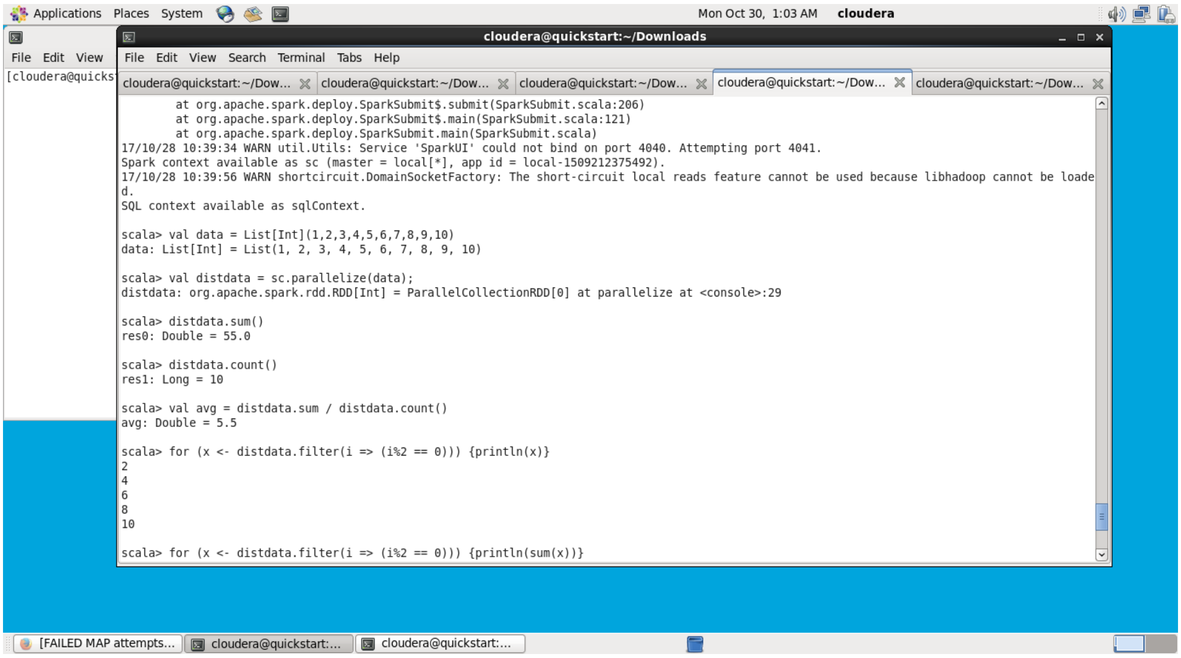
Given list of numbers - List[Int] (1,2,3,4,5,6,7,8,9,10)

- find the sum of all numbers

- find the total elements in the list

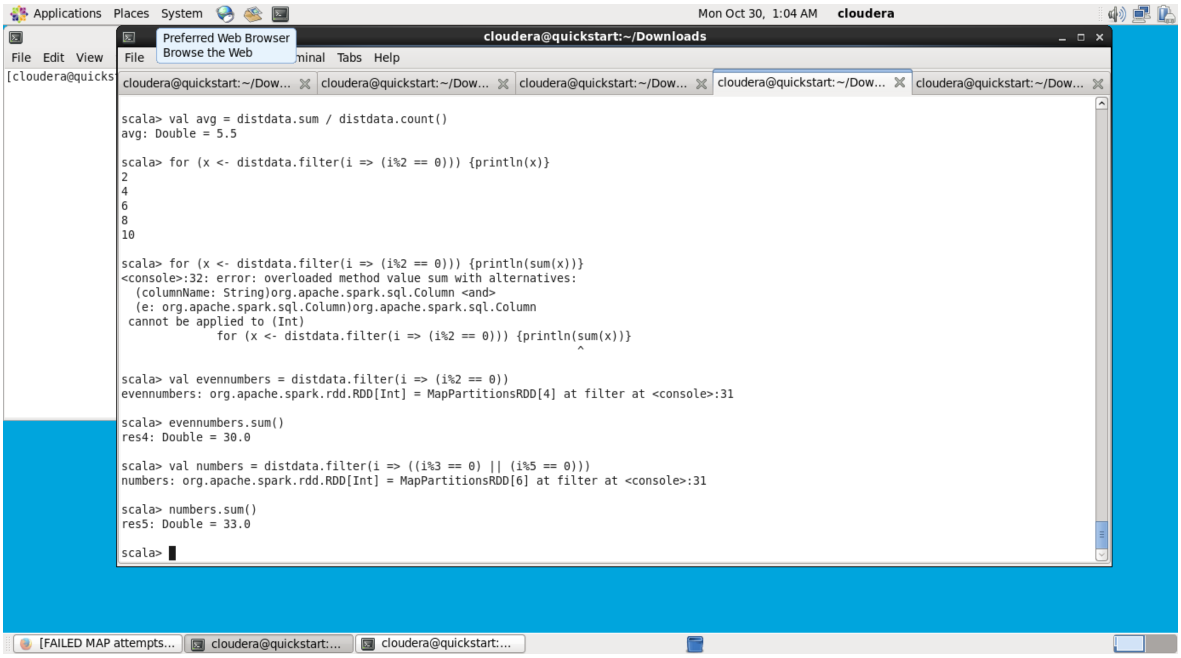
- calculate the average of the numbers in the list


Applications Places System
Mon Oct 30, 1:03 AM cloudera
File Edit View
[cloudera@quicks
cloudera@quickstart:~/Downloads
File Edit View Search Terminal Tabs Help
cloudera@quickstart:~/Dow...
cloudera@quickstart:~/Dow...
cloudera@quickstart:~/Dow...
cloudera@quickstart:~/Dow...
cloudera@quickstart:~/Dow...
[FAILED MAP attempts...
cloudera@quickstart:...
cloudera@quickstart:...

```
        at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:206)
        at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:121)
        at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
17/10/28 10:39:34 WARN util.Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Spark context available as sc (master = local[*], app id = local-1509212375492).
17/10/28 10:39:56 WARN shortcircuit.DomainSocketFactory: The short-circuit local reads feature cannot be used because libhadoop cannot be loade
d.
SQL context available as sqlContext.

scala> val data = List[Int](1,2,3,4,5,6,7,8,9,10)
data: List[Int] = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

scala> val distdata = sc.parallelize(data);
distdata: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:29

scala> distdata.sum()
res0: Double = 55.0

scala> distdata.count()
res1: Long = 10

scala> val avg = distdata.sum / distdata.count()
avg: Double = 5.5

scala> for (x <- distdata.filter(i => (i%2 == 0))) {println(x)}
2
4
6
8
10

scala> for (x <- distdata.filter(i => (i%2 == 0))) {println(sum(x))}
```

- find the sum of all the even numbers in the list

- find the total number of elements in the list divisible by both 5 and 3


Applications Places System
Mon Oct 30, 1:04 AM cloudera
File Edit View
[cloudera@quicks
cloudera@quickstart:~/Downloads
Preferred Web Browser
Browse the Web
File
minal Tabs Help
cloudera@quickstart:~/Dow...
cloudera@quickstart:~/Dow...
cloudera@quickstart:~/Dow...
cloudera@quickstart:~/Dow...
cloudera@quickstart:~/Dow...
[FAILED MAP attempts...
cloudera@quickstart:...
cloudera@quickstart:...

```
scala> val avg = distdata.sum / distdata.count()
avg: Double = 5.5

scala> for (x <- distdata.filter(i => (i%2 == 0))) {println(x)}
2
4
6
8
10

scala> for (x <- distdata.filter(i => (i%2 == 0))) {println(sum(x))}
<console>:32: error: overloaded method value sum with alternatives:
  (columnName: String)org.apache.spark.sql.Column <and>
  (e: org.apache.spark.sql.Column)org.apache.spark.sql.Column
 cannot be applied to (Int)
              for (x <- distdata.filter(i => (i%2 == 0))) {println(sum(x))}
                                                                    ^


scala> val evennumbers = distdata.filter(i => (i%2 == 0))
evennumbers: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[4] at filter at <console>:31

scala> evennumbers.sum()
res4: Double = 30.0

scala> val numbers = distdata.filter(i => ((i%3 == 0) || (i%5 == 0)))
numbers: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[6] at filter at <console>:31

scala> numbers.sum()
res5: Double = 33.0

scala> 
```