

RandomWriter

RandomWriter example writes 10 gig (by default) of random data/host to DFS using Map/Reduce.

Each map takes a single file name as input and writes random BytesWritable keys and values to the DFS sequence file. The maps do not emit any output and the reduce phase is not used.

The specifics of the generated data are configurable. The configuration variables are:

Name	Default Value	Description
test.randomwriter.maps_per_host	10	Number of maps/host
test.randomwrite.bytes_per_map	1073741824	Number of bytes written/map
test.randomwrite.min_key	10	minimum size of the key in bytes
test.randomwrite.max_key	1000	maximum size of the key in bytes
test.randomwrite.min_value	0	minimum size of the value
test.randomwrite.max_value	20000	maximum size of the value

This example uses a useful pattern for dealing with Hadoop's constraints on InputSplits. Since each input split can only consist of a file and byte range and we want to control how many maps there are (and we don't really have any inputs), we create a directory with a set of artificial files, each of which contains the filename that we want a given map to write to. Then, using the text line reader and this "fake" input directory, we generate exactly the right number of maps. Each map gets a single record that is the filename, to which it is supposed to write its output.

To run the example, the command syntax is

```
bin/hadoop jar hadoop-examples.jar randomwriter <out-dir> [<configuration file>]
```

[RandomWriter](#) supports generic options : see [DevelopmentCommandLineOptions](#)