**Standard deviation = (sum of (x-mean) ^2)/N, where mean = (sum of x)/N. How do you calculate the standard deviation efficiently using MR.**

This requires two jobs for execution

**First Job –**

Mapper – Iterate all items in mapper and pass it to reducer

Reducer – Iterate the numbers in reducer and add them. Maintain a local counter and increment the same in iteration.

After the iteration, take the sum and divide it by Count.

Add the result in Context.SetResource.(This is average)

**Second Job –**

Mapper – Iterate all the items and subtract the same with average we got in the previous job. And Square the result. Pass it to the Reducer

Reducer – Iterate the numbers in reducer and add them. Maintain a local counter and increment the same in iteration.

After the iteration, take the sum and divide it by Count.

*This gives standard deviation.*

**In the above, how do you find the number of values above the Standard deviation in the input file?**

This can be achieved by only mapper.

In driver class we can take input as standard deviation. Or Job configuration can take that as variable. In mapper while iterating values, we need to compare the values and write the values greater than in standard deviation in context.

**In the above, how do u find the number of values above the Standard deviation in the input file along with the difference of the standard deviation from the value from file.**

This can be achieved by only mapper.

In driver class we can take input as standard deviation. Or Job configuration can take that as variable. In mapper while iterating values, we need to compare the values and get the values greater than in standard deviation and write subtraction value in context.

**Multiple output formats have few limitations. What are they & how can you compensate those?**

There are two drawbacks to this approach.

The first is that since the number of partitions needs to be known before the job is run. A key that appears in the metadata but not in the data wastes a reducer slot. Worse, a key that appears in the data but not in the metadata doesn’t get a reducer slot—it has to be thrown away. One way of mitigating this problem would be to write a job to extract the Keys , but it’s a shame that we need an extra job to do this.

The second drawback is more subtle. It is generally a bad idea to allow the number of partitions to be rigidly fixed by the application, since it can lead to small or uneven sized partitions. Having many reducers doing a small amount of work isn’t an efficient way of organizing a job: it’s much better to get reducers to do more work and have fewer of them, as the overhead in running a task is then reduced. Uneven-sized partitions can be difficult to avoid too.

**Suppose you have 50 GB data consisting of only numbers. Find the average of all the numbers efficiently.**

It has two ways to solve the issue –

* Maintain the counter at mapper level.
* Maintain a local variable at Reducer class level and increment it at iteration

**What are all the different techniques to copy 100 TB of data from one machine to another set of 100 machines in the distributed environment? Which is the most efficient technique?**

Put, copyfromlocal

**Write M-R job to get total no of words in 1 TB of text data? How it can be optimized?**

Capture each word as Key in the mapper. And in reducer calculate by taking a local variable for Summation.

Use Stringtokenizer

Split the sentence by “ ” (space)

**Write a java program to sort 1 million records in an array? Which is the best way and how?**

If it is an array then u need to use Collection. Sort (object)

**How will you sort 50 GB file containing integer for 1 to 3? Write a pseudo code for same.**

**How will you sort 1 Billion POJO using count sort with dept id in 1 to 3?**

**POJO = Name: String**

**ADR     : String**

**Dept Id: INT [1, 2, 3]**

**Why to use layering in s\w development?**

The modular / layering approach makes bug fixing easier and development effort becomes much easier.

**How will you find top 100 integers for 1billion record list of random integers using M-R? Best approach.**

Approach asks for secondary sort.

While iteration in reducer, need to capture the top 100 key in collection.

**How will you find top 100 integers for 1billion record list of random integers using old school data structure (non Hadoop approach)? Best approach with time & space complexity.**

Tree Map is the best approach to sort