

FIO (Flexible IO Tester)

Синтетическое тестирование (v.1.0)

Начальные условия

Мероприятия по подготовке

1. Создать виртуальную машину следующей конфигурации:
 1. 8 vCPU
 2. 2 GB mem
 3. Linux Debian
 4. 256 GB Disk
2. Установить fio;
3. Установить nmon;
4. Создать файлы файлы по шаблону ниже. Или скопировать из предоставленного *.iso образа
файлы внутрь виртуальной машины;
 1. Командой lsblk найти тестируемое устройство;
 2. Указать в файле bench.ini тестируемое устройство, в примере - "/dev/sdb";

bench.sh

```
#!/bin/sh
BASEDIR=/home/admin #<---- Укажите рабочую директорию, где будут храниться
отчёты
INIDIR=$BASEDIR/ini
LOGDIR=$BASEDIR/log
DEV=/dev/sdb #<---- Укажите девайс который будем тестировать
IODEPTH=16
RUNTIME=300
TPAUSE=30
STOR="sdb" #<---- Укажите девайс который будем тестировать
echo "stor $STOR" > $INIDIR/nmon.gr

# -----
/usr/bin/nmon -f -t -s 5 -c 5000 -g $INIDIR/nmon.gr -m $LOGDIR/fio
fiocmdlc="--filename=$DEV --iodepth=$IODEPTH --runtime=$RUNTIME "
for fiocmdl in " --blocksize=4k --output=$LOGDIR/rand_read_4.out
$INIDIR/rand_read.ini"\
" --blocksize=8k --output=$LOGDIR/rand_read_8.out
$INIDIR/rand_read.ini"\
" --blocksize=16k --output=$LOGDIR/rand_read_16.out
$INIDIR/rand_read.ini"\
" --blocksize=256k --output=$LOGDIR/rand_read_256.out
$INIDIR/rand_read.ini"\
" --blocksize=4k --output=$LOGDIR/rand_write_4.out
```

```

$INIDIR/rand_write.ini"\
" --blocksize=8k --output=$LOGDIR/rand_write_8.out
$INIDIR/rand_write.ini"\
" --blocksize=16k --output=$LOGDIR/rand_write_16.out
$INIDIR/rand_write.ini"\
" --blocksize=256k --output=$LOGDIR/rand_write_256.out
$INIDIR/rand_write.ini"\
" --blocksize=4k --output=$LOGDIR/rand_read_write_4.out
$INIDIR/rand_read_write.ini"\
" --blocksize=8k --output=$LOGDIR/rand_read_write_8.out
$INIDIR/rand_read_write.ini"\
" --blocksize=16k --output=$LOGDIR/rand_read_write_16.out
$INIDIR/rand_read_write.ini"\
" --blocksize=256k --output=$LOGDIR/rand_read_write_256.out
$INIDIR/rand_read_write.ini"\
" --blocksize=4k --output=$LOGDIR/read_4.out $INIDIR/read.ini"\
" --blocksize=8k --output=$LOGDIR/read_8.out $INIDIR/read.ini"\
" --blocksize=16k --output=$LOGDIR/read_16.out $INIDIR/read.ini"\
" --blocksize=256k --output=$LOGDIR/read_256.out $INIDIR/read.ini"\
" --blocksize=4k --output=$LOGDIR/read_write_4.out
$INIDIR/read_write.ini"\
" --blocksize=8k --output=$LOGDIR/read_write_8.out
$INIDIR/read_write.ini"\
" --blocksize=16k --output=$LOGDIR/read_write_16.out
$INIDIR/read_write.ini"\
" --blocksize=256k --output=$LOGDIR/read_write_256.out
$INIDIR/read_write.ini"\
" --blocksize=4k --output=$LOGDIR/write_4.out $INIDIR/write.ini"\
" --blocksize=8k --output=$LOGDIR/write_8.out $INIDIR/write.ini"\
" --blocksize=16k --output=$LOGDIR/write_16.out $INIDIR/write.ini"\
" --blocksize=256k --output=$LOGDIR/write_256.out $INIDIR/write.ini";
do
echo " ----- $fiocmdlc $fiocmdl" >> $LOGDIR/fio/test.log
fio $fiocmdlc $fiocmdl >> $LOGDIR/fio/test.log
echo " ----- endtest" >> $LOGDIR/fio/test.log
sleep $TPAUSE
done
pkill nmon
# -----

```

[/ini/write.ini](#)

```

[writetest]
rw=write
direct=1
buffered=0
ioengine=libaio

```

[/ini/read.ini](#)

```
[readtest]
rw=read
direct=1
buffered=0
ioengine=libaio
```

/ini/read_write.ini

```
[writetest]
rw=rw
direct=1
buffered=0
ioengine=libaio
```

/ini/rand_write.ini

```
[writetest]
rw=randwrite
direct=1
buffered=0
ioengine=libaio
```

/ini/rand_read.ini

```
[readtest]
rw=randread
direct=1
buffered=0
ioengine=libaio
```

/ini/rand_read_write.ini

```
[writetest]
rw=randrw
direct=1
buffered=0
ioengine=libaio
```

Мероприятия по проверке

1. Запустить тестирование командой `bench.sh`

Мероприятия после проверки

1. Скачать с сервера полученные результаты теста в файле вида "fio-bench_XXXXXX_XXXX.nmon"
2. На ПК администратора установить [nmon analyzer](#)
3. Открыть полученный результат с помощью nmon analyzer
4. Перейти на вкладку DISK_SUMM



Сводные результаты тестов представлены на данной странице: [Результаты тестов fio](#)

Синтетическое тестирование (v.1.1)

Начальные условия

Мероприятия по подготовке

1. Создать виртуальную машину следующей конфигурации:
 1. 8 vCPU
 2. 2 GB mem
 3. Linux Debian
 4. 256 GB Disk
2. Установить fio;
3. Установить nmon;
4. Создать файлы файлы по шаблону ниже. Или скопировать из предоставленного *.iso образа
файлы внутрь виртуальной машины;
 1. Командой lsblk найти тестируемое устройство;
 2. Указать в файле bench.ini тестируемое устройство, в примере - "/dev/sdb";

[bench.sh](#)

```
#!/bin/sh
BASEDIR=/home/admin #<---- Укажите рабочую директорию, где будут храниться
отчёты
INIDIR=$BASEDIR/ini
LOGDIR=$BASEDIR/log
DEV=/dev/sdb #<---- Укажите девайс который будем тестировать
IODEPTH=16
RUNTIME=300
TPAUSE=30
STOR="sdb" #<---- Укажите девайс который будем тестировать
echo "stor $STOR" > $INIDIR/nmon.gr

# -----
```

```
/usr/bin/nmon -f -t -s 5 -c 5000 -g $INIDIR/nmon.gr -m $LOGDIR/fio
fiocfg="--filename=/dev/sdb --iodepth=$IODEPTH --runtime=$RUNTIME "
for fiorun in \
" --blocksize=4k --output=$LOGDIR/rand_write_4.out
$INIDIR/rand_write.ini" \
" --blocksize=8k --output=$LOGDIR/rand_write_8.out
$INIDIR/rand_write.ini" \
" --blocksize=16k --output=$LOGDIR/rand_write_16.out
$INIDIR/rand_write.ini" \
" --blocksize=256k --output=$LOGDIR/rand_write_256.out
$INIDIR/rand_write.ini" \
" --blocksize=4k --output=$LOGDIR/rand_read_write_4.out
$INIDIR/rand_read_write.ini" \
" --blocksize=8k --output=$LOGDIR/rand_read_write_8.out
$INIDIR/rand_read_write.ini" \
" --blocksize=16k --output=$LOGDIR/rand_read_write_16.out
$INIDIR/rand_read_write.ini" \
" --blocksize=256k --output=$LOGDIR/rand_read_write_256.out
$INIDIR/rand_read_write.ini" \
" --blocksize=4k --output=$LOGDIR/rand_read_4.out
$INIDIR/rand_read.ini" \
" --blocksize=8k --output=$LOGDIR/rand_read_8.out
$INIDIR/rand_read.ini" \
" --blocksize=16k --output=$LOGDIR/rand_read_16.out
$INIDIR/rand_read.ini" \
" --blocksize=256k --output=$LOGDIR/rand_read_256.out
$INIDIR/rand_read.ini" \
" --blocksize=4k --output=$LOGDIR/write_4.out $INIDIR/write.ini" \
" --blocksize=8k --output=$LOGDIR/write_8.out $INIDIR/write.ini" \
" --blocksize=16k --output=$LOGDIR/write_16.out $INIDIR/write.ini" \
" --blocksize=256k --output=$LOGDIR/write_256.out $INIDIR/write.ini" \
" --blocksize=4k --output=$LOGDIR/read_write_4.out
$INIDIR/read_write.ini" \
" --blocksize=8k --output=$LOGDIR/read_write_8.out
$INIDIR/read_write.ini" \
" --blocksize=16k --output=$LOGDIR/read_write_16.out
$INIDIR/read_write.ini" \
" --blocksize=256k --output=$LOGDIR/read_write_256.out
$INIDIR/read_write.ini" \
" --blocksize=4k --output=$LOGDIR/read_4.out $INIDIR/read.ini" \
" --blocksize=8k --output=$LOGDIR/read_8.out $INIDIR/read.ini" \
" --blocksize=16k --output=$LOGDIR/read_16.out $INIDIR/read.ini" \
" --blocksize=256k --output=$LOGDIR/read_256.out $INIDIR/read.ini" ;
do
    echo " ----- $fiocfg $fiorun" >> $LOGDIR/fio/test.log
    fio $fiocfg $fiorun >> $LOGDIR/fio/test.log
    echo " ----- endtest" >> $LOGDIR/fio/test.log
    sleep $TPAUSE
done
pkill nmon
```

```
# -----
```

/ini/write.ini

```
[writetest]
rw=write
direct=1
buffered=0
ioengine=libaio
```

/ini/read.ini

```
[readtest]
rw=read
direct=1
buffered=0
ioengine=libaio
```

/ini/read_write.ini

```
[writetest]
rw=rw
direct=1
buffered=0
ioengine=libaio
```

/ini/rand_write.ini

```
[writetest]
rw=randwrite
direct=1
buffered=0
ioengine=libaio
```

/ini/rand_read.ini

```
[readtest]
rw=randread
direct=1
buffered=0
ioengine=libaio
```

/ini/rand_read_write.ini

```
[writetest]
```

```
rw=randrw
direct=1
buffered=0
ioengine=libaio
```

Мероприятия по проверке

1. Запустить тестирование командой `bench.sh`

Мероприятия после проверки

1. Скачать с сервера полученные результаты теста в файле вида “`fiobench_XXXXXX_XXXX.nmon`”
2. На ПК администратора установить [nmon analyzer](#)
3. Открыть полученный результат с помощью `nmon analyzer`
4. Перейти на вкладку `DISK_SUMM`

Нагрузочное тестирование СХД

Начальные условия

Мероприятия по подготовке

1. Создать каталог куда будут записываться результаты теста. В примере указан каталог `/var/log/storage1`
2. Создать конфигурационный файл теста
 1. Указать примонтированные блочные устройства (`vdb, vdc,...`)
 2. Указать время замера в переменной **`runtime=`**

[/var/fio.rtf](#)

```
[global]
ioengine=libaio
direct=1
group_reporting
time_based
runtime=1h
buffered=0
random_generator=tausworthe64
norandommap
disable_clat=1
disable_slart=1
log_avg_msec=5000

[ NVMe_1 ]
```

```
description=Random read/write 80/20
rw=randrw
rwmixread=80
bs=8k
filename=/dev/vdb
iodepth=16
numjobs=8
```

```
[ NVMe_2 ]
description=Random read/write 80/20
rw=randrw
rwmixread=80
bs=8k
filename=/dev/vdc
iodepth=16
numjobs=8
```

```
[ HDD_1 ]
description=Read/write 80/20
rw=rw
rwmixread=80
bs=256k
filename=/dev/vdd
iodepth=1
numjobs=1
```

```
[ HDD_2 ]
description=Read/write 80/20
rw=rw
rwmixread=80
bs=256k
filename=/dev/vde
iodepth=1
numjobs=1
```

```
[ HDD_3 ]
description=Read/write 80/20
rw=rw
rwmixread=80
bs=256k
filename=/dev/vdf
iodepth=1
numjobs=1
```

Мероприятия по проверке

1. Запустить тестирование командой

```
fio /var/fio.rtf -output-format=json --
```

```
output=/var/log/storage1/fio.json --  
write_lat_log=/var/log/storage1/lat.json --  
write_bw_log=/var/log/storage1/bw.json  
-write_iops_log=/var/log/storage1/iops.json
```

Мероприятия после проверки

Смотри также

[Примеры от Oracle](#)

From:

<https://micronode.ru/> - **micronode.ru**

Permanent link:

<https://micronode.ru/wiki/benchmark/tests/storage/fio>

Last update: **2025/06/02 07:44**

