Explain the following terms in detail

-- What are the uses of counters

**Solution:**

-Counters are a useful channel for gathering statistics about the job which means it show for quality control or for application level-statistics.

-They are also useful for problem diagnosis .Counter values being much easier to retrieve than log output for large distributed jobs.  
-If we are tempted to put a log message into your map or reduce task, then it is often better to see whether you can use a counter instead to record that a particular condition occurred.

-The MapReduce framework provides Counters as an efficient mechanism for tracking the occurrences of global events within the map and reduces the phases of jobs.

-For example, a typical MapReduce job will kick off several mapper instances, one for each block of the input data, all running the same code. These instances are part of the same job, but run independent of one another.

-Counters allow a developer to track aggregated events from all of those separate instances.

-- MR Unit testing is based on

**Solution:**

MRUnit testing framework is based on JUnit and it can test Map Reduce programs written on 0.20 , 0.23.x , 1.0.x , 2.x version of Hadoop.

-- How testing is useful in industry

**Solution:**

Testing could be simply put as "Software Testing", since the field it is involved is IT. Any product/ software created requires testing before it is sold to customers, because Quality matters. Testing a product/ software not only assures that the product is 100% stable/working, but it also ensures the level of Quality.

Software testing is a process of executing a program or application with the intent of finding the software bugs.

* It can also be stated as the process of validating and verifyingthat a software program or application or product meets the business and technical requirements that guided it’s design and development, Works as expected and Can be implemented with the same characteristic.
* Software Testing can be broadly divided into 2 fields: Manual Testing and Automation Testing
* **Manual Testing** is the process of manually testing software for defects. It requires a tester to play the role of an end user and use most of all features of the application to ensure correct behavior.
* **Automation** **Testing** uses special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes.

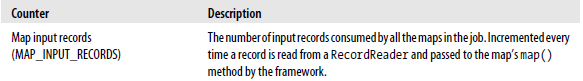
-- Mapreduce Task Counters,File system counters,Job Counter

**Solution:**

**Mapreduce Task Counters**

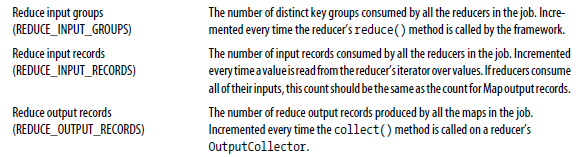
Task counters gather information about tasks over the course of their execution, and the results are aggregated over all the tasks in a job. For example, the MAP\_INPUT\_RECORDS counter counts the input records read by each map task and aggregates over all map tasks in a job, so that the final figure is the total number of input records for the whole job. Etc

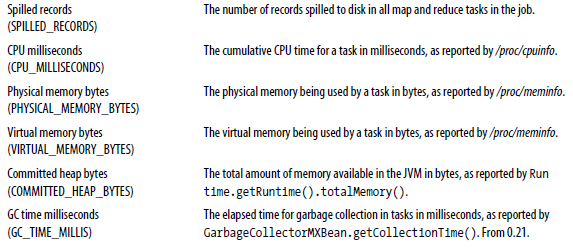
Below are the list of important Task counters maintained by Hadoop:

****

****

****

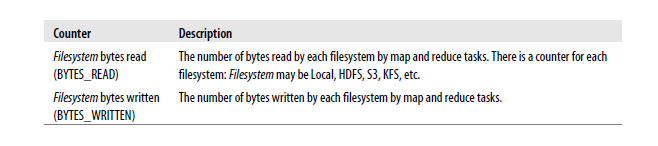
****



**File system counters**

File system counterss track 2 main details , number of bytes read by the file system and number of bytes written.

Below are the name and description of the file system counters:

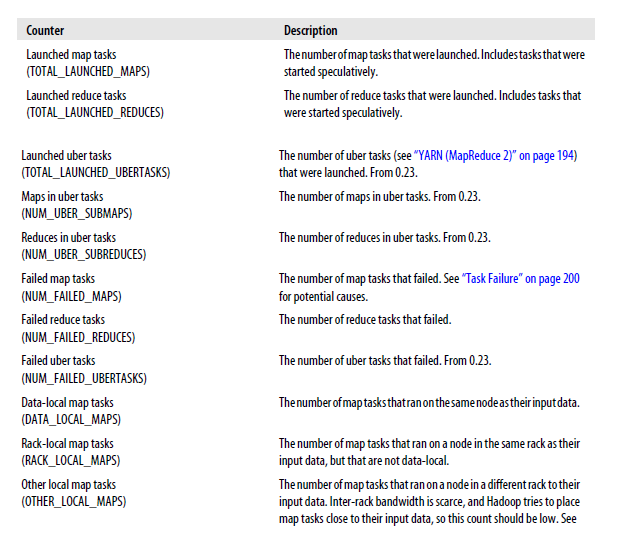


**Job Counters**

Job counters are maintained by the jobtracker (or application master in

YARN), so they don’t need to be sent across the network, unlike all other counters, including user-defined ones. They measure job-level statistics, not values that change while a task is running. For example, TOTAL\_LAUNCHED\_MAPS counts the number of map tasks that were launched over the course of a job (including ones that failed).

Below are the list of important Job counters maintained by Hadoop:



-- Raw comparator VS Writable Comparator

**Solution:**

|  |  |
| --- | --- |
| Raw comparator | Writable Comparable |
| Keys are compared using their corresponding raw bytes. | Keys are compared using you will deserialized objects. |
| It is faster as compared to Writable Comparable | It is slower as compared to Raw comparator. |

-- Partitioner, Sort comparator, Group comparator

**Solution:**

Group Comparator

– It decides which map output keys will be united (grouped) into one key, and of course all collections of values will be grouped too. Usually it takes a first key as the only one for summary collection.

Partitioner

– It is used to decide the which key should go to which reducer, by default it uses the hash code of the object to decide the reducer but one can override the partitioner to send particular to particular reducer. This is mostly used in case of composite key, secondary sort

Sort Comparator

- Used to define how map output keys are sorted,SortComparator decides how mapoutput keys are sorted. If the property mapred.output.key.comparator.class is set, either explicitly or by calling setSortComparatorClass() on Job, then an instance of that class is used. (In the old API the equivalent method is setOutputKeyComparatorClass() on JobConf.)