Solution: As the base file should be continuously updated with the new data of varying lengths and the volume, velocity of upcoming data is unknown and expected to increase in the near future, this forms a classical problem of Big Data. The ideal case would be to use a Kafka based messaging source for the ingestion of data into the system. This data can then be processed with Kafka-Spark Streaming solution to calculate metrics on the fly and store the updated data into the base source. This solution would be end to end linearly scalable and would amortize the cost incurred in the long run over data processing without distributed systems.

Pros:

- Linearly Scalable
- Faster
- Resilient
- Can handle any amount of data.
- Cost effective for a large amount of data

Cons:

- Maintainence
- Distributed Systems means manging the entire cluster efficiently.
- Costly if the data volume is not large

For this simple exercise, the assumption is that the base file is on the disk rather than Kafka system and the spark process reads this file in memory, merges it with another file read in memory and the metrics are calculated in memory and the new file is appended to the base file. It is assumed that the first file is the base file and the second file is the new file.