

# Using IOzone for Linux disk performance analysis

Article Number: 334 | Rating: Unrated | Last Updated: Wed, Jul 25, 2018 2:09 PM

Who never had to measure the disk performances on a system?

Speaking about Linux performance monitoring includes a significant part of disk performances. If you are facing applications or databases running slower than expected or than on other environments, you may take a look on the I/O performances.

This posting aims to provide an introduction to a pretty interesting tool: IOzone.

Under Linux environments, there are plenty of tools to perform I/O tests, stretching from the simplest and most common ones (i.e.: iostat) to more evolved (complicated?) ones. You may already have heard about bonnie++ or Flexible I/O, which is presented on Yann Neuhaus' blog ([see here](#)). Even Oracle provides its own tool Orion (see [Gregory Steulet's blog](#)) to evaluate disk performances.

## IOZone overview

IOzone ([www.iozone.org](http://www.iozone.org)) is an open source solution for file system benchmark. It was originally developed by William Norcott and has been enhanced by Don Capps. In 2007, IOzone has been rewarded the Inforworld BOSSIE award (<http://www.infoworld.com/infoworld-bossie-awards-755>) of the best file I/O tool. IOzone presents the advantage to be available for almost all main platforms such as Linux, AIX, Solaris, BSD, MacOSX, and also Windows through Cygwin.

Basically IOzone runs the following tests on a file, which is then split in smaller slices:

- ID 0 : Read – indicates the performance of reading a file that already exists in the filesystem.
- ID 1 : Write – indicates the performance of writing a new file to the filesystem.
- ID 2 : Re-read – after reading a file, this indicates the performance of reading a file again.
- ID 3 : Re-write – indicates the performance of writing to an existing file.
- ID 4 : Random Read – indicates the performance of reading a file by reading random information from the file – i. e. this is not a sequential.
- ID 5 : Read.

- ID 6 : Random Write – indicates the performance of writing to a file in various random locations. i.e this is not a sequential write.
- ID 7 : Backward Read.
- ID 8 : Record Re-Write.
- ID 9 : Stride Read.
- ID 10 : Fread.
- ID 11 : Fwrite.
- ID 12 : Freread.
- ID 13 : Frewrite

The complete description of each test can be found in the IOzone documentation. Note that in case of databases benchmarking the tests, 0 to 6 are particularly interesting.

## IOzone Installation

IOzone can be installed as follows:

```

1
2
3
4
5
6
7
8
9
10
[root@vmoratest1 tmp]# wget http://www.iozone.org/src/current/iozone-3-398.i386.rpm
--2012-03-15 10:15:42-- http://www.iozone.org/src/current/iozone-3-398.i386.rpm
Resolving www.iozone.org... 208.45.140.198
Connecting to www.iozone.org[208.45.140.198]:80...
connected. HTTP request sent, awaiting response...
200 OK
Length: 755116 (737K) [application/x-rpm]
Saving to: `iozone-3-398.i386.rpm'

100%[=====
=====>]
755,116 278K/s in 2.7s

2012-03-15 10:15:45 (278 KB/s) -
`iozone-3-398.i386.rpm' saved [755116/755116]

```

11

```
[root@vmoratest1 tmp]# ls
io*iozone-3-398.i386.rpm

[root@vmoratest1 tmp]# rpm -ivh
iozone-3-398.i386.rpm

Preparing... #####
##### [100%]

1:iozone #####
##### [100%]

[root@vmoratest1 tmp]# whereis iozoneiozone

/opt/iozone/bin/iozone
```

## Basic usage

Once installed, IOzone can be started in automatic mode simply using the command: `iozone -a`

However, several options are available through `iozone -h` or in the documentation. We are going to present the most useful ones.

- Specify the file system to test:

```
iozone /xxx/yyy --> iozone /u01
```

- Specify the size of the file used for the tests:

```
iozone -s #k|#m|#g --> iozone -s 10g /u01
```

- Specify the size of the record used by IOzone

```
iozone -r #k|#m|#g --> iozone -s 10g -r 8k /u01
```

The record is the slices in which the file will be divided during the tests. In case of an Oracle database benchmarking, the record size should be set to the `db_block_size` (or the tablespace block size, if it applies).

- Specify the tests to run

```
iozone -i test_id --> iozone -i 0 -i 1 -s 10g -r 8k /u01
```

- Specify the min and max number of thread to use: `-l` (lower) `-u` (upper)

```
iozone -l # -u # --> iozone -i 0 -i 1 -l 1 -u 10 -s 10g -r 8k /u01
```

## IOzone output

By default, IOzone prints its output in a text format on the standard output:

In addition, its results can be sent to an Excel or LibreOffice Calc File using:

```
iozone -R -b file.xls iozone -R -b file.ods
```

Once the results are generated, some graphs can be generated using Calc or Excel.

Below is an example meant to measure the throughput (KB/s) for each type of test, with different level of parallelization going from 2 to 10:

```
[root@vmoratest1 current]# ./iozone -R -b  
/u01/oradata/bench_u01_test.xls -r 8k -s 100m -l 2 -u 10
```

And here is a graph generated by these results:

Posted - Wed, Jul 25, 2018 2:09 PM. This article has been viewed 9384 times.

Online URL: <http://kb.ictbanking.net/article.php?id=334>