

# Тестирование производительности дисков с помощью fio

 [redos.red-soft.ru/base/redos-7\\_3/7\\_3-administration/7\\_3-system-perf/7\\_3-fio](https://redos.red-soft.ru/base/redos-7_3/7_3-administration/7_3-system-perf/7_3-fio)

## Окружение

- **Версия РЕД ОС:** 7.3
- **Конфигурация:** Рабочая станция, Сервер графический, Сервер минимальный
- **Версия ПО:** fio-3.21-4

**Fio (Flexible I/O Tester)** — это инструмент, который позволяет запускать несколько потоков или процессов для выполнения операций ввода-вывода, заданных пользователем. Обычно **fio** используется для создания файла задания, описывающего нагрузку на ввод-вывод, которую необходимо смоделировать. Для получения точного результата тестирования производительности системы хранения данных важно учитывать несколько ключевых аспектов.

Выбор типов тестов и параметров **fio** является важным, поскольку такие настройки, как размер блока данных (**bs**) и количество задач (**numjobs**), могут существенно влиять на результаты. Изменение этих параметров может привести к получению различных данных о производительности. Поэтому следует стремиться к единообразию параметров на всех тестируемых стендах.

Важно!

Не запускайте тесты **fio** с рабочей нагрузкой на запись (**readwrite**, **randrw**, **write**, **trimwrite**) непосредственно на используемом устройстве. Для проведения таких тестов рекомендуется использовать тестовые среды или резервные устройства, чтобы избежать потери данных и негативных последствий для основного рабочего устройства.

## Установка

Для установки утилиты выполните в терминале команду (потребуется права администратора):

```
sudo dnf install fio
```

## Синтаксис

---

Утилита доступна для запуска через терминал при помощи команды следующего вида:

```
fio [<Опции>] [<Файл задания>]
```

## Опции

---

Основные опции **fio**:

- **--debug=<параметры>** — включает ведение журнала отладки;
- **--parse-only** — анализирует файл задания и выводит настройки, без операций ввода;
- **--merge-blktrace-only** — объединяет вывод **blktrace** (трассировка блоков) из предыдущих тестов;
- **--output=имя\_файла** — указывает файл для записи результатов;
- **--output-format=формат** — устанавливает формат вывода;
- **--bandwidth-log** — включает ведение журнала пропускной способности;
- **--minimal** — выводит минимально необходимую информацию;
- **--append-terse** — дополняет короткий вывод предыдущих результатов;
- **--terse-version=версия** — указывает версию для короткого вывода;
- **--version** — выводит информацию о версии программы и завершает работу;
- **--help** — показывает справку по командам;
- **--cpuclock-test** — тестирует тактовую частоту процессора;
- **--crctest=[тест]** — запускает тест контрольной суммы;
- **--cmdhelp=команда** — выводит справку по указанной команде;
- **--enghelp=[ввод-вывод[,команда]]** — выводит справку по указанному механизму ввода-вывода и команде;
- **--showcmd** — показывает команду, которую нужно использовать для запуска текущей конфигурации;
- **--readonly** — работает в режиме только для чтения;
- **--eta=when** — показывает время до завершения (ETA) теста;

- **--eta-interval=время** — интервал обновления информации о ETA;
- **--eta-newline=время** — интервал добавления новой строки в вывод ETA;
- **--status-interval=время** — интервал обновления статуса;
- **--section=имя** — указывает конкретный раздел конфигурации для выполнения;
- **--alloc-size=kb** — устанавливает размер выделения памяти в КБ;
- **--warnings-fatal** — обрабатывает предупреждения как ошибки;
- **--max-jobs=номер** — максимальное количество параллельных задач;
- **--server=аргументы** — запускает **fio** в режиме сервера с указанными аргументами;
- **--daemonize=pid-файл** — запускает **fio** как фоновый процесс и записывает PID в файл;
- **--client=имя\_хоста** — подключается к серверу **fio** на указанном хосте;
- **--remote-config=файл** — использует удалённый конфигурационный файл;
- **--idle-prof=опция** — формирует отчёт о простое процессора, опция может быть одной из следующих:
  - **calibrate** — выполняет калибровку и завершает работу;
  - **system** — показывает общий уровень простоя системы и единичную работу;
  - **percpu** — как system, но также показывает время простоя каждого процессора;
- **--inflate-log=журнал\_регистрации** — увеличить размер лог-файла;
- **--trigger-file=файл** — выполняет команду запуска, когда файл существует;
- **--trigger-timeout=время** — время ожидания перед запуском триггера;
- **--trigger=команда** — устанавливает команду в качестве локального триггера;
- **--trigger-remote=команда** — устанавливает команду в качестве удалённого запуска;
- **--aux-path=путь** — вспомогательный путь для временных файлов вместо текущего рабочего каталога.

## Формат файла задания

---

Любые параметры, следующие за **опциями**, будут считаться **файлами заданий**, если они не соответствуют параметру файла задания. В списке может быть несколько файлов заданий, и каждый файл задания будет рассматриваться как отдельная группа.

**Fio** принимает один или несколько файлов заданий, описывающих, что он должен делать. **Формат файла задания** — классический ini-файл, в котором имена, заключённые в квадратные скобки [], определяют название задания. Можно использовать любое имя в формате ASCII, кроме **global**, которое имеет предопределённое значение. После названия задания указывается последовательность из параметров, которые определяют поведение задания.

В разделе **global** устанавливаются значения по умолчанию для заданий, описанных в этом файле. Задание может переопределять параметр `*global*` section, а файл задания может также содержать несколько **глобальных параметров**.

## Параметры файла задания

---

Для некоторых параметров используется параметр определённого типа, например, целое число или строка. Везде, где требуется числовое значение, можно использовать арифметическое выражение, при условии, что оно заключено в круглые скобки.

Поддерживаются следующие операции:

- сложение (+),
- вычитание (-),
- умножение (\*),
- деление (/),
- модуль (%),
- возведение в степень (^).

Для значений времени в выражениях по умолчанию единицами измерения являются микросекунды.

Основные параметры для задания:

- **--name** — имя теста (можно изменить на любое другое);
- **--filename** — путь к файлу, который будет использоваться для тестирования I/O;
- **--directory** — каталог, который будет использоваться для тестирования I/O;
- **--size** — размер файла для теста;

- **--direct** — использование прямых I/O, обходящих кеш операционной системы;
- **--rw** — режим теста (чтение, запись или оба);
- **--bs** — размер блока ввода-вывода (I/O) для теста;
- **--ioengine** — механизм ввода-вывода, который будет использоваться для теста (например, libaio для асинхронного ввода-вывода);
- **--iodepth** — глубина очереди запросов на ввод-вывод для теста (количество одновременных запросов к файловой системе);
- **--runtime** — время выполнения теста;
- **--numjobs** — количество параллельных задач для теста;
- **--time\_based** — время работы будет использоваться для определения длительности теста;
- **--group\_reporting** — режим групповой отчётности (результаты будут отображаться для каждой группы рабочих нитей, а не для каждого рабочего потока);
- **--eta-newline** — интервал отображения прогресса выполнения теста и перевода строки.

## Примеры работы

---

### Тесты производительности операций ввода-вывода в секунду

---

Для тестирования пропускной способности используется одновременное чтение и запись небольшими блоками с использованием множества процессов. Тесты измеряют количество операций **чтения/записи**, которые система может выполнить за секунду.

Тесты являются показателем производительности хранилища данных или системы ввода-вывода и позволяют оценить **скорость обработки операций** ввода и вывода данных, независимо от их объёма. Чем выше значение **IOPS**, тем быстрее система может обрабатывать запросы на чтение и запись данных.

Важно!

При выполнении тестирования на запись в указанном файле данные перезаписываются. Поэтому не рекомендуется указывать в качестве файла блочное устройство с нужными данными.

### Тест скорости записи

---

Перед выполнением теста убедитесь, что в указанной директории достаточно свободного места (более 5 ГБ) для завершения теста.

```
fio --name=fiotest --blocksize=4k --directory=/home/user --size=5g --  
ioengine=libaio --iodepth=256 --runtime=60 --rw=randwrite --direct=1
```

где:

- **--name=fiotest** — имя теста (можно изменить на любое другое);
- **--blocksize=4k** — размер блока данных для теста (4 КБ);
- **--directory=/home/user** — каталог, в котором будет проводиться тест;
- **--size=5g** — размер файла для теста (5 ГБ);
- **--ioengine=libaio** — использование libaio (асинхронный ввод-вывод);
- **--iodepth=256** — глубина очереди (256 запросов одновременно);
- **--runtime=60** — время выполнения теста (60 секунд);
- **--rw=randwrite** — режим теста (случайная запись);
- **--direct=1** — использование прямого доступа к устройству (чтобы избежать кеширования данных).

В результате запуска команды будет проведён тест производительности I/O на указанном каталоге с указанными параметрами.

Результат:

```

fiotest: (g=0): rw=randwrite, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-
4096B, ioengine=libaio, iodepth=256
fio-3.28
Starting 1 process
fiotest: Laying out IO file (1 file / 5120MiB)
Jobs: 1 (f=1): [w(1)][100.0%][w=33.4MiB/s][w=8540 IOPS][eta 00m:00s]
fiotest: (groupid=0, jobs=1): err= 0: pid=3906: Tue Jul 30 12:05:52 2024
  write: IOPS=1354, BW=5418KiB/s (5548kB/s)(317MiB/60001msec); 0 zone resets
    slat (usec): min=4, max=7229.8k, avg=733.37, stdev=53817.01
    clat (usec): min=23, max=11931k, avg=188197.99, stdev=1188675.61
    lat (usec): min=89, max=11931k, avg=188931.83, stdev=1190998.57
    clat percentiles (msec):
      | 1.00th=[ 10], 5.00th=[ 12], 10.00th=[ 12], 20.00th=[ 13],
      | 30.00th=[ 13], 40.00th=[ 14], 50.00th=[ 14], 60.00th=[ 15],
      | 70.00th=[ 20], 80.00th=[ 27], 90.00th=[ 75], 95.00th=[ 184],
      | 99.00th=[ 7886], 99.50th=[11476], 99.90th=[11879], 99.95th=[11879],
      | 99.99th=[11879]
    bw ( KiB/s): min= 8, max=68669, per=100.00%, avg=17068.00, stdev=18091.84,
samples=35
    iops      : min= 2, max=17167, avg=4266.94, stdev=4522.95, samples=35
    lat (usec) : 50=0.01%, 100=0.01%, 250=0.01%, 500=0.02%, 750=0.02%
    lat (usec) : 1000=0.02%
    lat (msec) : 2=0.07%, 4=0.12%, 10=1.00%, 20=69.48%, 50=15.80%
    lat (msec) : 100=5.15%, 250=4.88%, 500=0.76%, 750=0.08%, 1000=0.57%
    lat (msec) : 2000=0.16%, >=2000=1.87%
    cpu       : usr=0.67%, sys=4.04%, ctx=77891, majf=0, minf=15
    IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=99.9%
    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=0.1%
    issued rwts: total=0,81270,0,0 short=0,0,0,0 dropped=0,0,0,0
    latency   : target=0, window=0, percentile=100.00%, depth=256

Run status group 0 (all jobs):
  WRITE: bw=5418KiB/s (5548kB/s), 5418KiB/s-5418KiB/s (5548kB/s-5548kB/s),
io=317MiB (333MB), run=60001-60001msec

Disk stats (read/write):
  sda: ios=3/77906, merge=0/1835, ticks=63/698196, in_queue=716266, util=95.37%

```

где:

- **slat** — задержка отправки (submission latency). Это время, которое затрачивается на передачу запроса ввода-вывода в очередь устройства;
- **clat** — задержка выполнения (completion latency). Это время, которое прошло с момента передачи запроса устройству до его завершения и возврата результата;
- **lat** — общая задержка (total latency) — это сумма slat+clat;
- **bw** — пропускная способность: количество образцов и процент от общей пропускной способности;

- **iops** — количество операций ввода-вывода, выполненных за одну секунду. Показатель производительности хранилища. Чем выше IOPS, тем больше операций устройство способно обработать за единицу времени;
- **lat (nsec/usec/msec)** — распределение задержек завершения операций ввода-вывода;
- **cpu** — отражает статистику использования ресурсов процессора (CPU) во время выполнения теста ввода-вывода;
- **IO depths** — распределение глубин ввода-вывода за время существования задания;
- **IO submit** — количество операций ввода-вывода, отправленных в одном вызове;
- **IO complete** — количество завершённых операций ввода-вывода;
- **IO issued rwt** — количество отправленных запросов на чтение/запись/обрезку, короткие или отброшенные;
- **IO latency** — задержки ввода-вывода для достижения заданной цели.

После того как каждый клиент был перечислен, появляются **статистические данные для группы**:

```
Run status group 0 (all jobs):
  WRITE: bw=5418KiB/s (5548kB/s), 5418KiB/s-5418KiB/s (5548kB/s-5548kB/s),
  io=317MiB (333MB), run=60001-60001msec
```

где:

- **WRITE** — указывает, что результаты относятся к операциям записи;
- **bw** — совокупная пропускная способность потоков в заданной группе, за которой следуют минимальная и максимальная пропускная способность всех потоков в группе. 5418KiB/s-5418KiB/s (5548kB/s-5548kB/s) — диапазон скоростей записи, достигнутых в ходе теста, является фиксированным;
- **io** — совокупный объём выполненных операций ввода-вывода всех потоков в группе;
- **run** — самое короткое и самое длинное время выполнения потоков в группе.

В конце выводится **статистика по дискам**:

```
Disk stats (read/write):
  sda: ios=3/77906, merge=0/1835, ticks=63/698196, in_queue=716266, util=95.37%
```

где:



- **ios** — количество операций ввода-вывода, выполненных всеми группами (**3** — количество операций чтения, **77906** — количество операций записи);
- **merge** — количество объединений, выполненных планировщиком ввода-вывода (иногда несколько операций могут быть объединены в одну, чтобы снизить нагрузку на систему). **0** — количество объединённых операций чтения, **1835** — количество объединённых операций записи.
- **ticks** — количество тактов, в течение которых диск был занят, **63** — время, затраченное на операции чтения (в миллисекундах), **698196** — время, затраченное на операции записи (в миллисекундах);
- **in\_queue** — общее время, проведённое в очереди на выполнение (в миллисекундах);
- **util** — занятость устройства, отображаемая в процентах (**95.37%**). Значение, близкое к **100%**, может указывать на высокую загрузку диска и на необходимость оптимизации или добавления дополнительного оборудования.

## Тест скорости чтения

---

Для корректного выполнения команды тестирования скорости чтения с приведёнными ниже параметрами необходимо выполнить следующие условия:

- для выполнения теста в директории `/home/user` автоматически будут созданы файлы для каждого задания, количество которых определяется параметром **--numjobs**, размер файлов определяется параметром **--size**;
- пользователь, под которым запускается команда **fio**, должен иметь права на чтение/запись файлов в директории `/home/user`.

```
fio --directory=/home/user --direct=1 --rw=randread --bs=4k --ioengine=libaio --
iodepth=256 --runtime=60 --numjobs=3 --time_based --group_reporting --name=iops-
test-job --eta-newline=1 --size=1G
```

где:

- **--directory=/home/user** — каталог `/home/user`, в котором будут созданы тестовые файлы;
- **--direct=1** — включает режим прямого ввода-вывода, обходя кеширование в операционной системе;
- **--rw=randread** — тип выполняемых операций ввода-вывода (`randread` означает случайное чтение). В режиме `randread` **fio** будет случайным образом выбирать блоки для чтения;

- **--bs=4k** — устанавливает размер блока ввода-вывода на 4 КБ, каждая операция чтения будет обрабатывать 4 КБ данных;
- **--ioengine=libaio** — указывает механизм ввода-вывода, **libaio** использует асинхронные операции ввода-вывода, позволяет одновременно запускать несколько запросов ввода-вывода, не дожидаясь завершения предыдущих;
- **--iodepth=256** — задаёт глубину очереди операций ввода-вывода, 256 операций могут находиться в очереди на выполнение одновременно;
- **--runtime=60** — устанавливает время выполнения теста в секундах;
- **--numjobs=3** — определяет количество параллельно выполняемых заданий, тест будет выполняться с тремя потоками;
- **--time\_based** — указывает, что тест должен быть основан на времени, а не на объёме данных;
- **--group\_reporting** — включает групповой отчёт, результаты всех заданий будут объединены и представлены как один общий результат;
- **--name=iops-test-job** — имя задания;
- **--eta-newline=1** — опция, которая указывает **fio** обновлять строку с расчётным оставшимся временем (ETA) каждую секунду.

Результат:

iops-test-job: (g=0): rw=randread, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio,iodepth=256

...

fio-3.28

Starting 3 processes

Jobs: 3 (f=3): [r(3)][4.9%][r=129MiB/s][r=33.1k IOPS][eta 00m:58s]  
Jobs: 3 (f=3): [r(3)][6.6%][r=120MiB/s][r=30.7k IOPS][eta 00m:57s]  
Jobs: 3 (f=3): [r(3)][9.8%][r=128MiB/s][r=32.8k IOPS][eta 00m:55s]  
Jobs: 3 (f=3): [r(3)][13.1%][r=138MiB/s][r=35.4k IOPS][eta 00m:53s]  
Jobs: 3 (f=3): [r(3)][14.8%][r=139MiB/s][r=35.6k IOPS][eta 00m:52s]  
Jobs: 3 (f=3): [r(3)][16.4%][r=137MiB/s][r=35.0k IOPS][eta 00m:51s]  
Jobs: 3 (f=3): [r(3)][18.0%][r=139MiB/s][r=35.6k IOPS][eta 00m:50s]  
Jobs: 3 (f=3): [r(3)][19.7%][r=139MiB/s][r=35.7k IOPS][eta 00m:49s]  
Jobs: 3 (f=3): [r(3)][21.3%][r=133MiB/s][r=34.0k IOPS][eta 00m:48s]  
Jobs: 3 (f=3): [r(3)][24.6%][r=139MiB/s][r=35.7k IOPS][eta 00m:46s]  
Jobs: 3 (f=3): [r(3)][27.9%][r=138MiB/s][r=35.3k IOPS][eta 00m:44s]  
Jobs: 3 (f=3): [r(3)][29.5%][r=140MiB/s][r=35.8k IOPS][eta 00m:43s]  
Jobs: 3 (f=3): [r(3)][32.8%][r=139MiB/s][r=35.5k IOPS][eta 00m:41s]  
Jobs: 3 (f=3): [r(3)][36.1%][r=140MiB/s][r=35.8k IOPS][eta 00m:39s]  
Jobs: 3 (f=3): [r(3)][40.0%][r=139MiB/s][r=35.5k IOPS][eta 00m:36s]  
Jobs: 3 (f=3): [r(3)][41.0%][r=139MiB/s][r=35.7k IOPS][eta 00m:36s]  
Jobs: 3 (f=3): [r(3)][44.3%][r=140MiB/s][r=35.7k IOPS][eta 00m:34s]  
Jobs: 3 (f=3): [r(3)][47.5%][r=138MiB/s][r=35.4k IOPS][eta 00m:32s]  
Jobs: 3 (f=3): [r(3)][50.8%][r=139MiB/s][r=35.7k IOPS][eta 00m:30s]  
Jobs: 3 (f=3): [r(3)][53.3%][r=138MiB/s][r=35.2k IOPS][eta 00m:28s]  
Jobs: 3 (f=3): [r(3)][54.1%][r=126MiB/s][r=32.4k IOPS][eta 00m:28s]  
Jobs: 3 (f=3): [r(3)][57.4%][r=139MiB/s][r=35.7k IOPS][eta 00m:26s]  
Jobs: 3 (f=3): [r(3)][60.0%][r=139MiB/s][r=35.6k IOPS][eta 00m:24s]  
Jobs: 3 (f=3): [r(3)][60.7%][r=139MiB/s][r=35.7k IOPS][eta 00m:24s]  
Jobs: 3 (f=3): [r(3)][63.9%][r=139MiB/s][r=35.7k IOPS][eta 00m:22s]  
Jobs: 3 (f=3): [r(3)][65.6%][r=139MiB/s][r=35.7k IOPS][eta 00m:21s]  
Jobs: 3 (f=3): [r(3)][68.9%][r=139MiB/s][r=35.7k IOPS][eta 00m:19s]  
Jobs: 3 (f=3): [r(3)][70.5%][r=136MiB/s][r=34.9k IOPS][eta 00m:18s]  
Jobs: 3 (f=3): [r(3)][73.8%][r=134MiB/s][r=34.2k IOPS][eta 00m:16s]  
Jobs: 3 (f=3): [r(3)][78.3%][r=135MiB/s][r=34.5k IOPS][eta 00m:13s]  
Jobs: 3 (f=3): [r(3)][80.3%][r=135MiB/s][r=34.5k IOPS][eta 00m:12s]  
Jobs: 3 (f=3): [r(3)][83.6%][r=136MiB/s][r=34.7k IOPS][eta 00m:10s]  
Jobs: 3 (f=3): [r(3)][85.2%][r=133MiB/s][r=34.2k IOPS][eta 00m:09s]  
Jobs: 3 (f=3): [r(3)][86.9%][r=134MiB/s][r=34.3k IOPS][eta 00m:08s]  
Jobs: 3 (f=3): [r(3)][88.5%][r=133MiB/s][r=34.1k IOPS][eta 00m:07s]  
Jobs: 3 (f=3): [r(3)][91.8%][r=136MiB/s][r=34.8k IOPS][eta 00m:05s]  
Jobs: 3 (f=3): [r(3)][95.1%][r=128MiB/s][r=32.6k IOPS][eta 00m:03s]  
Jobs: 3 (f=3): [r(3)][96.7%][r=126MiB/s][r=32.2k IOPS][eta 00m:02s]  
Jobs: 3 (f=3): [r(3)][100.0%][r=130MiB/s][r=33.2k IOPS][eta 00m:00s]

iops-test-job: (groupid=0, jobs=3): err= 0: pid=3593: Tue Jul 30 21:43:19 2024

read: IOPS=34.7k, BW=136MiB/s (142MB/s)(8133MiB/60007msec)

slat (nsec): min=1166, max=97127k, avg=75924.61, stdev=510067.56

clat (usec): min=2, max=123674, avg=22035.95, stdev=4534.57

lat (msec): min=6, max=123, avg=22.11, stdev= 4.54

clat percentiles (msec):

| 1.00th=[ 16], 5.00th=[ 18], 10.00th=[ 19], 20.00th=[ 20],  
| 30.00th=[ 21], 40.00th=[ 21], 50.00th=[ 22], 60.00th=[ 23],  
| 70.00th=[ 24], 80.00th=[ 24], 90.00th=[ 26], 95.00th=[ 27],  
| 99.00th=[ 34], 99.50th=[ 40], 99.90th=[ 91], 99.95th=[ 101],  
| 99.99th=[ 117]

bw ( KiB/s): min=103294, max=148248, per=100.00%, avg=138926.83,

```

stdev=2412.68, samples=357
  iops      : min=25823, max=37062, avg=34731.20, stdev=603.19, samples=357
  lat (usec) : 4=0.01%
  lat (msec) : 10=0.03%, 20=27.34%, 50=72.31%, 100=0.25%, 250=0.06%
  cpu       : usr=2.31%, sys=13.81%, ctx=45133, majf=0, minf=805
  IO depths  : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
    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=0.1%
    issued rwts: total=2081938,0,0,0 short=0,0,0,0 dropped=0,0,0,0
    latency   : target=0, window=0, percentile=100.00%, depth=256

Run status group 0 (all jobs):
  READ: bw=136MiB/s (142MB/s), 136MiB/s-136MiB/s (142MB/s-142MB/s), io=8133MiB
(8528MB), run=60007-60007msec

Disk stats (read/write):
  sda: ios=2055328/37, merge=82/22, ticks=487922/130, in_queue=488058, util=80.34%

```

где:

- **iops** — среднее количество операций ввода/вывода в секунду (34,700 операций чтения в секунду);
- **bw** — средняя скорость передачи данных — 136 МБ/с;
- **latency** — среднее время, которое требуется для завершения одной операции чтения — 22.11 миллисекунды;
- **clat percentiles** — показывает время завершения операций (completion latency, clat) в миллисекундах (ms):
  - 1% операций завершается за 16 мс или меньше;
  - 50% операций завершается за 22 мс или меньше;
  - 99% операций завершается за 34 мс или меньше.
- **cpu** — процентное использование процессора: usr=2.31%, sys=13.81% (2.31% используется для пользовательских задач, 13.81% для системных задач), ctx - Количество переключений контекста (context switches) между процессами. Много переключений контекста может указывать на сильную конкуренцию за ресурсы или неэффективную работу с очередями I/O.
- **IO depths** — все операции ввода/вывода были выполнены с глубиной очереди 64 или больше: >=64=100.0%;
- **Run status group:**
  - io - общее количество данных прочитано: 8133 МБ (8.528 ГБ);
  - bw (Bandwidth) - средняя скорость чтения: 136 МБ/с;
  - run - общее время выполнения теста: 60007 мс (60.007 секунд);

- **Disk stats (read/write):**

- ios=2055328/37 - выполнено 2055328 операций чтения и 37 операций записи.
- merge=82/22 - показатель, сколько операций чтения/записи были объединены в одно обращение.
- ticks - Время обработки операций ввода-вывода: 487922 мс затрачено на операции чтения, 130 мс затрачено на операции записи.
- in\_queue - отражает суммарное время ожидания всех запросов в миллисекундах.
- util - Утилизация диска (время, в течение которого устройство было занято обработкой операций) составила 80.34%.

Результат показывает:

- выполнение операций чтения с пропускной способностью в 136 МБ/с и 34.7k IOPS;
- среднее время задержки в 22 мс. Высокий показатель 95-го перцентиля (27 мс) указывает на стабильное выполнение большинства операций что задержки относительно стабильны и приемлемы для большинства операций;
- процент использования процессора относительно низкий, что указывает на то, что CPU не является узким местом в тесте.

## Тест скорости чтения/записи

---

```
fio --directory=/home/user --direct=1 --rw=randrw --bs=4k --ioengine=libaio --  
iodepth=256 --runtime=60 --numjobs=3 --time_based --group_reporting --name=iops-  
test-job --eta-newline=1
```

где:

- **--directory=/home/user** — директория, где будут производиться тесты ввода-вывода;
- **--direct=1** — активирует прямой ввод-вывод, что означает, что операции не будут кешироваться операционной системой;
- **--rw=randrw** — режим работы, тест будет одновременно выполнять операции чтения и записи в случайном порядке;
- **--bs=4k** — устанавливает размер блока для операций ввода-вывода, размер блока равен 4 КБ (4096 байт);

- **--ioengine=libaio** — механизм ввода-вывода, libaio — асинхронный ввод-вывод через библиотеку libaio;
- **--iodepth=256** — определяет глубину очереди ввода-вывода (256 — максимальное количество операций ввода-вывода, которые могут находиться в ожидании одновременно);
- **--runtime=60** — устанавливает время работы теста в секундах — 60 секунд (1 минута);
- **--numjobs=3** — количество задач, которые будут выполняться параллельно;
- **--time\_based** — задаёт режим тестирования по времени, а не по количеству операций;
- **--group\_reporting** — указывает, что результаты теста должны группироваться и выводиться в одном отчёте для всех задач, а не по отдельности для каждой;
- **--name=iops-test-job** — задаёт имя теста, которое будет отображаться в результатах;
- **--eta-newline=1** — выводит оставшееся время в следующей строке в процессе выполнения теста.

Результат:

```

iops-test-job: (g=0): rw=randrw, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio, iodepth=256
...
fio-3.28
Starting 3 processes
fio: io_u error on file /home/user/iops-test-job.1.0: No space left on device:
write offset=1053564928, buflen=4096
fio: io_u error on file /home/user/iops-test-job.1.0: No space left on device:
write offset=1066758144, buflen=4096
fio: io_u error on file /home/user/iops-test-job.1.0: No space left on device:
write offset=1072381952, buflen=4096
fio: io_u error on file /home/user/iops-test-job.1.0: No space left on device:
write offset=1066737664, buflen=4096
fio: io_u error on file /home/user/iops-test-job.1.0: No space left on device:
write offset=1039486976, buflen=4096
fio: io_u error on file /home/user/iops-test-job.1.0: No space left on device:
write offset=1033895936, buflen=4096
fio: pid=3773, err=28/file:io_u.c:1845, func=io_u error, error=No space left on
device
Jobs: 2 (f=2): [m(2),X(1)][4.9%][r=47.4MiB/s,w=46.8MiB/s][r=12.1k,w=12.0k IOPS]
[eta 00m:58s]
Jobs: 2 (f=2): [m(2),X(1)][8.2%][r=51.2MiB/s,w=50.3MiB/s][r=13.1k,w=12.9k IOPS]
[eta 00m:56s]
Jobs: 2 (f=2): [m(2),X(1)][9.8%][r=47.4MiB/s,w=48.7MiB/s][r=12.1k,w=12.5k IOPS]
[eta 00m:55s]
Jobs: 2 (f=2): [m(2),X(1)][11.5%][r=44.5MiB/s,w=44.5MiB/s][r=11.4k,w=11.4k IOPS]
[eta 00m:54s]
Jobs: 2 (f=2): [m(2),X(1)][13.1%][r=47.0MiB/s,w=46.6MiB/s][r=12.0k,w=11.9k IOPS]
[eta 00m:53s]
Jobs: 2 (f=2): [m(2),X(1)][16.4%][r=44.6MiB/s,w=45.0MiB/s][r=11.4k,w=11.5k IOPS]
[eta 00m:51s]
Jobs: 2 (f=2): [m(2),X(1)][18.0%][r=47.4MiB/s,w=47.5MiB/s][r=12.1k,w=12.2k IOPS]
[eta 00m:50s]
Jobs: 2 (f=2): [m(2),X(1)][21.3%][r=43.3MiB/s,w=43.6MiB/s][r=11.1k,w=11.2k IOPS]
[eta 00m:48s]
Jobs: 2 (f=2): [m(2),X(1)][23.0%][r=43.4MiB/s,w=43.7MiB/s][r=11.1k,w=11.2k IOPS]
[eta 00m:47s]
Jobs: 2 (f=2): [m(2),X(1)][26.2%][r=47.7MiB/s,w=47.3MiB/s][r=12.2k,w=12.1k IOPS]
[eta 00m:45s]
Jobs: 2 (f=2): [m(2),X(1)][29.5%][r=46.8MiB/s,w=47.2MiB/s][r=12.0k,w=12.1k IOPS]
[eta 00m:43s]
Jobs: 2 (f=2): [m(2),X(1)][32.8%][r=47.6MiB/s,w=47.4MiB/s][r=12.2k,w=12.1k IOPS]
[eta 00m:41s]
Jobs: 2 (f=2): [m(2),X(1)][36.1%][r=46.5MiB/s,w=47.1MiB/s][r=11.9k,w=12.1k IOPS]
[eta 00m:39s]
Jobs: 2 (f=2): [m(2),X(1)][40.0%][r=50.2MiB/s,w=49.8MiB/s][r=12.9k,w=12.7k IOPS]
[eta 00m:36s]
Jobs: 2 (f=2): [m(2),X(1)][41.0%][r=49.3MiB/s,w=49.1MiB/s][r=12.6k,w=12.6k IOPS]
[eta 00m:36s]
Jobs: 2 (f=2): [m(2),X(1)][44.3%][r=47.6MiB/s,w=47.6MiB/s][r=12.2k,w=12.2k IOPS]
[eta 00m:34s]
Jobs: 2 (f=2): [m(2),X(1)][47.5%][r=49.8MiB/s,w=49.7MiB/s][r=12.8k,w=12.7k IOPS]
[eta 00m:32s]
Jobs: 2 (f=2): [m(2),X(1)][49.2%][r=49.0MiB/s,w=48.8MiB/s][r=12.6k,w=12.5k IOPS]
[eta 00m:31s]
Jobs: 2 (f=2): [m(2),X(1)][53.3%][r=48.3MiB/s,w=49.3MiB/s][r=12.4k,w=12.6k IOPS]

```

```

[eta 00m:28s]
Jobs: 2 (f=2): [m(2),X(1)][56.7%][r=48.6MiB/s,w=49.0MiB/s][r=12.4k,w=12.5k IOPS]
[eta 00m:26s]
Jobs: 2 (f=2): [m(2),X(1)][60.0%][r=48.0MiB/s,w=47.8MiB/s][r=12.3k,w=12.2k IOPS]
[eta 00m:24s]
Jobs: 2 (f=2): [m(2),X(1)][60.7%][r=48.0MiB/s,w=48.1MiB/s][r=12.3k,w=12.3k IOPS]
[eta 00m:24s]
Jobs: 2 (f=2): [m(2),X(1)][63.9%][r=51.0MiB/s,w=50.6MiB/s][r=13.1k,w=13.0k IOPS]
[eta 00m:22s]
Jobs: 2 (f=2): [m(2),X(1)][67.2%][r=49.8MiB/s,w=49.5MiB/s][r=12.7k,w=12.7k IOPS]
[eta 00m:20s]
Jobs: 2 (f=2): [m(2),X(1)][70.5%][r=49.0MiB/s,w=50.3MiB/s][r=12.5k,w=12.9k IOPS]
[eta 00m:18s]
Jobs: 2 (f=2): [m(2),X(1)][73.8%][r=48.5MiB/s,w=49.4MiB/s][r=12.4k,w=12.7k IOPS]
[eta 00m:16s]
Jobs: 2 (f=2): [m(2),X(1)][78.3%][r=48.2MiB/s,w=47.3MiB/s][r=12.3k,w=12.1k IOPS]
[eta 00m:13s]
Jobs: 2 (f=2): [m(2),X(1)][80.3%][r=50.3MiB/s,w=49.4MiB/s][r=12.9k,w=12.7k IOPS]
[eta 00m:12s]
Jobs: 2 (f=2): [m(2),X(1)][83.6%][r=50.2MiB/s,w=49.0MiB/s][r=12.9k,w=12.5k IOPS]
[eta 00m:10s]
Jobs: 2 (f=2): [m(2),X(1)][85.2%][r=47.5MiB/s,w=47.7MiB/s][r=12.2k,w=12.2k IOPS]
[eta 00m:09s]
Jobs: 2 (f=2): [m(2),X(1)][86.9%][r=50.1MiB/s,w=49.5MiB/s][r=12.8k,w=12.7k IOPS]
[eta 00m:08s]
Jobs: 2 (f=2): [m(2),X(1)][90.2%][r=48.8MiB/s,w=49.6MiB/s][r=12.5k,w=12.7k IOPS]
[eta 00m:06s]
Jobs: 2 (f=2): [m(2),X(1)][93.4%][r=48.4MiB/s,w=47.4MiB/s][r=12.4k,w=12.1k IOPS]
[eta 00m:04s]
Jobs: 2 (f=2): [m(2),X(1)][96.7%][r=47.5MiB/s,w=48.7MiB/s][r=12.2k,w=12.5k IOPS]
[eta 00m:02s]
Jobs: 2 (f=2): [m(2),X(1)][98.4%][r=49.8MiB/s,w=49.5MiB/s][r=12.8k,w=12.7k IOPS]
[eta 00m:01s]
Jobs: 2 (f=2): [m(2),X(1)][100.0%][r=48.9MiB/s,w=47.9MiB/s][r=12.5k,w=12.3k IOPS]
[eta 00m:00s]
iops-test-job: (groupid=0, jobs=3): err=28 (file:io_u.c:1845, func=io_u error,
error=No space left on device): pid=3771: Tue Jul 30 23:06:17 2024
  read: IOPS=12.3k, BW=48.1MiB/s (50.4MB/s)(2886MiB/60002msec)
    slat (nsec): min=1679, max=61519k, avg=57980.28, stdev=147149.84
    clat (usec): min=1126, max=163039, avg=20880.35, stdev=5236.80
    lat (usec): min=1188, max=163066, avg=20938.57, stdev=5236.55
    clat percentiles (usec):
      | 1.00th=[14746], 5.00th=[15664], 10.00th=[16057], 20.00th=[16581],
      | 30.00th=[17171], 40.00th=[17957], 50.00th=[18744], 60.00th=[21627],
      | 70.00th=[23200], 80.00th=[24773], 90.00th=[27132], 95.00th=[31327],
      | 99.00th=[36963], 99.50th=[39060], 99.90th=[49021], 99.95th=[51643],
      | 99.99th=[76022]
    bw ( KiB/s): min=35632, max=53506, per=100.00%, avg=49309.31, stdev=1275.25,
samples=236
    iops      : min= 8907, max=13376, avg=12326.85, stdev=318.82, samples=236
  write: IOPS=12.3k, BW=48.1MiB/s (50.5MB/s)(2888MiB/60002msec); 0 zone resets
    slat (usec): min=2, max=112744, avg=98.70, stdev=586.77
    clat (usec): min=1013, max=162272, avg=20443.88, stdev=5185.99
    lat (usec): min=1125, max=163299, avg=20542.90, stdev=5202.03
    clat percentiles (usec):
      | 1.00th=[14353], 5.00th=[15401], 10.00th=[15795], 20.00th=[16319],

```



```

| 30.00th=[16712], 40.00th=[17433], 50.00th=[18220], 60.00th=[21103],
| 70.00th=[22938], 80.00th=[24249], 90.00th=[26608], 95.00th=[30802],
| 99.00th=[36439], 99.50th=[38536], 99.90th=[48497], 99.95th=[51119],
| 99.99th=[76022]
bw ( KiB/s): min=37137, max=53249, per=100.00%, avg=49333.05, stdev=1188.12,
samples=236
iops      : min= 9283, max=13312, avg=12332.74, stdev=297.01, samples=236
lat (msec) : 2=0.01%, 4=0.01%, 10=0.02%, 20=56.72%, 50=43.18%
lat (msec) : 100=0.07%, 250=0.01%
cpu       : usr=2.87%, sys=16.33%, ctx=447882, majf=0, minf=56
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
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=0.1%
issued rwts: total=738903,739381,0,0 short=0,0,0,0 dropped=0,0,0,0
latency   : target=0, window=0, percentile=100.00%, depth=256

Run status group 0 (all jobs):
  READ: bw=48.1MiB/s (50.4MB/s), 48.1MiB/s-48.1MiB/s (50.4MB/s-50.4MB/s),
io=2886MiB (3027MB), run=60002-60002msec
  WRITE: bw=48.1MiB/s (50.5MB/s), 48.1MiB/s-48.1MiB/s (50.5MB/s-50.5MB/s),
io=2888MiB (3028MB), run=60002-60002msec

Disk stats (read/write):
  sda: ios=738033/738664, merge=43/87, ticks=767404/135599, in_queue=903136,
util=72.27%
```

## Тест пропускной способности

---

Для тестирования пропускной способности используется одновременное чтение и запись крупными блоками с использованием множества процессов. Эти тесты измеряют значение объёма данных, который может быть передан за определённое время. Чем выше значение пропускной способности, тем больше данных может быть передано или принято за единицу времени.

**bw** — основной параметр вывода для теста пропускной способности, означающий совокупную пропускную способность потоков в группе, за которой следуют минимальная и максимальная пропускная способность всех потоков.

Перед выполнением теста убедитесь, что в указанной директории достаточно свободного места (более 10 ГБ) для завершения теста.

Тест чтения/записи:

```
fio --name=fiotest --blocksize=256k --directory=/home/user --size=10G --
ioengine=psync --iodepth=1 --runtime=120 --rw=randrw --direct=1 --numjobs=25 --
group_reporting
```

где:

- **--name=fiotest** — имя задания, используемое для отчётов;

- **--blocksize=256k** — устанавливает размер читаемого или записываемого блока данных на 256 КБ;
- **--directory=/home/user** — задаёт каталог /home/user, в котором будут созданы тестовые файлы;
- **--size=10G** — общий объем данных, который будет использоваться для каждого задания (10 ГБ);
- **--ioengine=psync** — устанавливает синхронный механизм ввода-вывода с использованием стандартных системных вызовов (pread/pwrite);
- **--iodepth=1** — глубина очереди ввода-вывода, которая равна 1, каждая операция будет дожидаться завершения перед началом следующей;
- **--runtime=120** — устанавливает время выполнения теста в 120 секунд;
- **--rw=randrw** — режим работы с диском: данные будут читаться и записываться случайным образом;
- **--direct=1** — включает режим прямого ввода-вывода, минуя кеширование в операционной системе;
- **--numjobs=25** — количество параллельно выполняемых заданий (25 заданий);
- **--group\_reporting** — включает групповой отчёт, объединяющий результаты всех заданий.

## Использование файла задания

---

Можно получить более последовательные результаты, если использовать файл задания вместо выполнения команды напрямую.

Для этого создайте файл задания **fiorandomreadwrite.fio**:

```
nano fiorandomreadwrite.fio
```

Добавьте в файл следующие строки:

```
[global]
name=fiotest
blocksize=4k
directory=/home/user
size=5g
ioengine=libaio
iodepth=256
runtime=60
rw=randwrite
```

```
direct=1
runtime=60
rw=randwrite
direct=1
[job1]
filename=directory=/home/user
```

Для выполнения файла задания используйте команду:

```
fio ./fiorandomreadwrite.fio
```

Результат:

```
job1: (g=0): rw=randwrite, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B,
ioengine=libaio, iodepth=256
fio-3.28
Starting 1 process
job1: Laying out IO file (1 file / 5120MiB)
Jobs: 1 (f=1): [w(1)][100.0%][w=53.5MiB/s][w=13.7k IOPS][eta 00m:00s]
job1: (groupid=0, jobs=1): err= 0: pid=4312: Tue Jul 30 12:29:40 2024
  write: IOPS=5104, BW=19.9MiB/s (20.9MB/s)(1197MiB/60001msec); 0 zone resets
    slat (usec): min=4, max=6968.9k, avg=192.75, stdev=21138.71
    clat (nsec): min=884, max=8563.0M, avg=49933698.53, stdev=420390411.52
      lat (usec): min=89, max=8563.0k, avg=50126.78, stdev=421252.44
    clat percentiles (msec):
      | 1.00th=[ 11], 5.00th=[ 12], 10.00th=[ 12], 20.00th=[ 13],
      | 30.00th=[ 13], 40.00th=[ 14], 50.00th=[ 14], 60.00th=[ 14],
      | 70.00th=[ 14], 80.00th=[ 18], 90.00th=[ 25], 95.00th=[ 43],
      | 99.00th=[ 342], 99.50th=[ 2039], 99.90th=[ 7282], 99.95th=[ 8423],
      | 99.99th=[ 8423]
    bw ( KiB/s): min= 8, max=74003, per=100.00%, avg=38357.61, stdev=26329.13,
samples=61
    iops        : min= 2, max=18500, avg=9589.21, stdev=6582.25, samples=61
    lat (nsec)   : 1000=0.01%
    lat (usec)   : 50=0.01%, 100=0.01%, 250=0.01%, 500=0.01%, 750=0.01%
    lat (usec)   : 1000=0.01%
    lat (msec)   : 2=0.01%, 4=0.04%, 10=0.45%, 20=82.69%, 50=12.38%
    lat (msec)   : 100=2.49%, 250=0.46%, 500=0.70%, 750=0.09%, 1000=0.01%
    lat (msec)   : 2000=0.16%, >=2000=0.50%
    cpu          : usr=1.00%, sys=15.06%, ctx=334602, majf=0, minf=16
    IO depths    : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
      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=0.1%
      issued rwts: total=0,306304,0,0 short=0,0,0,0 dropped=0,0,0,0
      latency    : target=0, window=0, percentile=100.00%, depth=256
```

Run status group 0 (all jobs):

```
  WRITE: bw=19.9MiB/s (20.9MB/s), 19.9MiB/s-19.9MiB/s (20.9MB/s-20.9MB/s),
io=1197MiB (1255MB), run=60001-60001msec
```

Disk stats (read/write):

```
  sda: ios=4/304465, merge=0/6666, ticks=446/976801, in_queue=983934, util=84.55%
```

## Тест с разными значениями iodepth

---

Перед выполнением теста убедитесь, что в указанной директории достаточно свободного места (более 5 ГБ) для завершения теста.

Основная цель данного теста — измерить производительность дисковой подсистемы при различных значениях `iodepth`.

Значение `iodepth` определяет, сколько операций ввода-вывода может быть выполнено одновременно. Чем **выше значение** `iodepth`, тем **выше нагрузка** на систему, и тест позволяет увидеть, как это влияет на скорость чтения/записи.

Тест можно использовать для:

- оценки производительности устройства хранения;
- выявления пределов производительности для различных сценариев загрузки;
- сравнения различных устройств хранения;
- оптимизации конфигураций серверов/приложений, особенно при высокой нагрузке.

Скрипт для тестирования диска с разными значениями `iodepth` (разным количеством одновременных запросов):

Создайте файл **run\_fio\_tests.sh**:

```
nano run_fio_tests.sh
```

Добавьте в файл следующие строки:

```
#!/bin/bash
for i in 1 2 4 8 16 32 64 128 256 512 1024; do
echo "Testing with iodepth=$i"
fio --name=fiotest --ioengine=libaio --direct=1 --rw=randread --bs=4k --
size=5G --randrepeat=0 --norandommap --fsync=1 --numjobs=1 --runtime=60 --
group_reporting --directory=/base/fio_test --iodepth=$i
echo ""
done
где:
```

- **--name=fiotest** — название теста;
- **--ioengine=libaio** — задаёт используемый механизм ввода-вывода. `libaio` — асинхронный механизм ввода-вывода, который позволяет программе выполнять ввод-вывод в неблокирующем режиме;
- **--direct=1** — ввод-вывод будет осуществляться непосредственно на устройство, минуя кеш операционной системы;

- **--rw=randread** — тип операции ввода-вывода (read/write). randread означает, что будут выполняться случайные операции чтения;
- **--bs=4k** — размер блока ввода-вывода 4 КБ;
- **--size=5G** — определяет общий размер файла — 5 ГБ, который будет использоваться в тестировании;
- **--randrepeat=0** — указывает, что не нужно повторять одно и то же случайное последовательности ввода-вывода;
- **--norandommap** — отключает создание карты случайных блоков;
- **--fsync=1** — после каждой записи необходимо синхронизировать данные с устройством хранения;
- **--numjobs=1** — определяет количество параллельных задач, которые будут запущены. В примере 1 задача;
- **--runtime=60** — определяет продолжительность теста в секундах. Тест будет выполняться 60 секунд;
- **--group\_reporting** — результаты должны быть собраны и отображены глобально;
- **--directory=/home/user** — определяет директорию, где будет осуществляться тестирование;
- **--iodepth=\$i** — глубина очереди ввода-вывода, которая будет варьироваться от 1 до 1024 в зависимости от текущего значения \$i в цикле.

Сделайте файл исполняемым:

```
chmod +x run_fio_tests.sh
```

Запустите скрипт:

```
./run_fio_tests.sh
```

## Удаление результатов тестирования

---

Удалить файлы после тестирования можно с помощью следующей команды:

```
rm /home/user/fiotest*
```

где:

**/home/user/fiotest\*** — путь и шаблон имён файлов, которые нужно удалить.

Перед удалением можно проверить, что удаляются только нужные файлы:

```
ls /home/user/fiotest*
```

Примерный вывод:

```
/home/user/fiotest.0.0 /home/user/fiotest.1.0
```

**Дата последнего изменения:** 20.11.2024