Part 1 Analysis:

Analysis of the time consumed while running the Vagrant Box in 1024mb RAM.

Scenario One:

1990	607

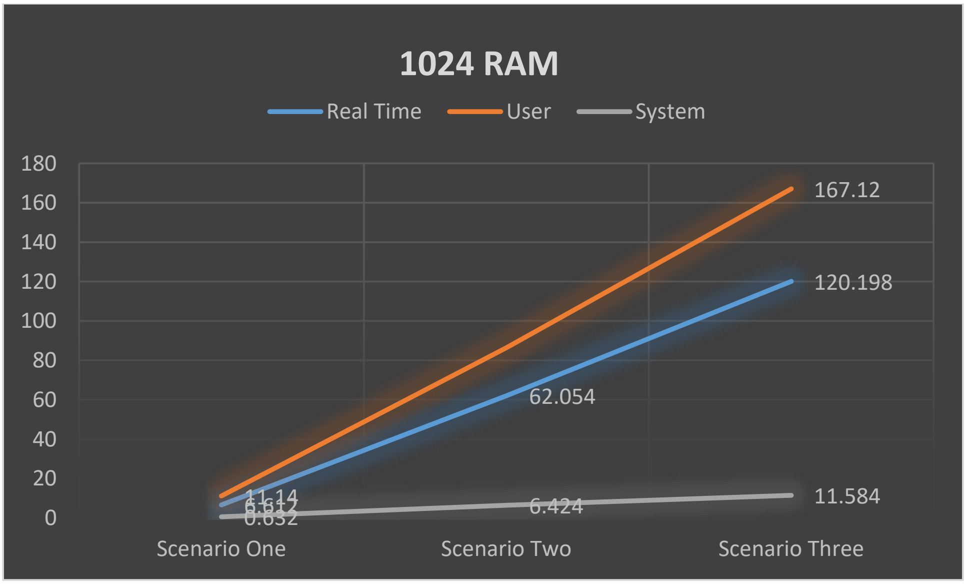
Scenario Two:

1990	607
1992	605

Scenario Three:

1990	607
1991	607
1992	605
1993	567

	Scenario One	Scenario Two	Scenario Three
Real Time	6.612	62.054	120.198
User	11.14	86.408	167.12
System	0.632	6.424	11.584



We can clearly see from the analysis graph, that the time taken in scenario one is much less than scenario two or scenario three. Scenario two is a little lesser than scenario three, this might be due to the sizes of the files involved. Scenario one uses only the 1990.gz dataset whereas scenario two uses 1990.gz and 1992.gz and scenario three uses 1990.gz,1991.gz,1992.gz and 1993.gz, hence takes the most time.

Analysis of the time consumed while running the Vagrant Box in 2048mb RAM.

Scenario One:

1990	607

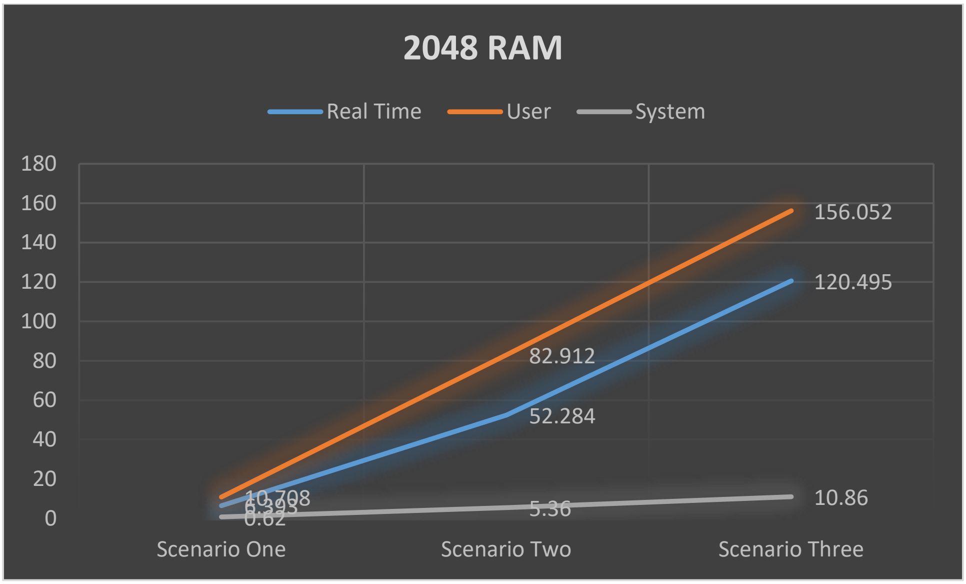
Scenario Two:

1990	607
1992	605

Scenario Three:

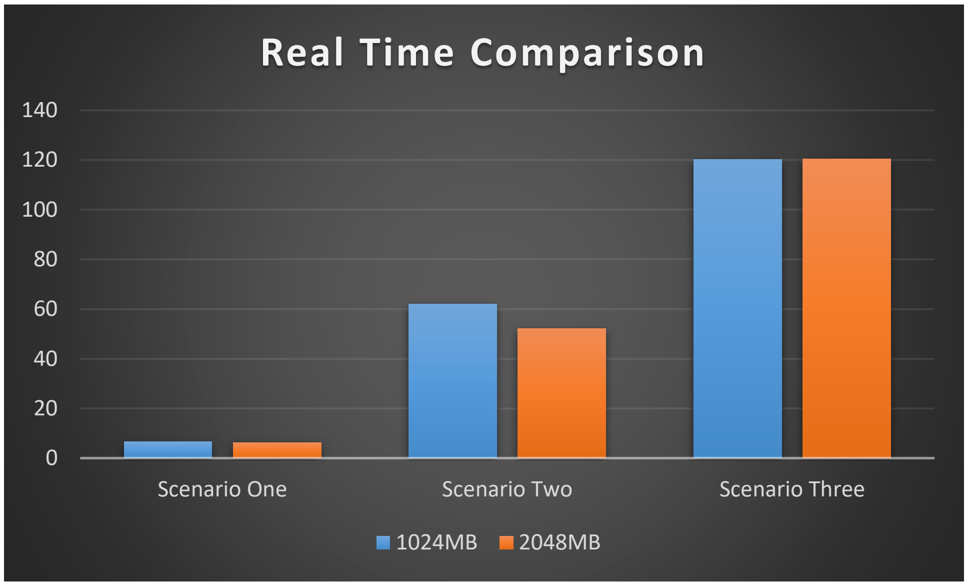
1990	607
1991	607
1992	605
1993	567

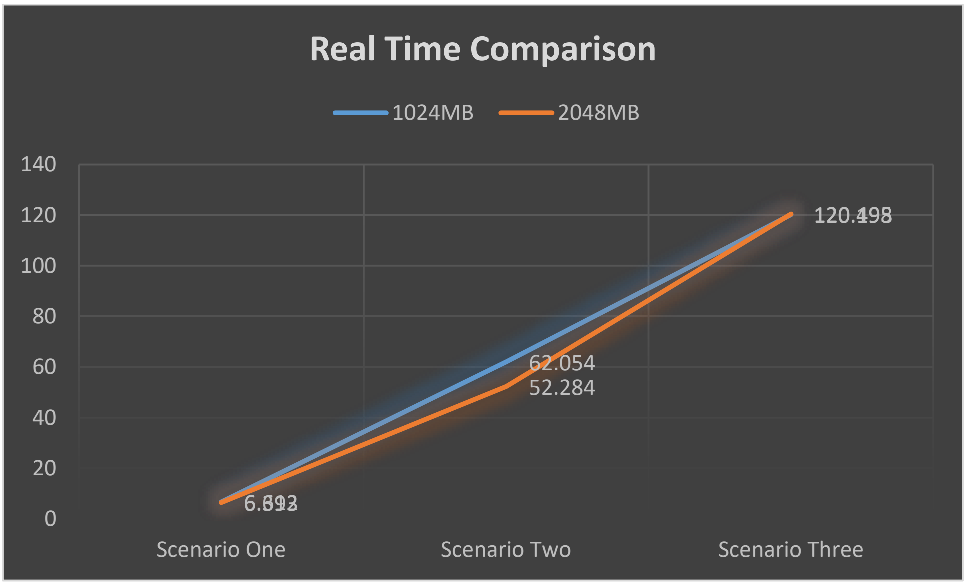
	Scenario One	Scenario Two	Scenario Three
Real Time	6.393	52.284	120.495
User	10.708	82.912	156.052
System	0.62	5.36	10.86



Even after increasing the vagrant size to 2048mb, we still observe that scenario one takes much lesser time than scenario two or scenario three. One thing I noticed is that scenario two took more time in 2048mb than in 1024mb but scenario one and scenario three took almost the same time more or less.

<u>Real Time Comparison of AWK between the two:</u>





PART 2 Analysis:

Analysis for part 2 is done by parsing the data from the GZ files. This data is parsed and then sent into the MYSQL database. After these are inserted in to the database, queries are run to find the maximum temperature and time taken.

Analysis of the time consumed while running the Vagrant Box in 1024mb RAM.

Scenario One:

1990	607

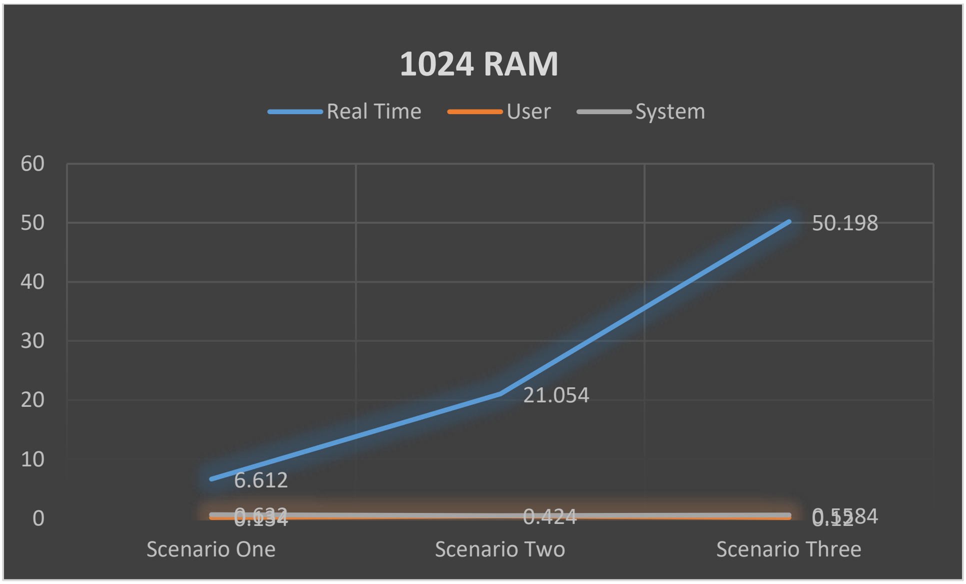
Scenario Two:

1990	607
1992	605

Scenario Three:

1990	607
1991	607
1992	605
1993	567

	Scenario One	Scenario Two	Scenario Three
Real Time	6.612	21.054	50.198
User	0.134	0.408	0.12
System	0.632	0.424	0.584



As we can see from the analysis that the real time between scenario one, scenario two and scenario three keeps increasing gradually. But User and System time is almost the same for all the scenarios.

Analysis of the time consumed while running the Vagrant Box in 2048mb RAM.

Scenario One:

1990	607

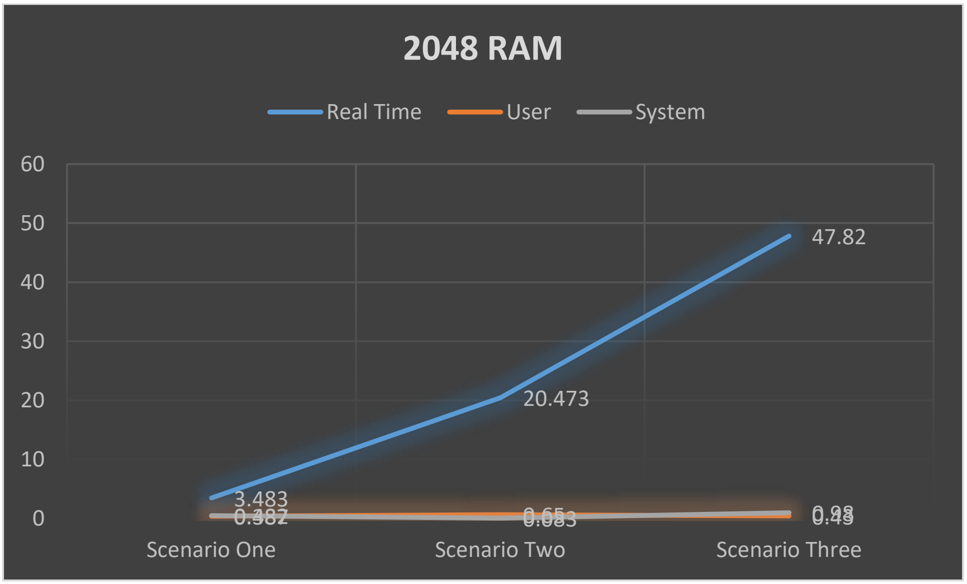
Scenario Two:

1990	607
1992	605

Scenario Three:

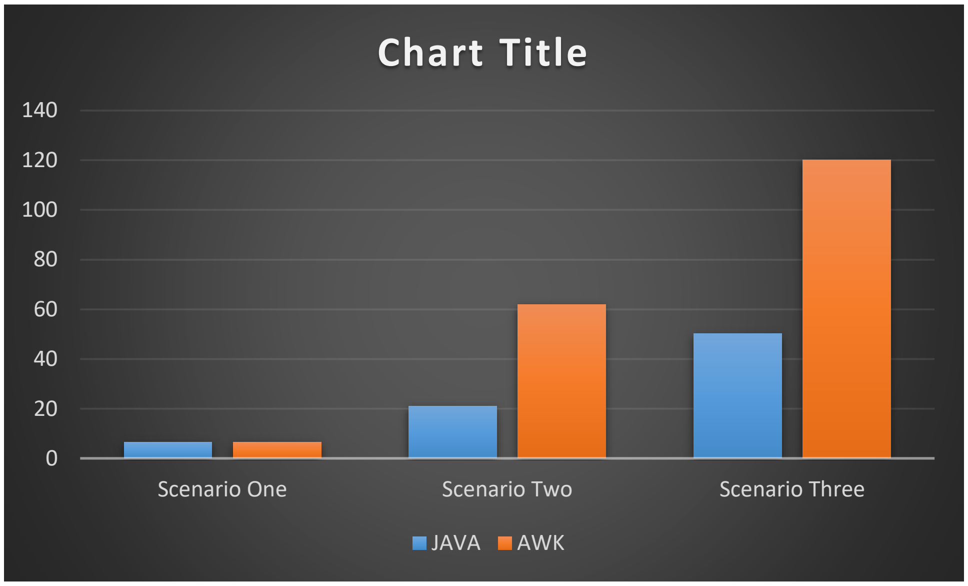
1990	607
1991	607
1992	605
1993	567

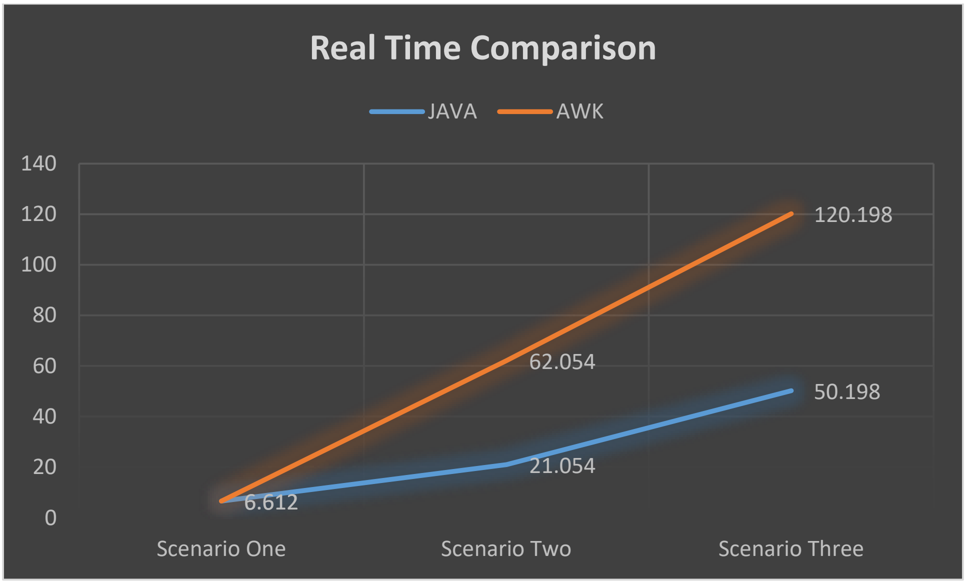
	Scenario One	Scenario Two	Scenario Three
Real Time	3.483	20.473	47.82
User	0.387	0.65	0.43
System	0.482	0.083	0.98



As we can see from the analysis graph above, the real time for 2048mb also keeps increasing gradually across all the scenarios but the user and system is almost the same for all the scenarios.

Real Time Comparison of Java between the two:

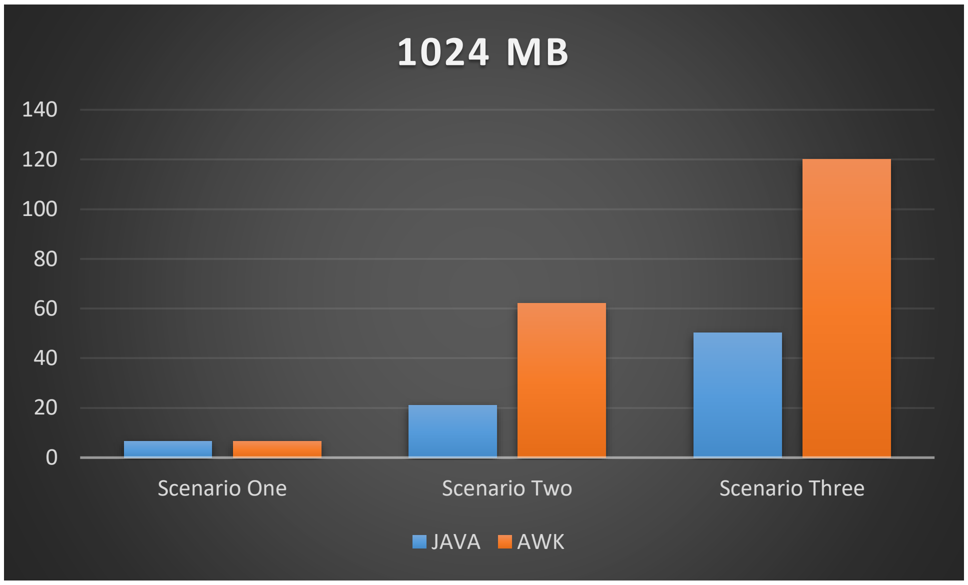




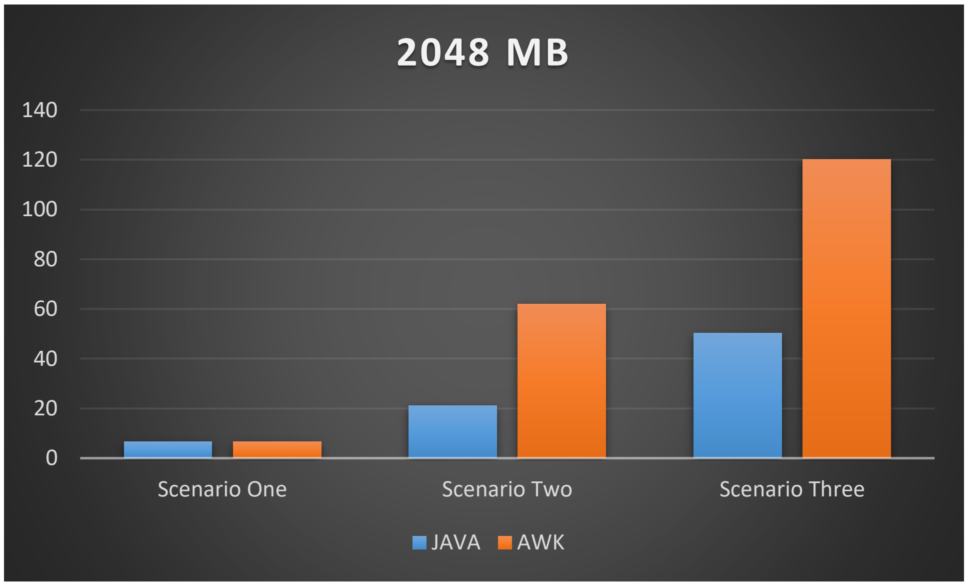
All the three scenarios have lesser real times in 2048MB than in 1024MB. Scenario was lesser by a little but scenario one and scenario three were lesser by a bigger margin.

Real time comparison of AWK and Java for 1024mb and 2048mb:

1024 MB:



2048 MB:



Sample analysis of real time between AWK and Java for 1024MB:

	JAVA	AWK
Scenario One	6.612	6.612
Scenario Two	21.054	62.054
Scenario Three	50.198	120.198

We can observe that AWK parses the data much faster than JAVA.

Java takes a longer time to parse the datasets.

On the other hand, for analytics, Java is much faster than AWK.

This seems to be the pattern for both 1024MB and 2048MB of RAM.

Thus, it depends on the basis of comparison, AWK is more efficient in parsing and Java is more efficient in running time.