fio Disk I/O benchmark

CONCEPTS

Writing to a raw device (/dev/sdX, /dev/nvme0n1) destroys its data. Benchmark a file on the filesystem, or a scratch device you can wipe.

Job file / CLI workload
↓ parsed
Jobs numjobs clones
↓ submit
ioengine iodepth in flight
↓ complete
Report IOPS, BW, latency
One fio run

RUNNING

fio *<jobfile>*
Run a job file

fio --name=*<job>* **--rw=***<mode>* **...**
Define a job on the CLI

fio --section=*<name>* *<jobfile>*
Run only one section

fio --output=*<file>* **--output-format=***<fmt>*
Write report; normal/terse/json/json+

fio --parse-only *<jobfile>*
Check options, no I/O

fio --readonly ...
Refuse write jobs

fio --cmdhelp=*<option>*
Help for one option

fio --version
Show version

ACCESS PATTERN

--rw=read / write
Sequential read / write

--rw=randread / randwrite
Random read / write

--rw=randrw --rwmixread=*<pct>*
Mixed random I/O

--bs=*<size>*
Block size, e.g. 4k or 1m

SIZE AND FILES

--filename=*<path>*
File or device to test

SIZE AND FILES (cont.)

--directory=*<dir>*
Place generated files here

--size=*<size>*
I/O region per job

--nrfiles=*<n>*
Files per job

ENGINE AND QUEUE

--ioengine=libaio
Linux async I/O

--ioengine=io_uring
Modern Linux async I/O

--iodepth=*<n>*
I/Os kept in flight

--direct=1
O_DIRECT, skip page cache

TIME AND LOAD

--runtime=*<time>*
Stop after this long

--time_based
Loop until runtime ends

--ramp_time=*<time>*
Warm-up excluded from stats

--numjobs=*<n>*
Parallel clones of the job

--group_reporting
One summary for all clones

JOB FILES

Job file layout

```
[global]
ioengine=libaio
direct=1
runtime=60
time_based

[randread-4k]
rw=randread
bs=4k
iodepth=32
```

[global]
Defaults for all jobs

OUTPUT AND LOGS

--output-format=json
JSON report