015, 17:58:13), [GCC 4.8.2] on linux2

e. Setup & Execution Process

i. Setting up Single Node Cluster for Spark

- Download and extract the latest Spark version
- Existing HDFS (if any from Hadoop installation) can be used directly

ii. Setting up 17 Node Cluster for Hadoop

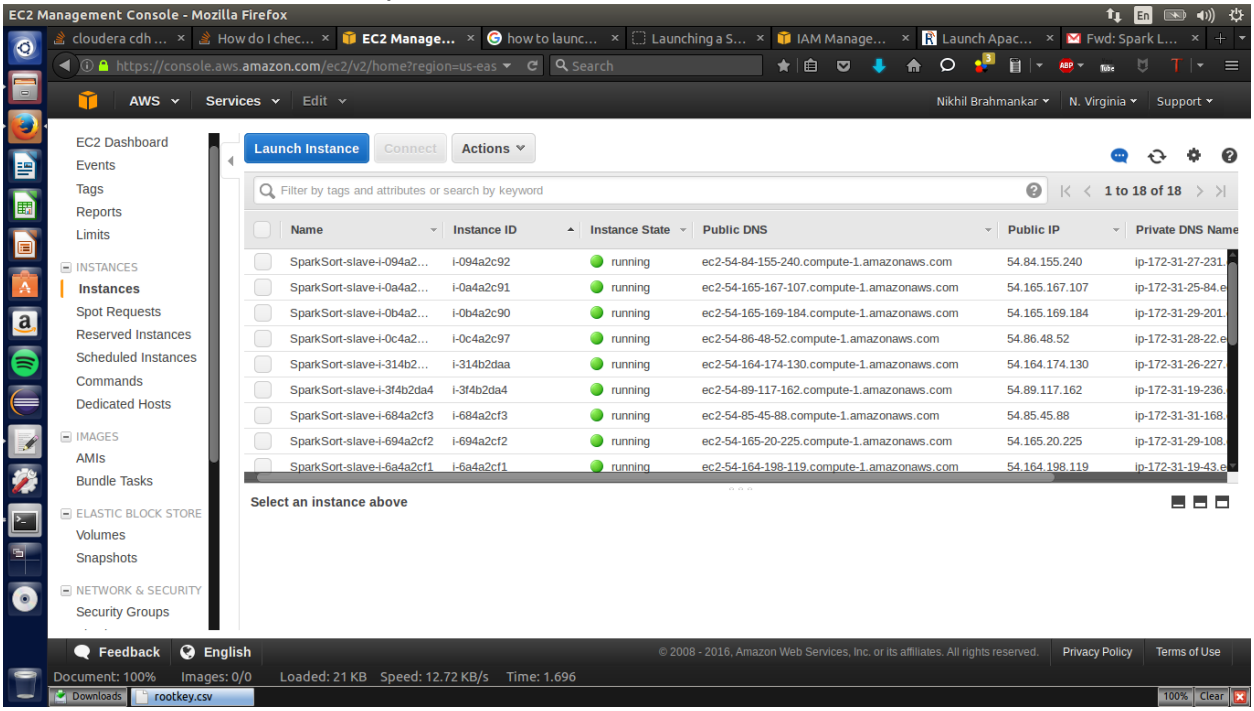
- Export the AWS Access Key and Secret Access Key
- Spark Cluster can be launched using the below given command
  ./spark-ec2 -k <key name> -i <PEM file name> -s <no. of instances> -t <instance type> --spot-price <spot price> launch <Cluster name>

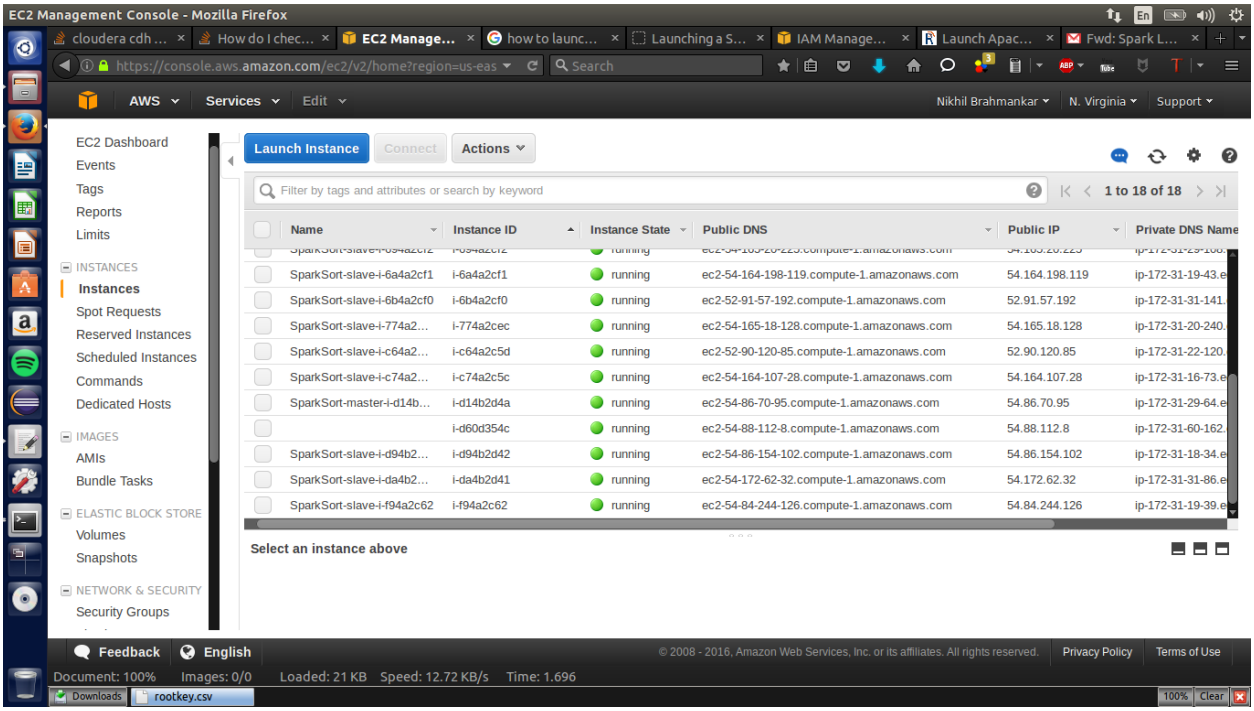
iii. Execution Process

- Copy the *"SparkSort.py"* and "*SparkSort.scala*" in Spark directory
- Execute the above mentioned Python code using "*spark-submit*" command with Input File Path (HDFS) and Output Folder Path (HDFS) to start the sorting process.
- Scala file can be executed through Spark Shell using command "*spark-shell -I <scala file name>*". The input file name and output path are hardcoded into program.

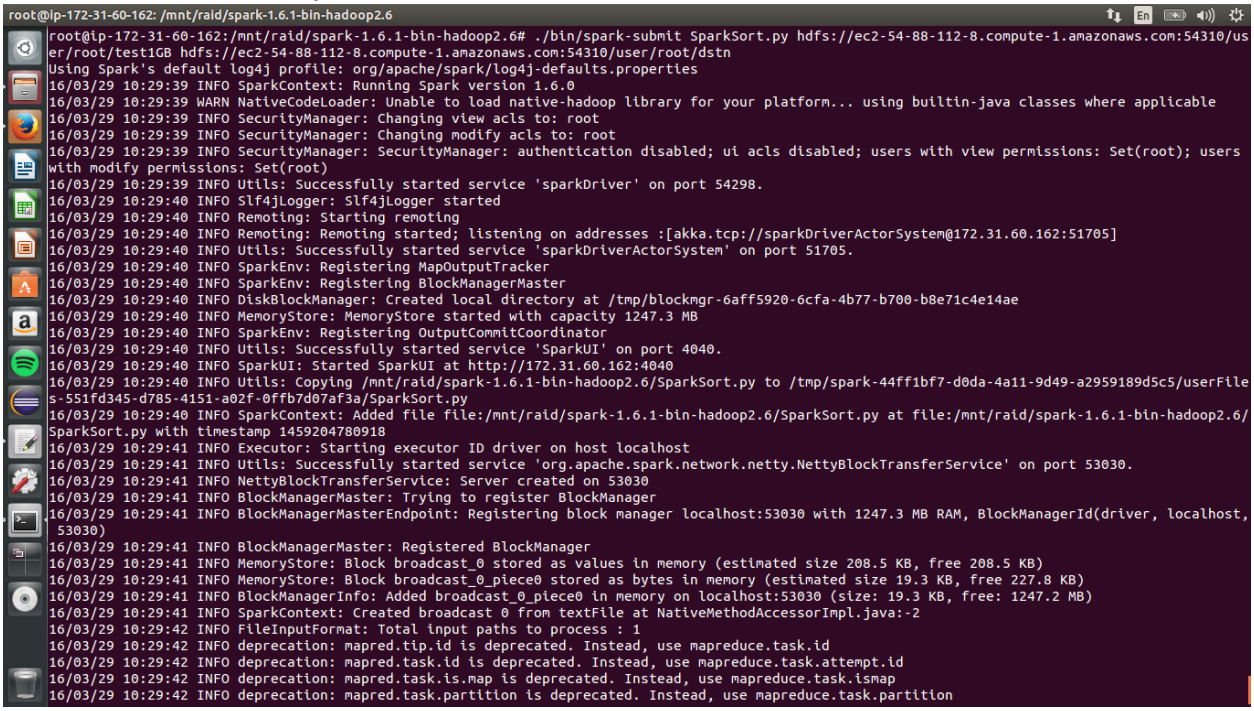
f. Output Screenshots

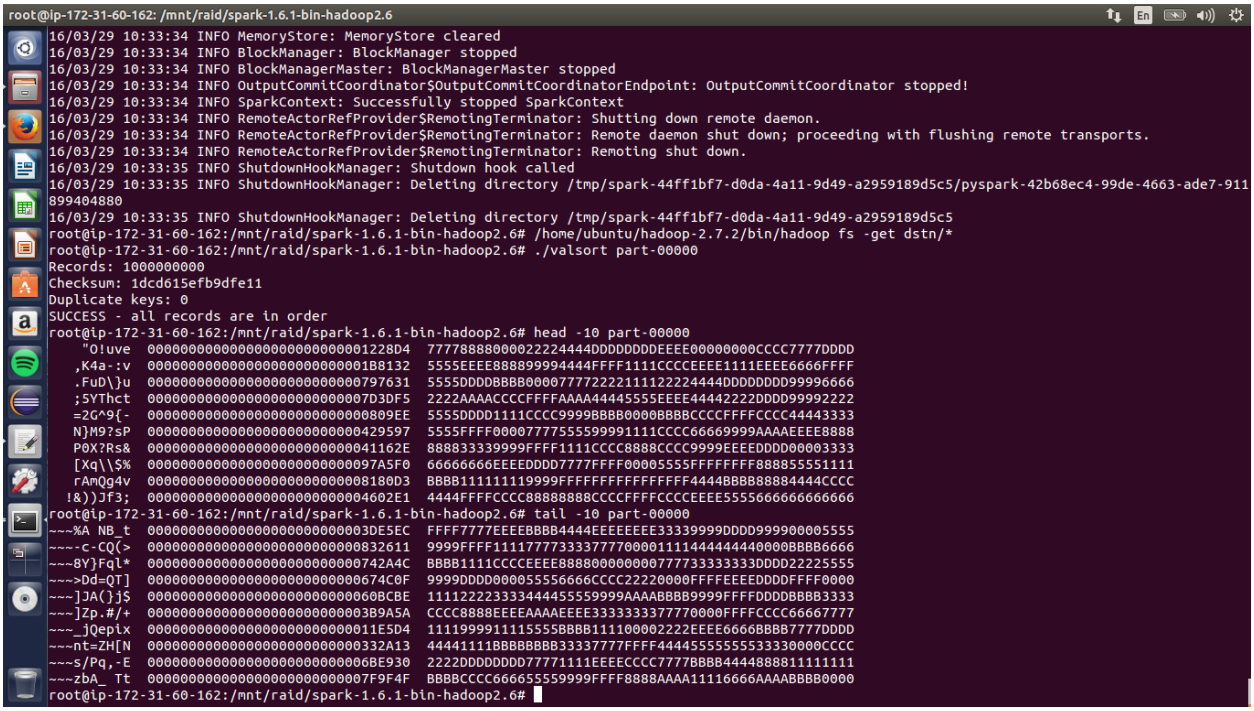
i. Hadoop AWS Instances Screenshots



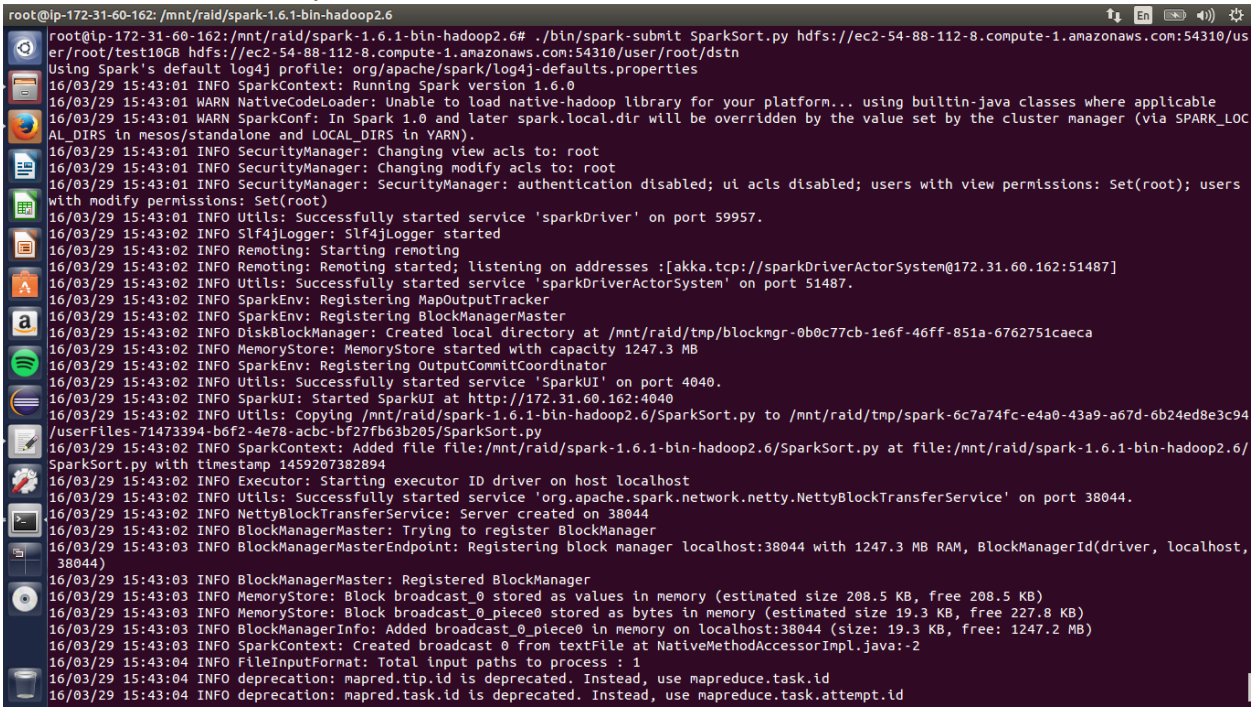


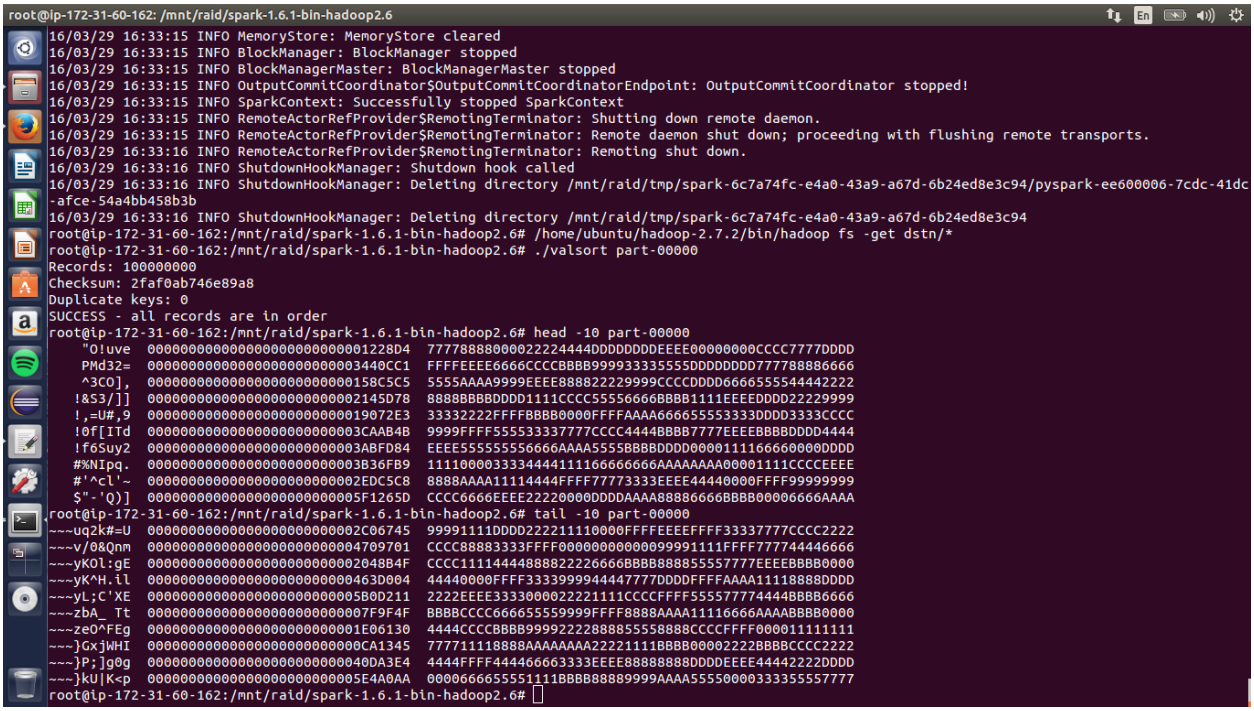
ii. Spark 1 GB 1 Node Sort



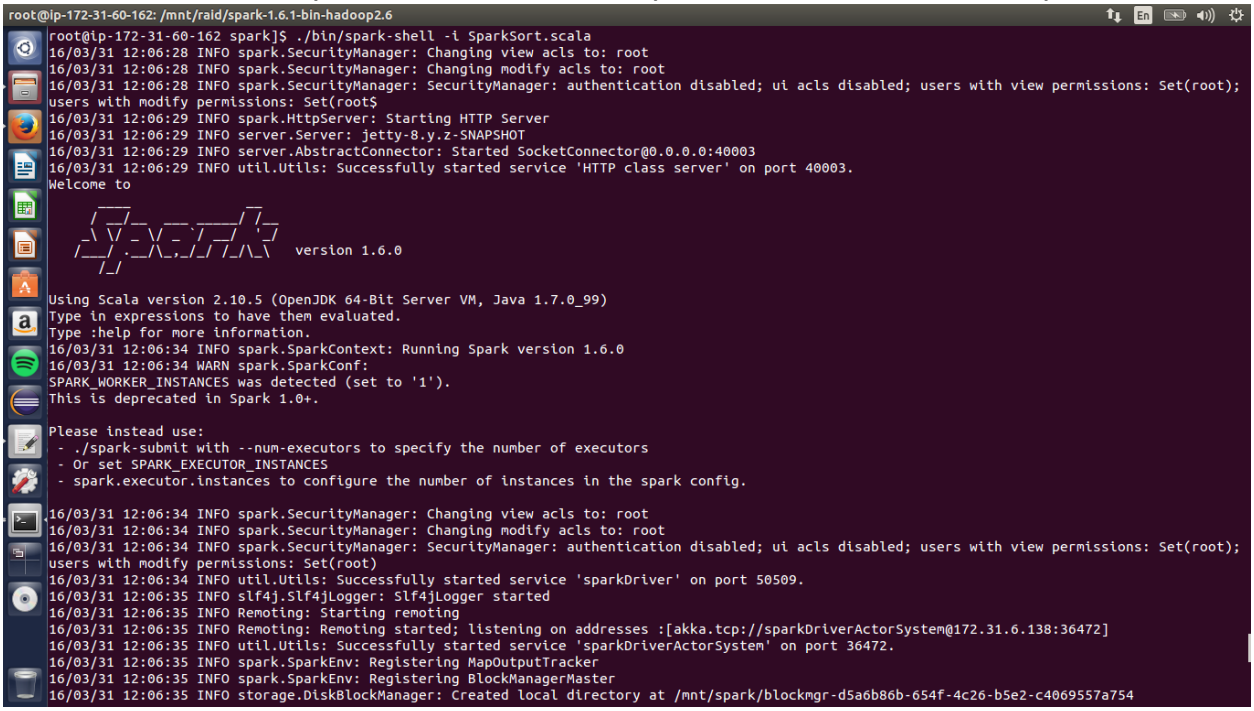


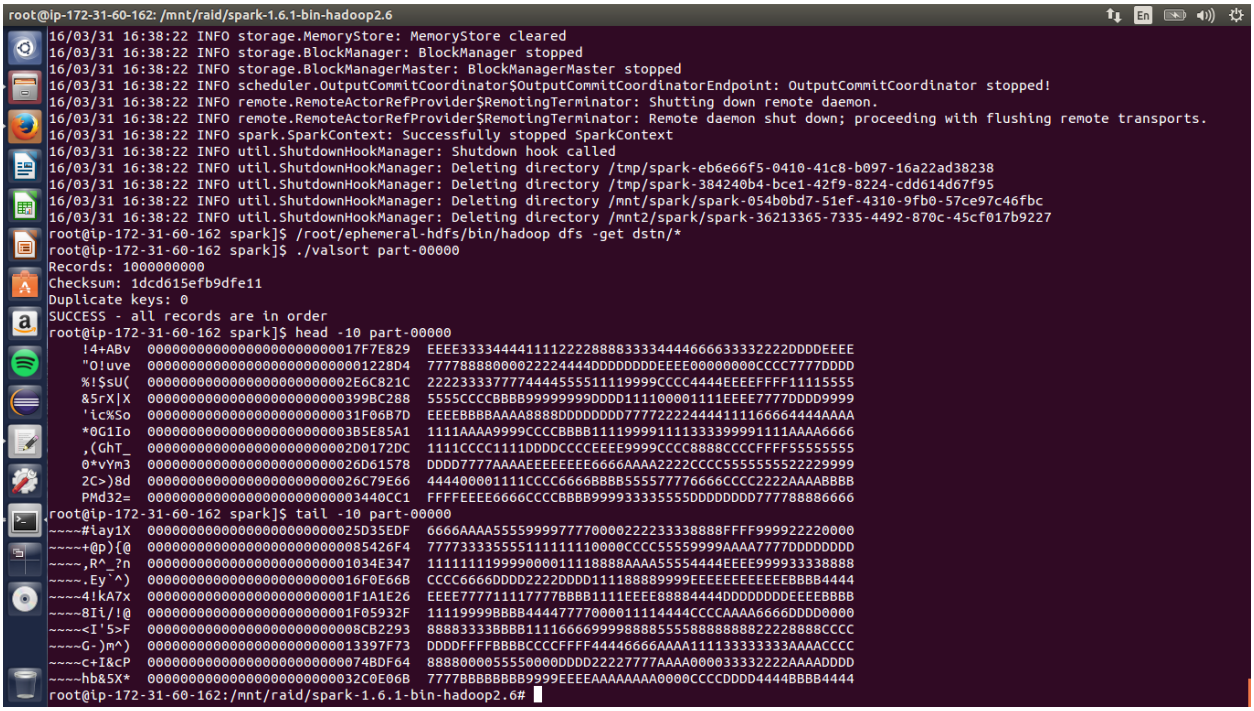
iii. Spark 10 GB 1 Node Sort





iv. Spark 100 GB 17 Nodes (1 Master + 16 Slave Nodes) Sort





g. Output Files

Output files “Sort-spark-1GB.txt”, “Sort-spark-10GB.txt” and “Sort-spark-100GB.txt” are available in “\outputs\spark” folder inside the main folder.

h. Tradeoffs made

- Hadoop was not utilized optimum because of the lack of knowledge

i. Difficulties faced

- Spot Instances were terminated automatically and the progress made was lost
- Understanding and removing the errors by trial and error method in absence of in-depth knowledge of Hadoop

j. Possible Improvements

- Need to understand the configuration files and their properties in a better way
- Need to understand the usage of TotalOrderPartitioner, InputSampler, RandomSampler classes which may help to optimize sorting performance in better way
- Dedicated instances would have helped in more learning that reconfiguring the instances

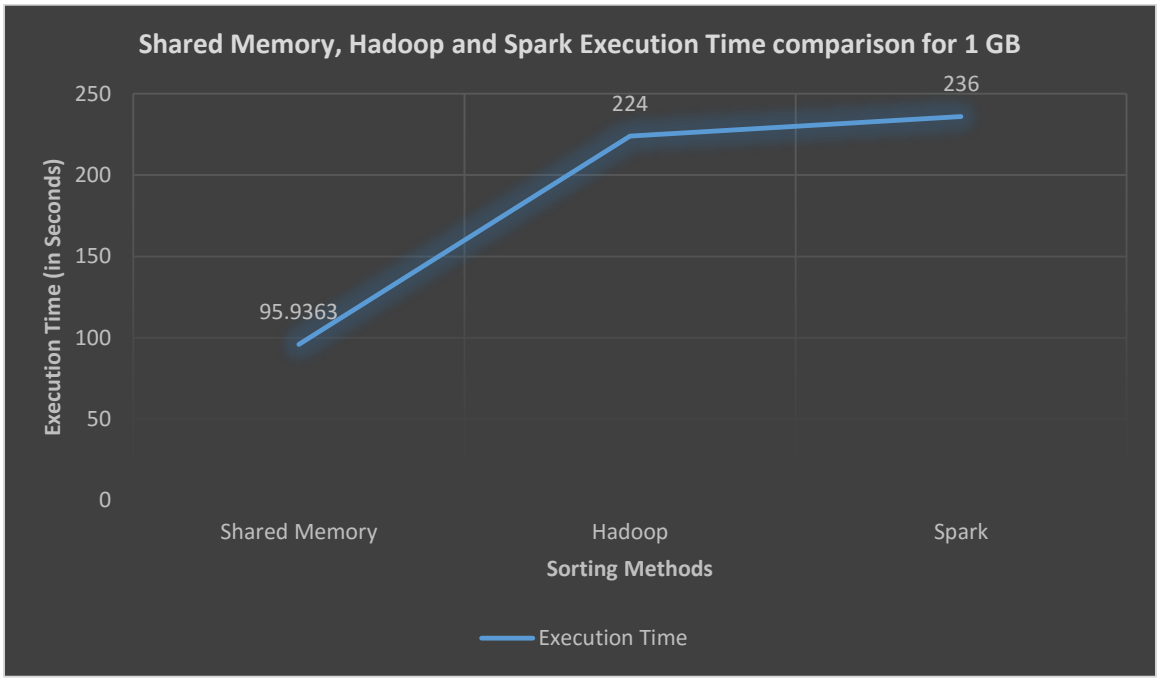
4. Performance

a. Shared Memory, Hadoop and Spark Performance comparison for 1 Node and 1 GB

i. Execution Time comparison

	Shared Memory	Hadoop	Spark
1 GB	95.9363	224	236

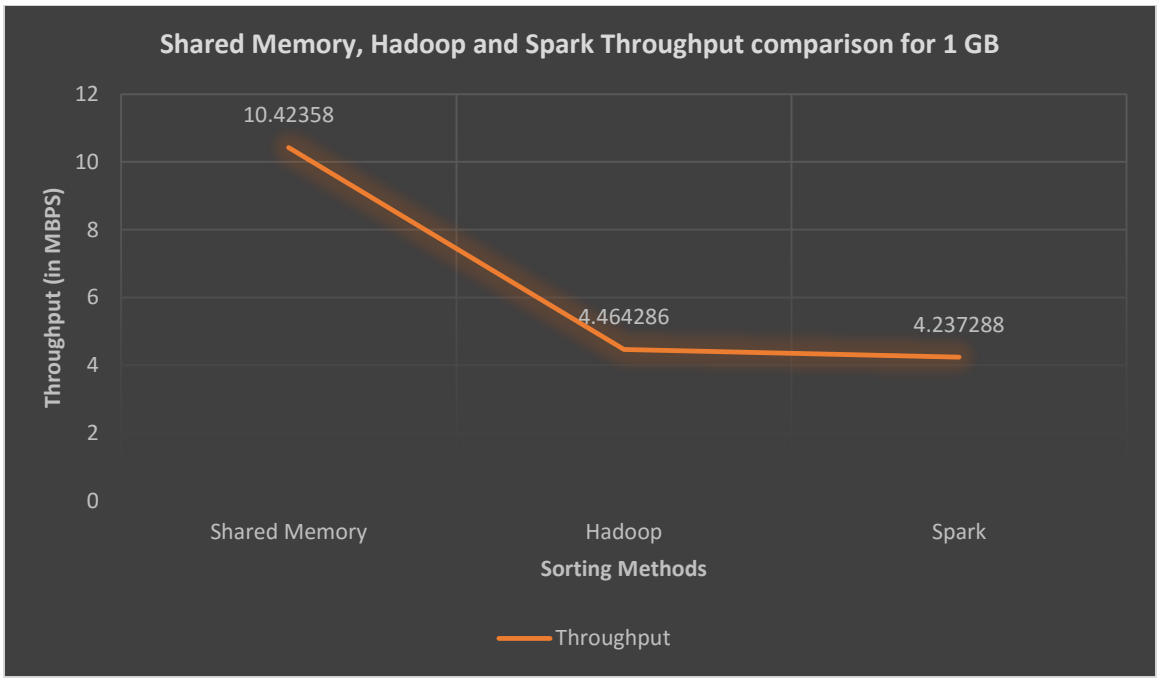
(assuming best case for 1 GB Shared Memory Sort (with 2 threads)



ii. Throughput Comparison

	Shared Memory	Hadoop	Spark
1 GB	10.42358	4.464286	4.237288

(assuming best case for 1 GB Shared Memory Sort (with 2 threads)



Performance comparison analysis

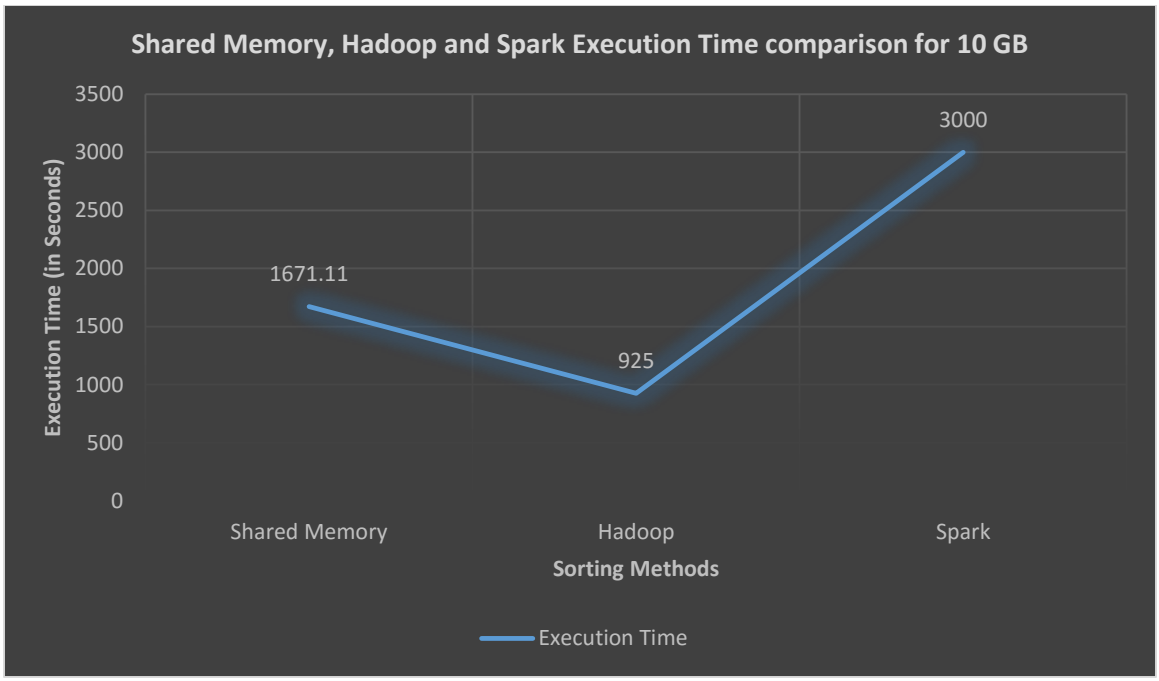
- Since the Sort Dataset Size is less than Memory Size, Shared Memory Sort proves to be the best considering the overheads of Hadoop and Spark

b. Shared Memory, Hadoop and Spark Performance comparison for 1 Node and 10 GB

i. Execution Time comparison

	Shared Memory	Hadoop	Spark
10 GB	1671.11	925	3000

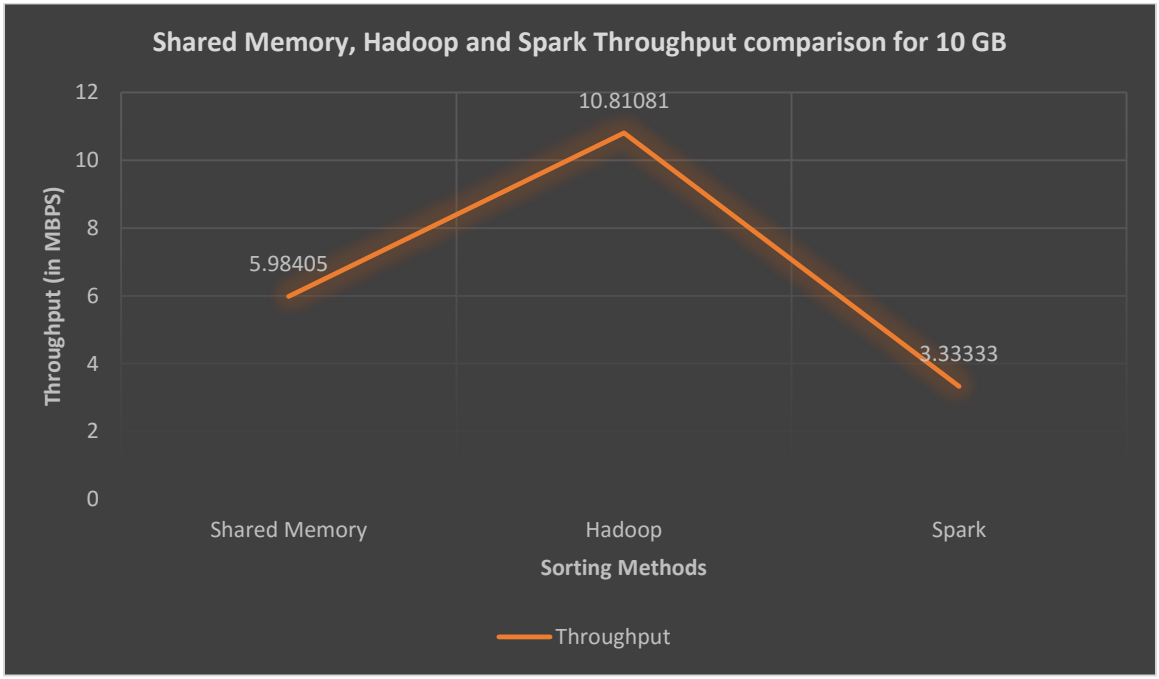
(assuming best case for 10 GB Shared Memory Sort (with 1 thread)



ii. Throughput Comparison

	Shared Memory	Hadoop	Spark
10 GB	5.98405	10.81081	3.33333

(assuming best case for 10 GB Shared Memory Sort (with 1 thread)



Performance comparison analysis

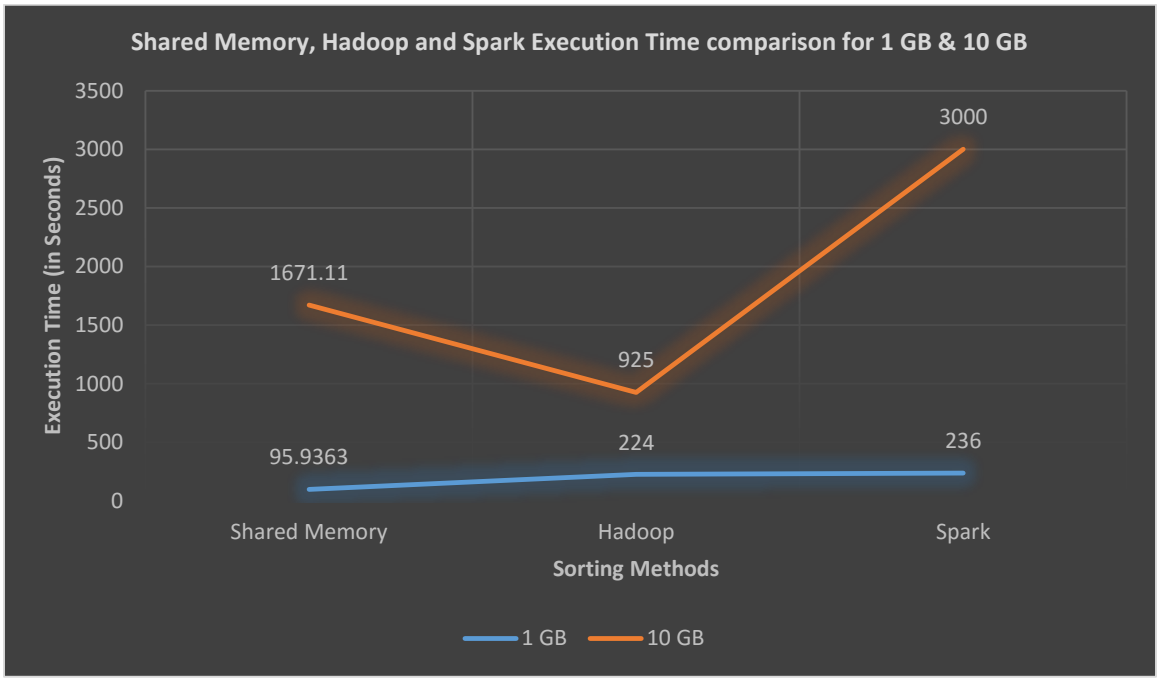
- Since the Sort Dataset Size is greater than Memory Size, Hadoop Sort proves to be the best
- Based on the analysis, Spark sorts faster than Hadoop, however Spark takes more time converting the intermediate outputs into a single file (the final merge / reduce pass)

c. Shared Memory, Hadoop and Spark Performance comparison for 1 Node (1 GB and 10 GB)

i. Execution Time comparison

	Shared Memory	Hadoop	Spark
1 GB	95.9363	224	236
10 GB	1671.11	925	3000

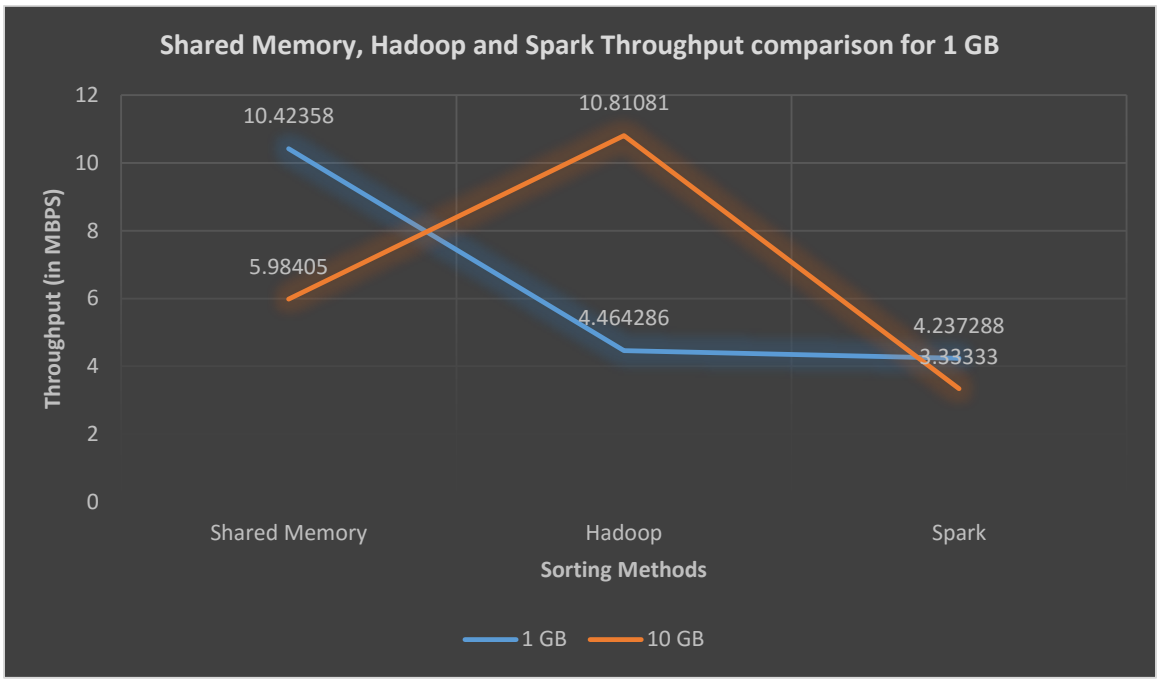
(assuming best case for 1 GB Shared Memory Sort (with 2 threads)



ii. Throughput Comparison

	Shared Memory	Hadoop	Spark
1 GB	10.42358	4.464286	4.237288
10 GB	5.98405	10.81081	3.33333

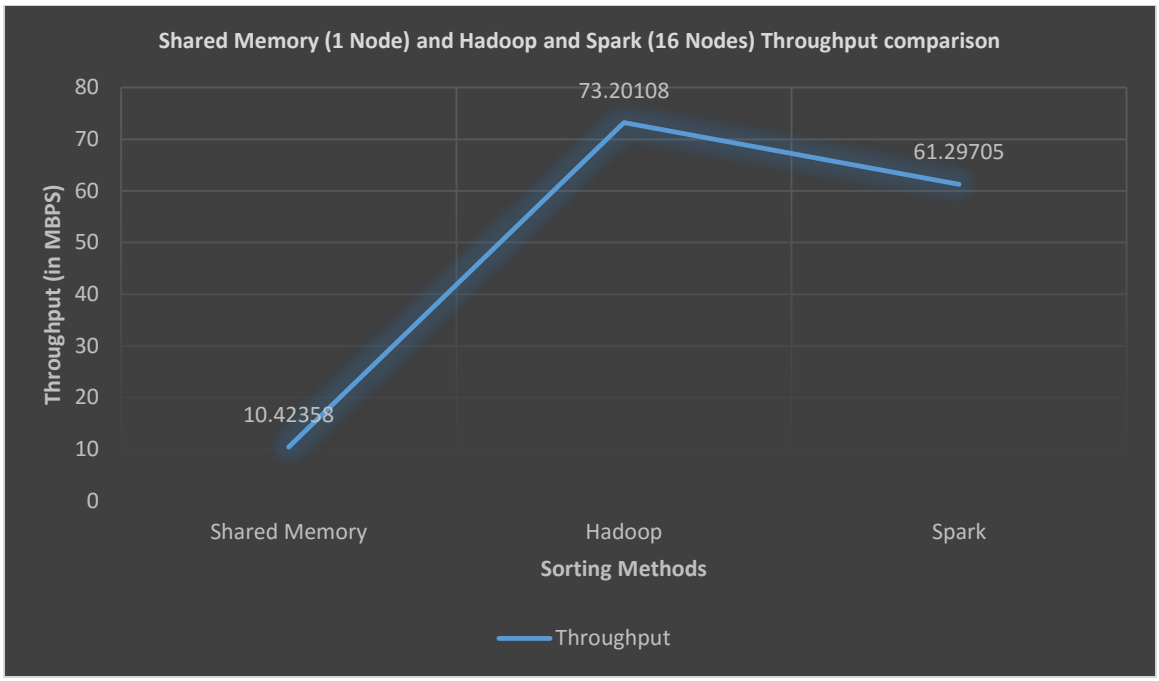
(assuming best case for 1 GB Shared Memory Sort (with 2 threads)



d. Shared Memory (1 Node) and Hadoop and Spark (16 Nodes) Performance comparison

	Shared Memory	Hadoop	Spark
Throughput	10.42358	73.20108	61.29705

(assuming best case for 1 GB Shared Memory Sort (with 2 threads)



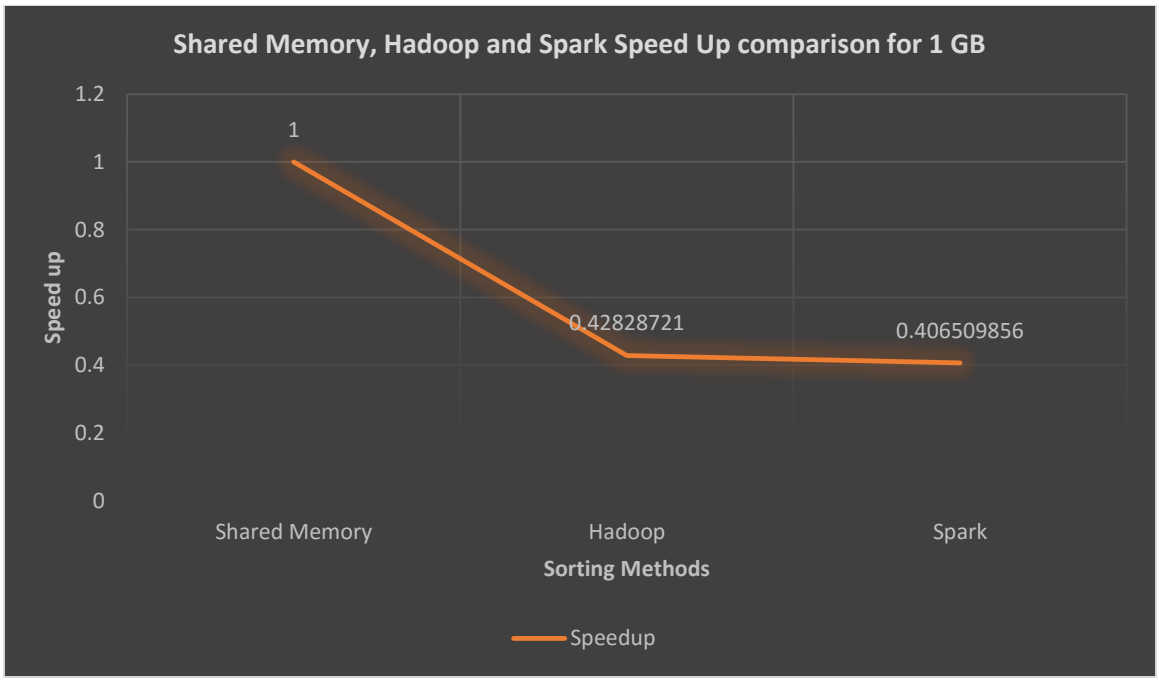
Performance comparison analysis

- Since the Sort Dataset Size is greater than Memory Size, Hadoop Sort proves to be the best
- Based on the analysis, Spark sorts faster than Hadoop, however Spark takes more time converting the intermediate outputs into a single file (the final merge / reduce pass)

e. Shared Memory, Hadoop and Spark Speedup comparison for 1 Node and 1 GB (assuming Shared Memory Throughput as base or 1)

	Shared Memory	Hadoop	Spark
1 GB Throughput	10.42358	4.464286	4.237288
Speed Up factor	1	0.42828721	0.406509856

(assuming best case for 1 GB Shared Memory Sort (with 2 threads)



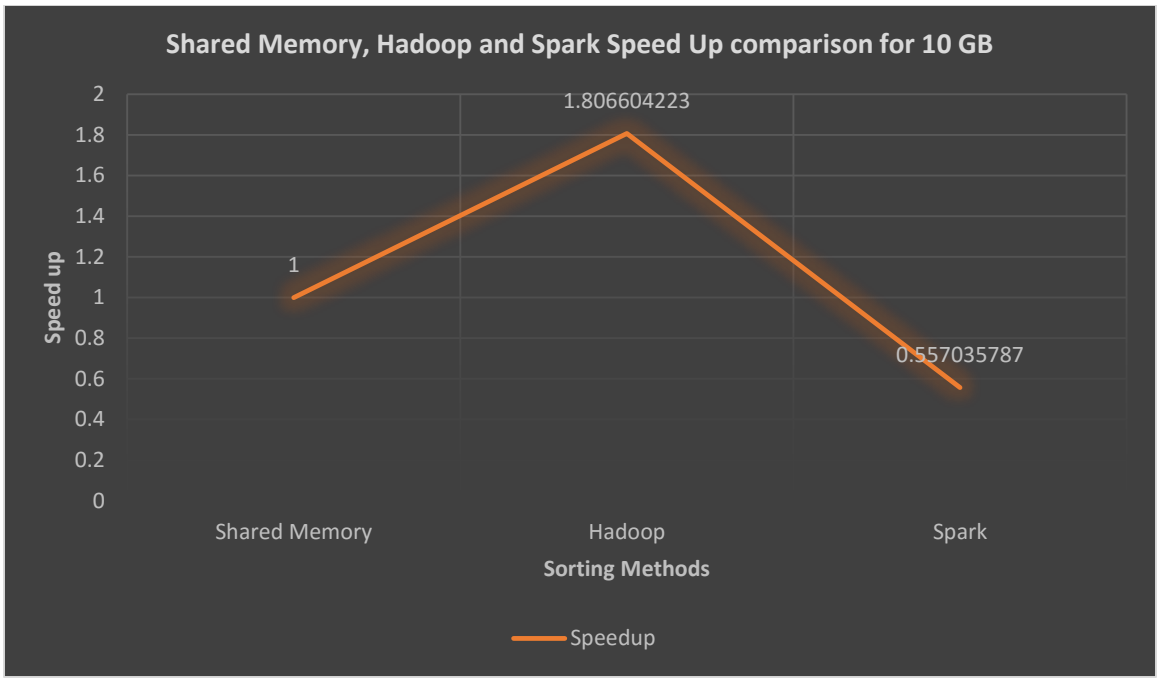
Speedup comparison analysis

- Since the Sort Dataset Size is less than Memory Size, Shared Memory Sort proves to be the best considering the overheads of Hadoop and Spark

f. Shared Memory, Hadoop and Spark Speedup comparison for 1 Node and 10 GB (assuming Shared Memory Throughput as base or 1)

	Shared Memory	Hadoop	Spark
10 GB Throughput	5.98405	10.81081	3.33333
Speed Up factor	1	1.806604223	0.557035787

(assuming best case for 10 GB Shared Memory Sort (with 1 thread)



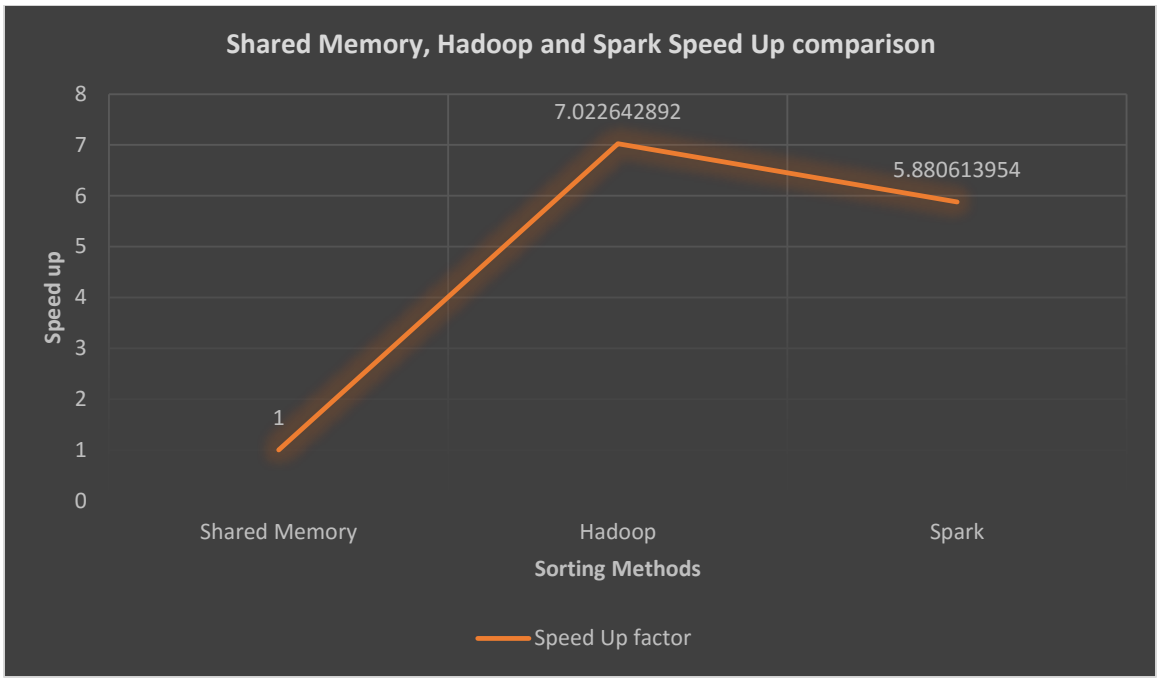
Speedup comparison analysis

- Since the Sort Dataset Size is greater than Memory Size, Hadoop Sort proves to be the best
- Based on the analysis, Spark sorts faster than Hadoop, however Spark takes more time converting the intermediate outputs into a single file (the final merge / reduce pass)

g. Shared Memory (10 GB) Speedup comparison with Hadoop and Spark (100 GB) (assuming Shared Memory Throughput as base or 1)

	Shared Memory	Hadoop	Spark
Throughput	10.42358	73.20108	61.29705
Speed Up factor	1	7.022642892	5.880613954

(assuming best case for 1 GB Shared Memory Sort (with 2 threads)

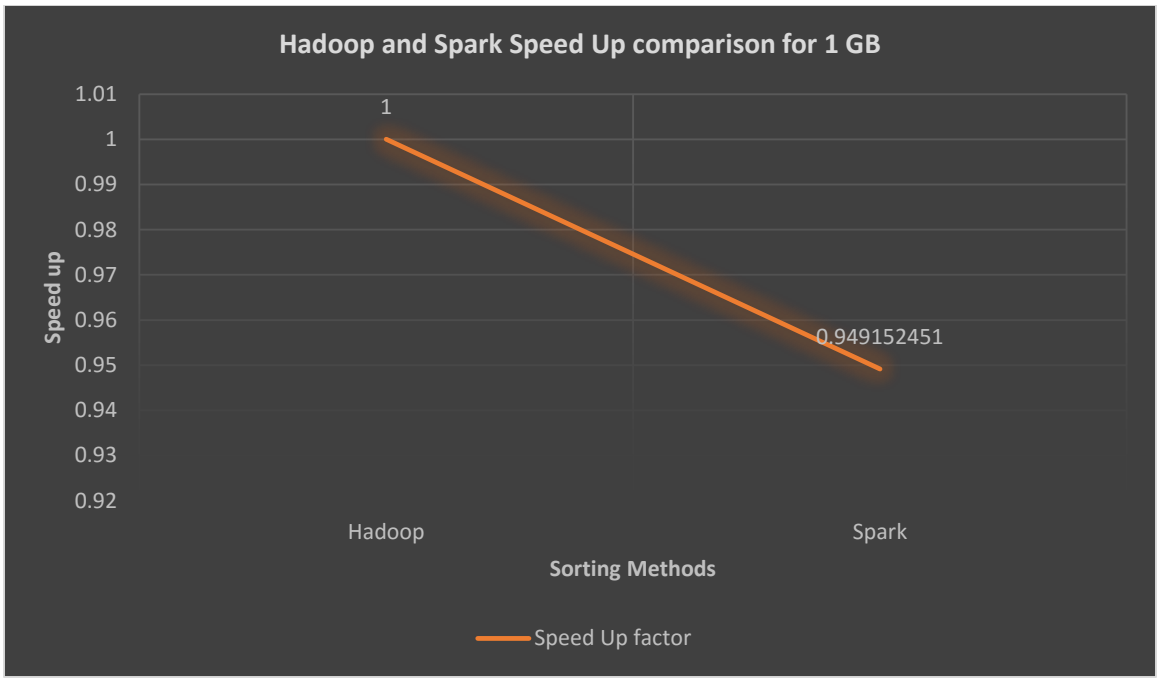


Performance comparison analysis

- Since the Sort Dataset Size is greater than Memory Size, Hadoop Sort proves to be the best
- Based on the analysis, Spark sorts faster than Hadoop, however Spark takes more time converting the intermediate outputs into a single file (the final merge / reduce pass)

h. Hadoop and Spark Speedup comparison for 1 Node and 1 GB (assuming Hadoop Throughput as base or 1)

	Hadoop	Spark
1 GB Throughput	4.464286	4.237288
Speed Up factor	1	0.949152451

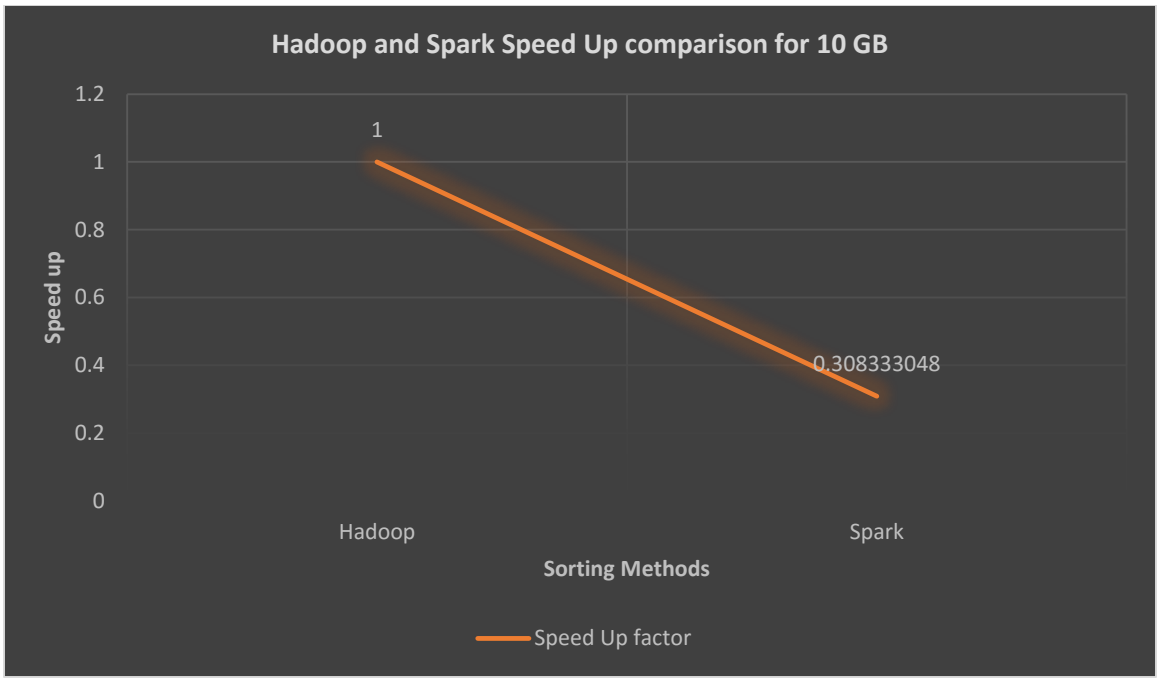


Speedup comparison analysis

- Since the Sort Dataset Size is greater than Memory Size, Hadoop Sort proves to be the best
- Based on the analysis, Spark sorts faster than Hadoop, however Spark takes more time converting the intermediate outputs into a single file (the final merge / reduce pass)

i. Hadoop and Spark Speedup comparison for 1 Node and 10 GB (assuming Hadoop Throughput as base or 1)

	Hadoop	Spark
10 GB Throughput	10.81081	3.33333
Speed Up factor	1	0.308333048

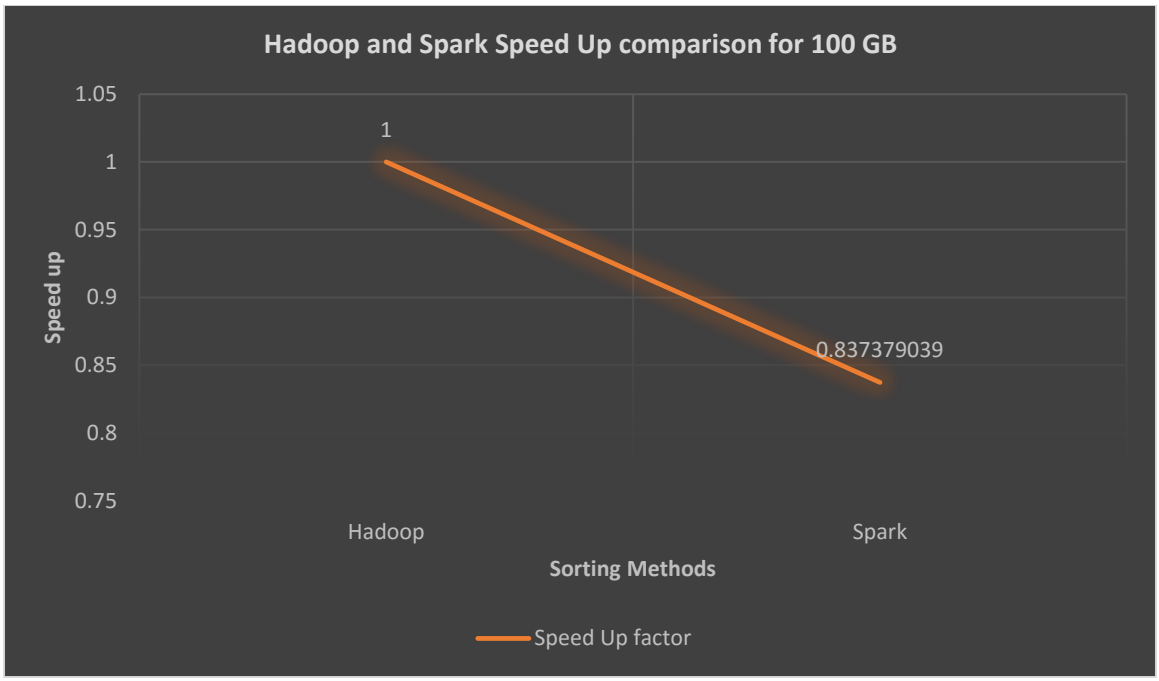


Speedup comparison analysis

- Since the Sort Dataset Size is greater than Memory Size, Hadoop Sort proves to be the best
- Based on the analysis, Spark sorts faster than Hadoop, however Spark takes more time converting the intermediate outputs into a single file (the final merge / reduce pass)

j. Hadoop and Spark Speedup comparison for 16 Nodes and 100 GB (assuming Hadoop Throughput as base or 1)

	Hadoop	Spark
100 GB Throughput	73.20108	61.29705
Speed Up factor	1	0.837379039

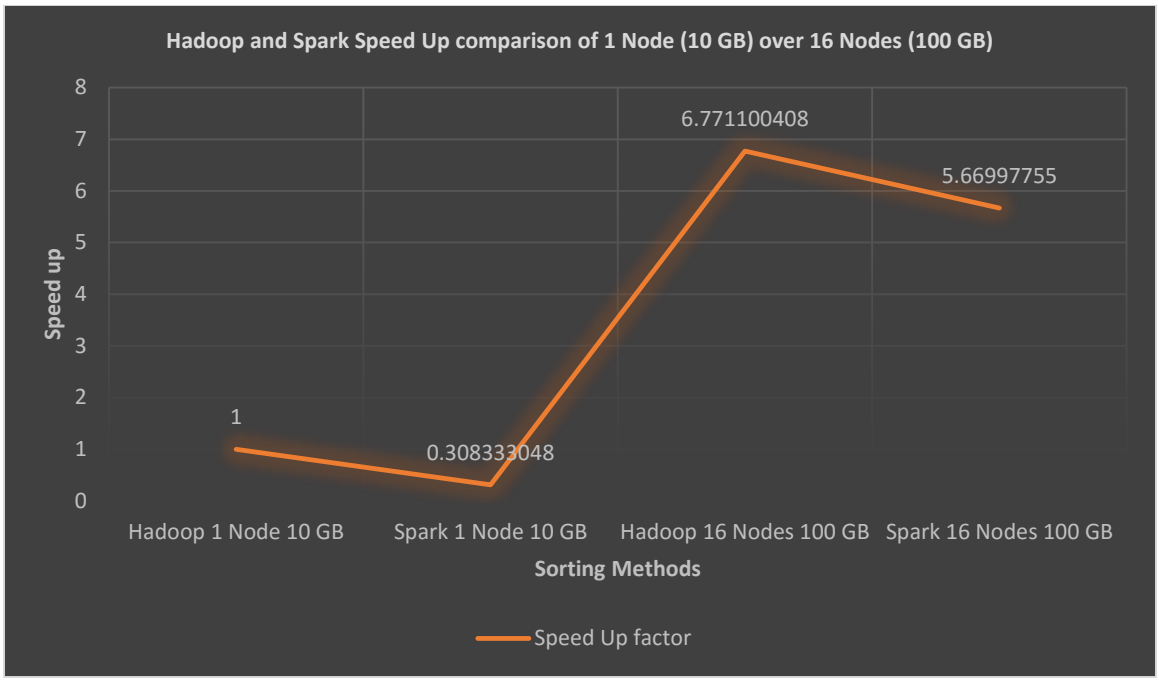


Speedup comparison analysis

- Since the Sort Dataset Size is greater than Memory Size, Hadoop Sort proves to be the best
- Based on the analysis, Spark sorts faster than Hadoop, however Spark takes more time converting the intermediate outputs into a single file (the final merge / reduce pass)

k. Hadoop and Spark Speedup comparison of 1 Node (10 GB) over 16 Nodes (100 GB) (assuming Hadoop 1 Node 10 GB Throughput as base or 1)

	Hadoop 1 Node 10 GB	Spark 1 Node 10 GB	Hadoop 16 Nodes 100 GB	Spark 16 Nodes 100 GB
Throughput	10.81081	3.33333	73.20108	61.29705
Speed Up factor	1	0.308333048	6.771100408	5.66997755



Speedup comparison analysis

- Since the Sort Dataset Size is greater than Memory Size, Hadoop Sort proves to be the best
- Based on the analysis, Spark sorts faster than Hadoop, however Spark takes more time converting the intermediate outputs into a single file (the final merge / reduce pass)

I. Conclusions

i. Which seems to be best at 1 node scale?

- If Sort Dataset Size is lesser than Memory, then Shared Memory Sort is the best option to sort the data
- If Sort Dataset Size is greater than Memory, then Hadoop or Spark is the best option to sort the data. If we are able to reduce the time taken for final Merge or Reduce pass for then Spark is the best as Spark uses in-memory sorting which are always faster than Disk based External Sorting

ii. How about 16 nodes?

- For 16 node scale, Hadoop or Spark is the best option to sort the data. If we are able to reduce the time taken for final Merge or Reduce pass for then Spark is the best as Spark uses in-memory sorting which are always faster than Disk based External Sorting

iii. Can you predict which would be best at 100 node scale?

- For 100 node scale, Hadoop and Spark both are the best options to sort the data. Spark's latest versions are able to sort 100 TB data using 206 EC2 nodes in 23 minutes. The previous world record was 72 minutes, set by Yahoo using a Hadoop MapReduce cluster of 2100 nodes. This means that Spark sorted the same data 3X faster using 10X fewer machines. All the sorting took place on disk (HDFS), without using Spark's in-memory cache

	Hadoop World Record	Spark 100 TB *	Spark 1 PB
Data Size	102.5 TB	100 TB	1000 TB
Elapsed Time	72 mins	23 mins	234 mins
# Nodes	2100	206	190
# Cores	50400	6592	6080
# Reducers	10,000	29,000	250,000
Rate	1.42 TB/min	4.27 TB/min	4.27 TB/min
Rate/node	0.67 GB/min	20.7 GB/min	22.5 GB/min
Sort Benchmark Daytona Rules	Yes	Yes	No
Environment	dedicated data center	EC2 (i2.8xlarge)	EC2 (i2.8xlarge)

* not an official sort benchmark record

iv. How about 1000 node scales?

- For 1000 node scale, Spark is the best option

m. Sort Benchmark comparison

Some of the configuration details for 2014 and 2013 winners are shared below

2014 winners

- Hadoop- 102.5 TB in 4,328 seconds
- Throughput: 23682.99 mb/sec
- 2100 nodes (2 2.3Ghz hexcore Xeon E5-2630, 64 GB memory, 12x3TB disks)

- Spark- 100 TB in 1,406 seconds
- Throughput: 71123.75 mb/sec

- 207 Amazon EC2 i2.8xlarge nodes (32 vCores - 2.5Ghz Intel Xeon E5-2670 v2, 244GB memory, 8x800 GB SSD)

2013 Winners

- Hadoop: 102.5 TB in 4,328 seconds
- Throughput: 23682.99 mb/sec
- 2100 nodes (2 2.3Ghz hexcore Xeon E5-2630, 64 GB memory, 12x3TB disks)

Conclusion

- The throughput differences are partly due to hardware configuration
- The differences are also due to the fact that we are still yet to understand and use Hadoop and Spark to its full extent

n. CloudSort Benchmark learning

CloudSort benchmark

- is used to combine various costs (the cost of equipment (hardware and software), cost of energy-use, Management and maintenance cost, etc.) to assess overall efficiency. The amortized pricing offered by various Cloud Platforms reflect all the above stated costs for profitably running these infrastructure services
- focuses on "What is the minimum cost for sorting a fixed number of records on any public cloud?"
- recommends using Public cloud in order to
    - highlight the efficacy (or deficiencies) of different cloud platforms rather than the TCO-efficiency of sort implementations.
    - enable the transition from in-house data centers to public cloud.
    - encourage and drive cloud platforms to quickly improve support for IO intensive operations.
    - be an indicator for both: the effectiveness of cloud platforms and the TCO efficiency of external sort.
- recommends using external sort because
    - it represents IO intensive workloads. It's a holistic workload that exercises memory, CPU, OS, File System, IO, Network, and Storage.
    - it's simple and therefore easy to port and optimize on cutting edge technologies. As a result, it's a technology bellwether indicating which platforms are likely to make great data pumps going forward
- Advantages of CloudSort Benchmark are

- **Accessibility:** One major concern for the larger sorts is only the people that have access to expensive cutting edge clusters at national labs, well-funded academic institutions, or large companies can afford to enter the contest. CloudSort levels the playing field because the public cloud amortizes upfront costs over time, so even groups with limited budgets can develop their implementations and run the benchmark. Moreover, given the limited cost of running the benchmark, cloud vendors may also offer to offset these costs to prove their capabilities.
- **Affordability:** A rough calculation suggests that, the cost for running a 100TB sort is manageable for groups with modest budgets. Moreover, Benchmark expect entrants to run significantly scaled down versions during development and testing.
- **Auditability:** With the previous sort benchmarks, auditing a result involves good detective work on the part of auditors. The actual runs are hardly ever reproduced. On the public cloud, entrants can easily offer a limited trial of their sort software as virtual machine images, so anyone can attempt to rerun the experiments