

Simulating and testing I/O performances with ORION

Article Number: 324 | Rating: Unrated | Last Updated: Wed, Jul 25, 2018 1:19 PM

Since Oracle 11.2, Oracle provides ORION in the RDBMS binaries (in `${ORACLE_HOME}/bin`). ORION – ORacle Input Output Numbers – is an I/O calibration tool allowing to simulating and testing I/O performances an Oracle database would be confronted with. ORION basically supports four kinds of database activities based on either small or large I/O. ORION can, as any respectable I/O simulation tool, generate an adapted workload using a given percentage of reads and write operations.

OLTP activity

ORION can simulate OLTP systems typically generating small random read and write activities. The I/O sizes are equivalent to the database block size (usually 8KB) and the application analyzes the following database I/O KPIs:

- IOPS (I/O Per seconds)
- I/O latency average (I/O turn-around time) per request

The first part of the “simple” run (option `-run simple`) simulates this kind of workload (only with read). ORION also provides an OLTP-like (option `-run oltp`) scheme in which you can specify the percentage of writes.

Large sequential read activity

ORION can also simulate large sequential I/Os which can be caused by database exports (through data pump), RMAN backup, or of course a data warehouse environment. In this case, large I/Os will be simulated (1 MB per default). These types of applications or procedures are processing large amount of data and the important KPI is the throughput in MBps (MB per seconds). In this case, ORION can simulate large I/O read or write streams.

This kind of workload requires the option “`-type seq`” to be set in order to simulate large sequential I/Os.

The “mixed” workload through a workload “matrix” might also be used, see below.

Large random read activity

Large random I/O is another kind of workload ORION can simulate. The tool therefore simulates large I/Os striped over several disk, allowing to simulate parallel activities. In this case, multiple large I/O streams are started against the available disks.

The “dss” run (option `-run dss`) simulates this type of activity.

Mixed Workload

Finally, the mixed workload allows to simulate together small random I/Os and large sequential or random I/Os. Of course, this workload type is useful for OLTP systems confronted with performance issues caused by online backups for instance. A percentage of writes can also be simulated during these tests.

The “normal” run (option `-run normal`) simulates mixed activity of the sort, but of course the “matrix” workload is perfectly adapted to this simulation.

ORION has one major drawback: It cannot perform a filesystem performance analysis. It was only designed to be started against devices. We tested it on raw devices, since they are not buffered as block devices.

ORION can be used to test performances of disks, luns, SAN, DAS, or even certain type of NAS (we have not tested this so far).

In order to prepare the test, first of all we have to purge the raw device(s) on which we will generate the load:

```
1  
# dd if=/dev/zero of=/dev/raw/raw1 bs=1k  
2 count=1000  
3 1000+0 records in
```

```
4                                1000+0 records out
                                1024000 bytes (1.0 MB) copied, 3.52424 seconds,
                                291 kB/s
```

A file containing the devices on which the load will be generated on must be created (testname.lun):

```
1                                # cat simpletest.lun
2                                /dev/raw/raw1
```

ORION offers a very interesting feature. Since most of the SAN have caches, there is a risk of being impacted by “cache hits” during the I/O load tests. For this purpose ORION will generate random I/O which cannot be re-used by the test in order to full-fill the SAN cache and avoid the “I/O cache hit” effect. ORION even offers the possibility to set the size of the SAN cache in order to be quite precise. If this parameter is not set, ORION fulfills per default during 2 minutes the cache with useless I/O.

In order to run a so called “simple test”, the “-run simple” option can be used. Take care that this test scheme will not mix up several types of workload. Small random I/O will be performed separated from large random and sequential I/Os. Moreover, only reads are simulated in this test scheme. Note that ORION refuses to start without a fully qualified path:

```
1                                # orion -run simple -testname simpletest
2
3                                ORION: ORacle IO Numbers -- Version 11.2.0.2.0
4
```

```
5 simpletest_20110201_1550
6 Calibration will take approximately 9 minutes.
7 Using a large value for -cache_size may take longer.
8 ORA-56727: orion must be invoked using its full,
9 absolute path
orion_main: orion_spawn sml failed
Test aborted due to errors.
```

Once started as expected, the following output is generated:

```
1 # $ORACLE_HOME/bin/orion -run simple
2 -testname simpletest
3
4 ORION: ORacle IO Numbers -- Version 11.2.0.2.0
5 simpletest_20110201_1551
6 Calibration will take approximately 9 minutes.
7 Using a large value for -cache_size may take longer.
8 Maximum Large MBPS=101.82 @ Small=0 and
9 Large=2
10 Maximum Small IOPS=635 @ Small=5 and
Large=0
```

```

11 Small Read Latency: avg=7867 us, min=1858 us,
12 max=210500 us, std dev=4914 us @ Small=5 and
13 Large=0
14
15 Minimum Small Latency=7867 usecs @ Small=5
16 and Large=0
17
18 Small Read Latency: avg=7867 us, min=1858 us,
19 max=210500 us, std dev=4914 us @ Small=5 and
20 Large=0
21
22 Small Read / Write Latency Histogram @ Small=5
23 and Large=0
24
25 Latency:          # of IOs (read)  # of IOs (write)
26
27 0 - 1      us:      0          0
28
29 2 - 4      us:      0          0
30
31 4 - 8      us:      0          0
32
33 8 - 16     us:      0          0
34
35 16 - 32    us:      0          0
36
37 32 - 64    us:      0          0
38
39 64 - 128   us:      0          0
40
41 128 - 256  us:      0          0
42
43 256 - 512  us:      0          0
44
45 512 - 1024 us:      0          0
46
47 1024 - 2048 us:     470         0

```

| | | | | |
|----|-----------------------|-----|-------|---|
| 30 | 2048 - 4096 | us: | 4863 | 0 |
| 31 | 4096 - 8192 | us: | 19006 | 0 |
| 32 | 8192 - 16384 | us: | 12410 | 0 |
| 33 | 16384 - 32768 | us: | 1207 | 0 |
| 34 | 32768 - 65536 | us: | 138 | 0 |
| 35 | 65536 - 131072 | us: | 23 | 0 |
| 36 | 131072 - 262144 | us: | 2 | 0 |
| 37 | 262144 - 524288 | us: | 0 | 0 |
| 38 | 524288 - 1048576 | us: | 0 | 0 |
| 39 | 1048576 - 2097152 | us: | 0 | 0 |
| 40 | 2097152 - 4194304 | us: | 0 | 0 |
| 41 | 4194304 - 8388608 | us: | 0 | 0 |
| | 8388608 - 16777216 | us: | 0 | 0 |
| | 16777216 - 33554432 | us: | 0 | 0 |
| | 33554432 - 67108864 | us: | 0 | 0 |
| | 67108864 - 134217728 | us: | 0 | 0 |
| | 134217728 - 268435456 | us: | 0 | 0 |

- Orion output 1 -

The output represented above shows:

- Maximum MBPS (MB per seconds): 101
- Maximum IOPS (I/O per seconds): 635
- I/O latencies information

The following files are produced:

```
1
2
3
4
5
6
7
8
9
10
11
12
```

| | | | | | |
|------------|---|------------|------|-------------|--------------------------------------|
| -rw-r--r-- | 1 | oracle dba | 174 | Feb 1 12:50 | simpletest_20110201_1550_trace.txt |
| -rw-r--r-- | 1 | oracle dba | 342 | Feb 1 12:50 | simpletest_20110201_1550_summary.txt |
| -rw-r--r-- | 1 | oracle dba | 0 | Feb 1 12:50 | simpletest_20110201_1550_mbps.csv |
| -rw-r--r-- | 1 | oracle dba | 0 | Feb 1 12:50 | simpletest_20110201_1550_lat.csv |
| -rw-r--r-- | 1 | oracle dba | 0 | Feb 1 12:50 | simpletest_20110201_1550_iops.csv |
| -rw-r--r-- | 1 | oracle dba | 982 | Feb 1 12:50 | simpletest_20110201_1550_hist.txt |
| -rw-r--r-- | 1 | oracle dba | 3227 | Feb 1 12:58 | simpletest_20110201_1551_trace.txt |
| -rw-r--r-- | 1 | oracle dba | 1773 | Feb 1 12:58 | simpletest_20110201_1551_summary.txt |

```
-rw-r--r-- 1 oracle dba 498 Feb 1 12:58
simpletest_20110201_1551_mbps.csv

-rw-r--r-- 1 oracle dba 549 Feb 1 12:58
simpletest_20110201_1551_lat.csv

-rw-r--r-- 1 oracle dba 526 Feb 1 12:58
simpletest_20110201_1551_iops.csv

-rw-r--r-- 1 oracle dba 7033 Feb 1 12:58
simpletest_20110201_1551_hist.txt
```

These files contain useful information about I/O performances and have the following structure:

- Lines represent large I/O load levels (number of outstanding large I/Os)
- Columns represent small I/O load levels (number of outstanding small I/Os)

Since there is no small/large I/O mix up in the “simple” test, the tables stored in the generated files will have only one dimension (either one column or one row).

The file containing the throughput performances measured in MBPS (MB per second) has the following layout:

```
# cat simpletest_20110201_1551_mbps.csv
```

This comma-separated-value file contains the rates sustained by large I/Os in MBps. Each value corresponds to a data point test that used a fixed number of outstanding small and large I/Os. The number of outstanding small I/Os for a value is specified by its column header in the first row. The number of outstanding large I/Os for a value is specified by its row header in the first column.

```
Large/Small,      0,      1,      2,      3,      4,      5

1,  56.41
2, 101.82
```


In our example, the highest throughput is 101 MB/s. This value is rather weak for a SAN. The faster SANs reach up to 200 or 300 MB/s.

The I/O latency can be found in the following .csv file (simpletest_20110201_1551_lat.csv):

```
# cat simpletest_20110201_1551_lat.csv
```

This comma-separated-value file contains the average latency sustained by small I/Os in microseconds. Each value corresponds to a data point test that used a fixed number of outstanding small and large I/Os. The number of outstanding small I/Os for a value is specified by its column header in the first row. The number of outstanding large I/Os for a value is specified by its row header in the first column.

```
Large/Small,      1,      2,      3,      4,      5
0, 9478.83, 8590.16, 8097.65, 8089.16, 7866.98
1
2
```

The measured latency is located between 7.8 to 9.4 ms. According to several forums, the accepted I/O latency should be below 10ms. The average latency is indeed below 10ms, however 12410 (about one third) small I/Os had a latency between 8 and 16ms (see above – Orion output 1 -).

To simulate mixed up I/O workload (small random, large random, and large sequential), you have to use the following command and perform a “normal” test:

```
# $ORACLE_HOME/bin/orion -run normal -testname normaltest
```

This command might fail after a while with the following output:

```
1
2
3
4
5
# $ORACLE_HOME/bin/orion -run normal
-testname normaltest
ORION: ORacle IO Numbers -- Version 11.2.0.2.0
normaltest_20110201_1654
Calibration will take approximately 19 minutes.
```

```
6          Using a large value for -cache_size may take longer.
7          Error completing
8          IO(storax_aiowait)
9          ORA-27061: waiting for async I/Os failed
10         Linux-x86_64 Error: 14:
11         Bad addressAdditional information: -1
12         Additional information: 1048576

          Test aborted due to errors.
```

This issue is documented in the bug 9104898: “ORION FAILS WITH ORA-27061: WAITING FOR ASYNC I/OS FAILED”. According to the bug description, ORION has difficulties with Large I/Os on Red Hat Enterprise Linux Server 5. The bug is currently under investigation at Oracle development.

It is also possible to define exactly how many outstanding small and large I/O you want ORION to simulate in parallel. For this purpose the “matrix” test should be used. In this case you specify the maximum number of outstanding small (-num_small) and large (-num_large) I/Os. You can even specify the size for the large I/O (depending on your infrastructure and future database settings):

```
1          # $ORACLE_HOME/bin/orion -run advanced
2          -testname advancedtest -matrix max -num_small 4
3          -num_large 4 -size_large 512
4
5          ORION: ORacle IO Numbers -- Version 11.2.0.2.0
```

```
6 advancedtest_20110201_1740

Calibration will take approximately 26 minutes.

Using a large value for -cache_size may take longer.
```

To simulate OLTP activities with about 20% of writes, the following parameters might be used:

```
1 # $ORACLE_HOME/bin/orion -run oltp -testname
2 oltp_write -write 20
3
4 ORION: ORacle IO Numbers -- Version 11.2.0.2.0
5 oltp_write_20110202_1140
6 Calibration will take approximately 22 minutes.
7 Using a large value for -cache_size may take longer.
8 Maximum Small IOPS=2820 @ Small=19 and
9 Large=0
10 Small Read Latency: avg=8333 us, min=118 us,
11 max=540359 us, std dev=9541 us @ Small=19 and
12 Large=0
13 Small Write Latency: avg=315 us, min=217 us,
max=11142 us, std dev=331 us @ Small=19 and
Large=0
Minimum Small Latency=5409 usecs @ Small=4
```

14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

and Large=0

Small Read Latency: avg=6685 us, min=243 us, max=202301 us, std dev=5715 us @ Small=4 and Large=0

Small Write Latency: avg=300 us, min=217 us, max=6300 us, std dev=296 us @ Small=4 and Large=0

Small Read / Write Latency Histogram @ Small=19 and Large=0

| Latency: | | # of IOs (read) | # of IOs (write) |
|-----------------|------|-----------------|------------------|
| 0 - 1 us: | 0 | 0 | |
| 2 - 4 us: | 0 | 0 | |
| 4 - 8 us: | 0 | 0 | |
| 8 - 16 us: | 0 | 0 | |
| 16 - 32 us: | 0 | 0 | |
| 32 - 64 us: | 0 | 0 | |
| 64 - 128 us: | 0 | 0 | |
| 128 - 256 us: | 1 | 6457 | |
| 256 - 512 us: | 339 | 1894 | |
| 512 - 1024 us: | 115 | 369 | |
| 1024 - 2048 us: | 2830 | 92 | |

| | | | | |
|----|-----------------------|-----|-------|----|
| 33 | 2048 - 4096 | us: | 7105 | 44 |
| 34 | 4096 - 8192 | us: | 16469 | 11 |
| 35 | 8192 - 16384 | us: | 7301 | 0 |
| 36 | 16384 - 32768 | us: | 1106 | 0 |
| 37 | 32768 - 65536 | us: | 190 | 0 |
| 38 | 65536 - 131072 | us: | 30 | 0 |
| 39 | 131072 - 262144 | us: | 4 | 0 |
| 40 | 262144 - 524288 | us: | 0 | 0 |
| 41 | 524288 - 1048576 | us: | 0 | 0 |
| 42 | 1048576 - 2097152 | us: | 0 | 0 |
| | 2097152 - 4194304 | us: | 0 | 0 |
| | 4194304 - 8388608 | us: | 0 | 0 |
| | 8388608 - 16777216 | us: | 0 | 0 |
| | 16777216 - 33554432 | us: | 0 | |
| | 0 | | | |
| | 33554432 - 67108864 | us: | 0 | |
| | 0 | | | |
| | 67108864 - 134217728 | us: | 0 | 0 |
| | 134217728 - 268435456 | us: | | |
| | 0 | | 0 | |

You will find the technical specifications concerning the hardware and operating system below:

```
1                                     # uname -a
2
3                                     Linux serveroracle01 2.6.18-194.26.1.el5 #1 SMP
4                                     x86_64 GNU/Linux
5
6                                     # cat /etc/redhat-release
7
8                                     Red Hat Enterprise Linux Server release 5.5
9                                     (Tikanga)
10
11                                    EVA: HSV210 (6220). Raid level of the virtual disk
12                                    is Raid 5
```

As a Summary

Considering all advantages and drawbacks, ORION is quite an interesting tool. However, as mentioned by several experts, this tool is mainly useful in case you are at the beginning of a project and have full access to the disk/LUN configuration. It is definitively not a good idea to overwrite some LUNs on the production environment. ORION also allows realizing test schemes that are perfectly fitted to the database behavior and offer a robust framework to compare storage vendors and storage infrastructures.

Posted - Wed, Jul 25, 2018 1:19 PM. This article has been viewed 4184 times.

Online URL: <http://kb.ictbanking.net/article.php?id=324>