

FIO (Flexible I/O) – a benchmark tool for any operating system

Article Number: 331 | Rating: Unrated | Last Updated: Wed, Jul 25, 2018 2:03 PM

I have just attended an interesting session held by Martin Nash (@mpnsh) at UKOUG 14 – Liverpool: “The least an Oracle DBA Should Know about Linux Administration” . During this session I had the opportunity to discover some interesting commands and tools such as FIO (Flexible I/O). FIO is a workload generator that can be used both for benchmark and stress/hardware verification.

FIO has support for 19 different types of I/O engines (sync, mmap, libaio, posixaio, SG v3, splice, null, network, syslet, guasi, solarisaio, and more), I/O priorities (for newer Linux kernels), rate I/O, forked or threaded jobs, and much more. It can work on block devices as well as files. fio accepts job descriptions in a simple-to-understand text format.

This tool has the huge advantage to be available for almost all kind of Operating Systems (POSIX, Linux, BSD, Solaris, HP-UX, AIX ,OS X, Android, Windows). If you want to use this tool in the context of Oracle database I invite you to have a look on the following blog from Yann Neuhaus: [Simulating database-like I/O activity with Flexible I/O](#)

In order to install it on ubuntu simply use the following command:

```
steulet@ThinkPad-T540p:~$ sudo apt-get install fio
```

After having installed fio you can run your first test. This first test will run 2 gigabyte of IO (read write) in directory /u01/fio.

```
steulet@ThinkPad-T540p:~$ mkdir /u01/fio
```

Once the directory have been created we can set up the configuration script as described below. However it is perfectly possible to execute this command in command line without configuration script (fio -name=global -ioengine=posixaio -rw=readwrite -size=2g -directory=/u01/fio -threads=1 -name=myReadWriteTest-Thread1):

```
1
2
3
4
5
6
7
      [global]
      ioengine=posixaio
      rw=readwrite
      size=2g
      directory=/u01/fio
      threads=1
      [myReadWriteTest-Thread1]
```

Now you can simply run your test with the command below:

```
steulet@ThinkPad-T540p:~$ fio testfio.fio
```

The output will looks like the following:

```
1
2
3
      myReadWriteTest-Tread1: (g=0): rw=rw,
      bs=4K-4K/4K-4K/4K-4K, ioengine=posixaio,
      iodepth=1
      fio-2.1.3
```

```
4 Starting 1 thread
5 Jobs: 1 (f=1): [M] [100.0% done]
6 [112.9MB/113.1MB/0KB /s] [28.9K/29.2K/0 iops]
7 [eta 00m:00s]
8
9 myReadWriteTest-Tread1: (groupid=0, jobs=1):
10 err= 0: pid=7823: Mon Dec 8 12:45:27 2014
11
12 read : io=1024.7MB, bw=98326KB/s, iops=24581,
13 runt= 10671msec
14
15 slat (usec): min=0, max=72, avg= 1.90, stdev=
16 0.53
17
18 clat (usec): min=0, max=2314, avg=20.25,
19 stdev=107.40
20
21 lat (usec): min=5, max=2316, avg=22.16,
22 stdev=107.41
23
24 clat percentiles (usec):
25
26 | 1.00th=[ 4], 5.00th=[ 6], 10.00th=[ 7],
27 20.00th=[ 7],
28
29 | 30.00th=[ 7], 40.00th=[ 7], 50.00th=[ 7],
30 60.00th=[ 7],
31
32 | 70.00th=[ 8], 80.00th=[ 8], 90.00th=[ 8],
33 95.00th=[ 10],
34
35 | 99.00th=[ 668], 99.50th=[ 1096], 99.90th=[
36 1208], 99.95th=[ 1208],
37
38 | 99.99th=[ 1256]
39
40 bw (KB /s): min= 2, max=124056,
```

```
23 per=100.00%, avg=108792.37, stdev=26496.59
24 write: io=1023.4MB, bw=98202KB/s, iops=24550,
runt= 10671msec
25
26 slat (usec): min=1, max=24, avg= 2.08, stdev=
0.51
27
28 clat (usec): min=0, max=945, avg= 9.71,
stdev=24.52
29
30 lat (usec): min=5, max=947, avg=11.79,
stdev=24.54
31
32 clat percentiles (usec):
33 | 1.00th=[ 5], 5.00th=[ 8], 10.00th=[ 8],
20.00th=[ 8],
34 | 30.00th=[ 8], 40.00th=[ 8], 50.00th=[ 9],
60.00th=[ 9],
35 | 70.00th=[ 9], 80.00th=[ 9], 90.00th=[ 10],
95.00th=[ 11],
36 | 99.00th=[ 15], 99.50th=[ 20], 99.90th=[
612], 99.95th=[ 628],
37 | 99.99th=[ 652]
38
39 bw (KB /s): min=108392, max=123536,
per=100.00%, avg=114596.33, stdev=3108.03
40
41 lat (usec) : 2=0.01%, 4=0.01%, 10=91.43%,
20=6.93%, 50=0.71%
42
43 lat (usec) : 100=0.13%, 250=0.01%, 500=0.01%,
750=0.47%, 1000=0.01%
```

```
lat (msec) : 2=0.31%, 4=0.01%

cpu       : usr=10.46%, sys=21.17%, ctx=527343,
majf=0, minf=12

IO depths  : 1=100.0%, 2=0.0%, 4=0.0%,
8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%

submit    : 0=0.0%, 4=100.0%, 8=0.0%,
16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%

complete  : 0=0.0%, 4=100.0%, 8=0.0%,
16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%

issued   : total=r=262309/w=261979/d=0,
short=r=0/w=0/d=0Run status group 0 (all jobs):

READ: io=1024.7MB, aggrb=98325KB/s,
minb=98325KB/s, maxb=98325KB/s,
mint=10671msec, maxt=10671msec

WRITE: io=1023.4MB, aggrb=98202KB/s,
minb=98202KB/s, maxb=98202KB/s,
mint=10671msec, maxt=10671msecDisk stats
(read/write):

sda: ios=6581/67944, merge=0/67,
ticks=4908/196508, in_queue=201408, util=56.49%
```

You will find some really good examples and a detailed list of parameters on the following website:
<http://www.bluestop.org/fio/HOWTO.txt>

This tool is really powerful and present the huge advantage to be available for more or less any Operating System. Such advantage will allow you to make some consistent comparison accross different kind of

architecture.

Posted - Wed, Jul 25, 2018 2:03 PM. This article has been viewed 19544 times.

Online URL: <http://kb.ictbanking.net/article.php?id=331>