Exploring Linux Memory Usage and Disk IO performance

Frits Hoogland
Yugabyte

Topic: disk IO and disk IO performance

- The main topic of this presentation is disk IO performance on linux
- In order to understand disk IO performance, a detailed understanding of disk IO and related technologies is necessary.
- This presentation explains some the mechanics, in order to let the attendee understand buffered disk IO performance better.
- Memory usage is quite fundamentally closely related to cached IO performance.

Disk IO and memory

- Any regular disk IO is performed *buffered.*
- Buffered means: using the operating system memory for caching.

- You can do IO *without* using the operating system for caching.
  - Only if you explicitly request it: O_DIRECT.
  - Makes sense if you don't want to stage memory in two caches.

- If you are not sure which you are using *you are quite probably doing buffered IO.*

Where does buffered IO go?

- Linux does not have a dedicated memory area as 'page cache'.
  - Traditional Unix such as HPUX and AIX have that.

- Buffered IO *must* allocate memory to store the IO.
  - Even if that means it will get removed immediately b/c memory pressure(!)
  - Writes are special.

- Therefore it *competes* with regular memory usage.

Okay: but where does buffered IO go?

- Linux provides an insight into its memory usage via `/proc/meminfo`
  - Which is a messy gathering of memory related statistics.

- Named values in 'meminfo' do contain overlapping memory allocations, and can contain multiple, different allocations.

- Roughly put, it should be in 'Cached', 'Dirty' and 'Mapped', which can contain other allocated memory, such as shared memory.

You are not really making it understandable!

- I know.
- I think it's wrong to try to capture page cache size.
- You must have memory that is usable for buffering purpose.
  - Which is also memory for application usage.
- The best way to assess usable memory is use `MemAvailable`

How about 'MemFree'?

- There also is the `MemFree` statistic in 'meminfo'?

- `MemFree` is not 'free' as in available.
  - It is a small amount of memory pre-cleaned for direct usage.
  - There will be lots after startup, because it was never touched.

- Linux tries to do the bare minimum, and thus keep used memory around.
  - And thus to reduce `MemFree` to a minimum (`vm.min_free_kbytes`).
  - The swapper force-frees memory. (Page daemon)
  - Processes explicitly freeing memory will add to `MemFree`.
  - https://dev.to/yugabyte/what-is-free-memory-in-linux-18km

MemAvailable

- Statistic in `/proc/meminfo`.
- Kernel estimation of available memory without requiring swapping.
- Many of the other statistics (in /proc/meminfo) contain information, are useful, but do not provide a full picture to assess available memory.

Why is this important actually?

- Buffering can do miracles for IO performance*.
- Equally it can do "miracles" for container/application performance.

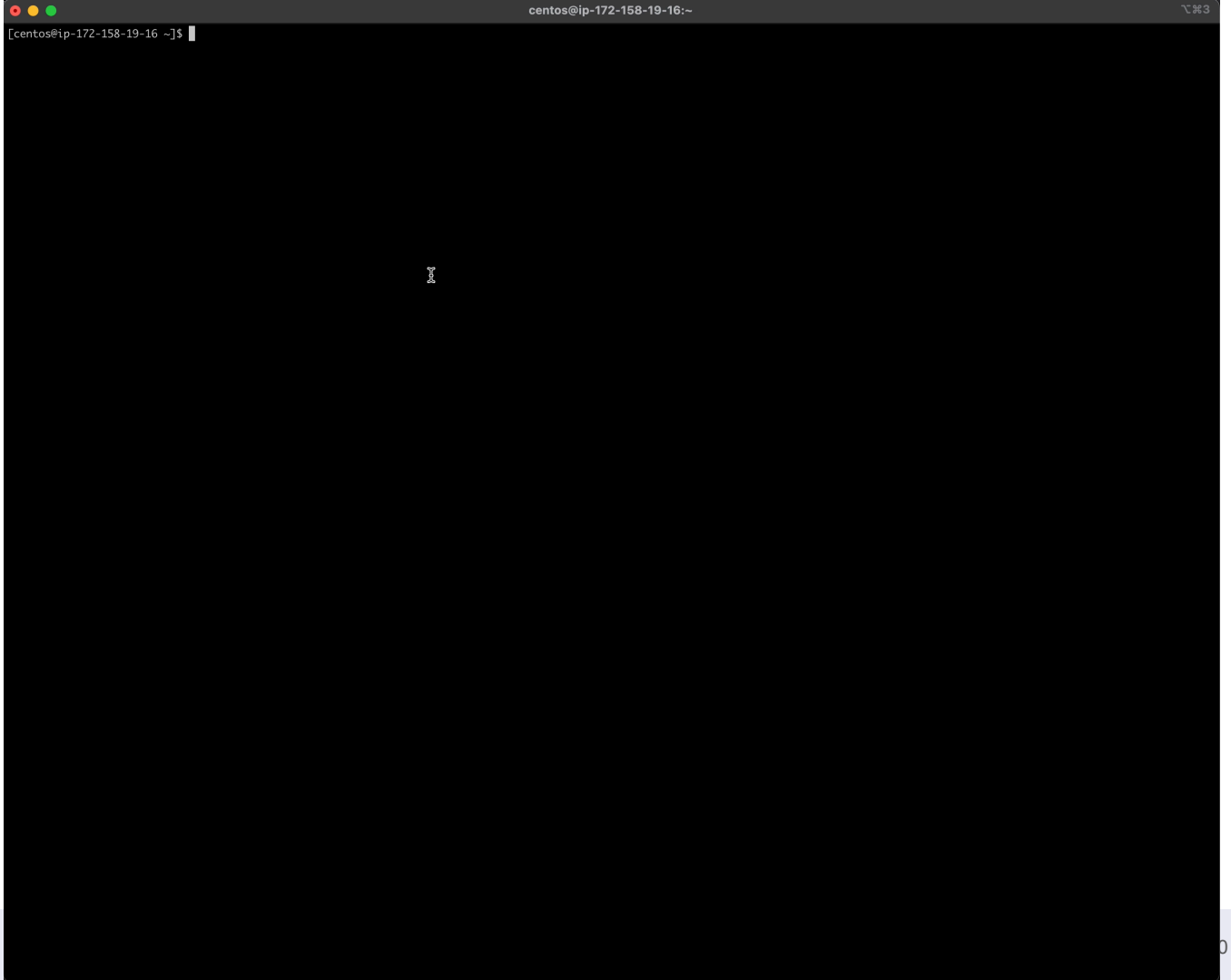
Let's test!

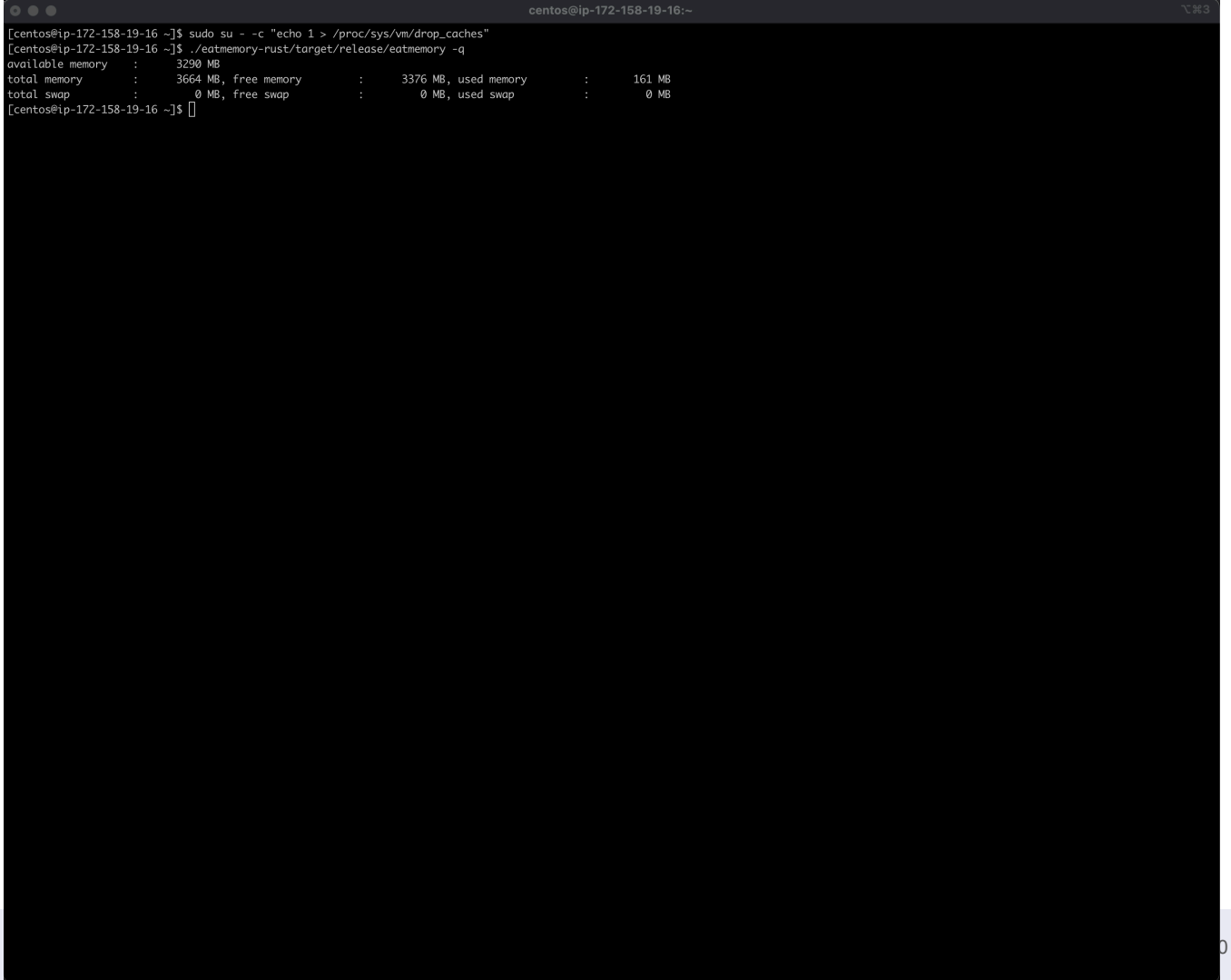
- Tests done on Amazon EC2:
  - c5.large VM (20000/4000 IOPS, 594/82 MBPS)
  - EBS: GP3 250M (3000 IOPS, 125 MBPS)
- I am not running into my bursting limits so concrete:
  - IOPS: 3000
  - MBPS: 125
- EC2 VM limits page:https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-optimized.html
  - Not easy to find.

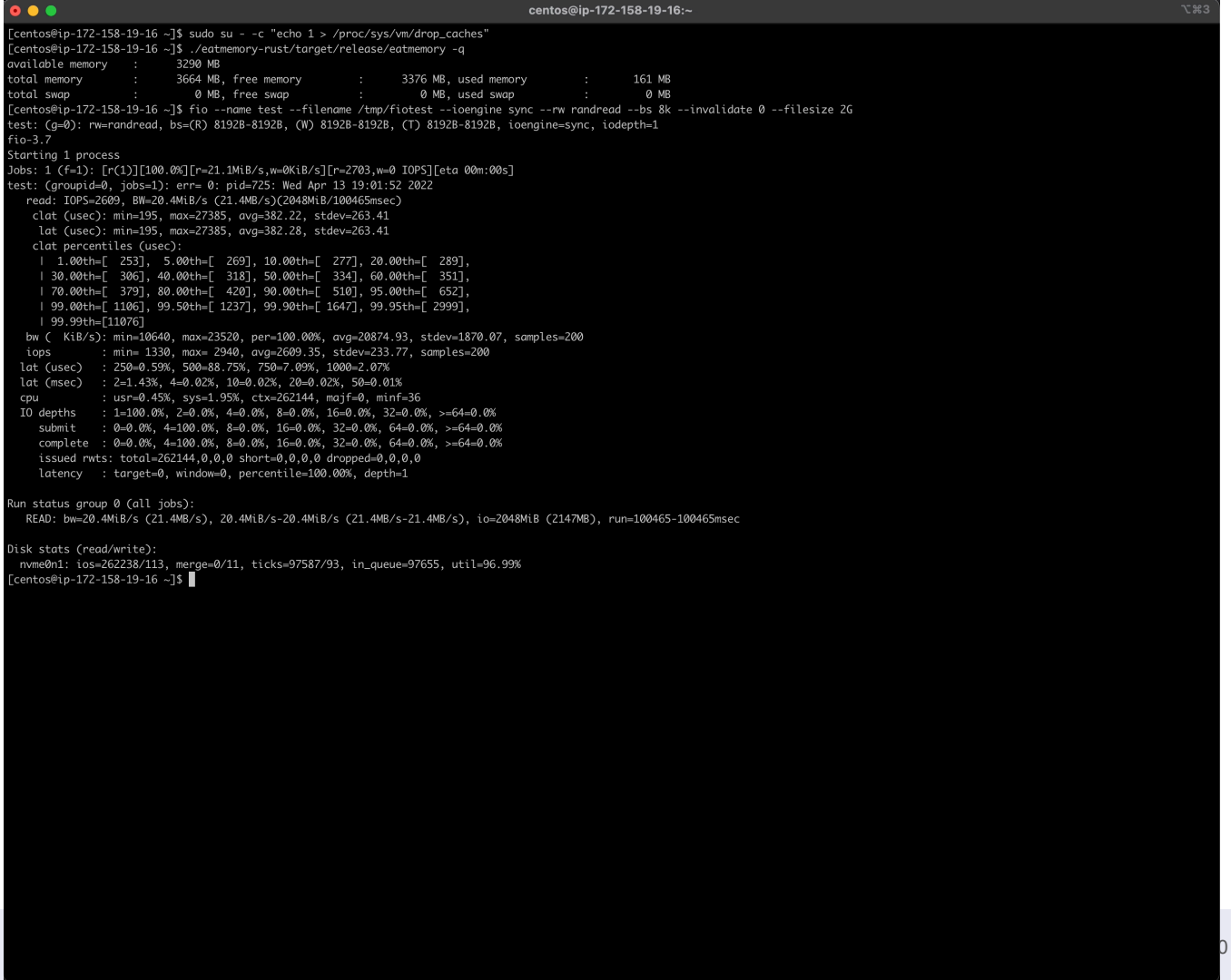
Read: 2G

- Drop the page cache
- Validate available memory
- Run a fio **read** test reading 2G:

```
fio --name test --filename /tmp/fiotest
    --ioengine sync --rw randread --bs 8k
    --invalidate 0 --filesize 2G
```







Read: 2G

- This is a summary from the run:

```
IOPS=2609, BW=20.4MiB/s (21.4MB/s)(2048MiB/100465msec)
```

- My limits are 125 MBPS and 3000 IOPS.
- Why didn't we reach any of these? Is AWS lying?
- No: look at the latency:

```
clat (usec): min=195, max=27385, avg=382.22, stdev=263.41
```

- 382 (avg usec) * 2609 (IOPS) ≈ 996638 ≈ 1 second: latency bound!

Read: 2G

- Now lets perform the exact same run again

```
centos@ip-172-158-19-16:~
[centos@ip-172-158-19-16 ~]$ sudo su - -c "echo 1 > /proc/sys/vm/drop_caches"
[centos@ip-172-158-19-16 ~]$ ./eatmemory-rust/target/release/eatmemory -q
available memory    :      3290 MB
total memory        :      3664 MB, free memory         :      3376 MB, used memory         :       161 MB
total swap          :         0 MB, free swap           :         0 MB, used swap           :         0 MB
[centos@ip-172-158-19-16 ~]$ fio --name test --filename /tmp/fiotest --ioengine sync --rw randread --bs 8k --invalidate 0 --filesize 2G
test: (g=0): rw=randread, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=sync, iodepth=1
fio-3.7
Starting 1 process
Jobs: 1 (f=1): [r(1)][100.0%][r=21.1MiB/s,w=0KiB/s][r=2703,w=0 IOPS][eta 00m:00s]
test: (groupid=0, jobs=1): err= 0: pid=725: Wed Apr 13 19:01:52 2022
   read: IOPS=2609, BW=20.4MiB/s (21.4MB/s)(2048MiB/100465msec)
    clat (usec): min=195, max=27385, avg=382.22, stdev=263.41
     lat (usec): min=195, max=27385, avg=382.28, stdev=263.41
    clat percentiles (usec):
     |  1.00th=[  253],  5.00th=[  269], 10.00th=[  277], 20.00th=[  289],
     | 30.00th=[  306], 40.00th=[  318], 50.00th=[  334], 60.00th=[  351],
     | 70.00th=[  379], 80.00th=[  420], 90.00th=[  510], 95.00th=[  652],
     | 99.00th=[ 1106], 99.50th=[ 1237], 99.90th=[ 1647], 99.95th=[ 2999],
     | 99.99th=[11076]
   bw (  KiB/s): min=10640, max=23520, per=100.00%, avg=20874.93, stdev=1870.07, samples=200
   iops        : min= 1330, max= 2940, avg=2609.35, stdev=233.77, samples=200
  lat (usec)   : 250=0.59%, 500=88.75%, 750=7.09%, 1000=2.07%
  lat (msec)   : 2=1.43%, 4=0.02%, 10=0.02%, 20=0.02%, 50=0.01%
  cpu          : usr=0.45%, sys=1.95%, ctx=262144, majf=0, minf=36
  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 rwts: total=262144,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=20.4MiB/s (21.4MB/s), 20.4MiB/s-20.4MiB/s (21.4MB/s-21.4MB/s), io=2048MiB (2147MB), run=100465-100465msec

Disk stats (read/write):
  nvme0n1: ios=262238/113, merge=0/11, ticks=97587/93, in_queue=97655, util=96.99%
[centos@ip-172-158-19-16 ~]$
```

Read: 2G

- This is quite much different, isn't it?

```
read: IOPS=585k, BW=4571MiB/s (4793MB/s)(2048MiB/448msec)
```

- My limits are 125 MBPS and 3000 IOPS.
- Now I did 585000 IOPS and 4571 MBPS!

```
clat (nsec): min=893, max=19771, avg=1342.90, stdev=406.95
```

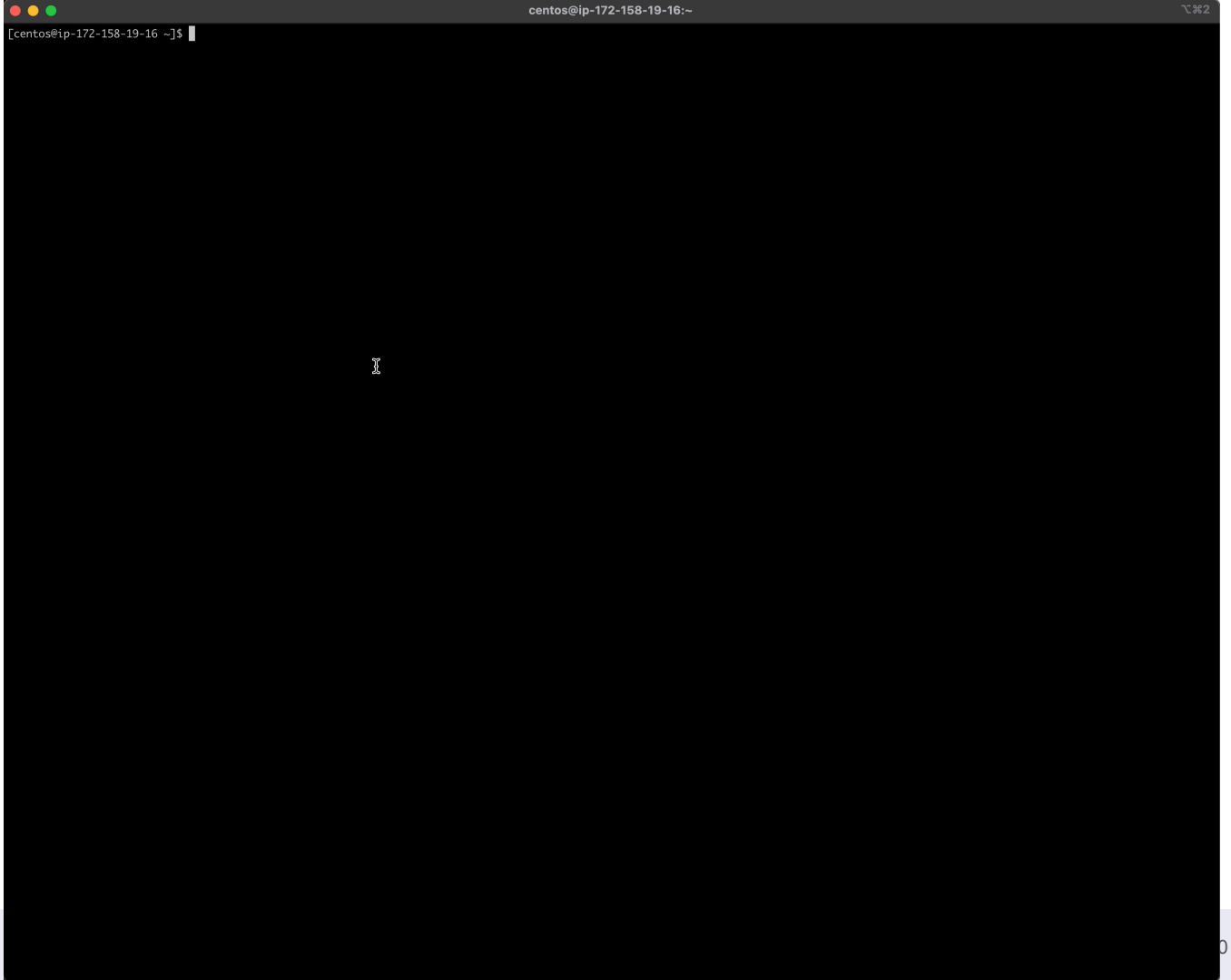
- It was all cache, no physical IOs were performed:

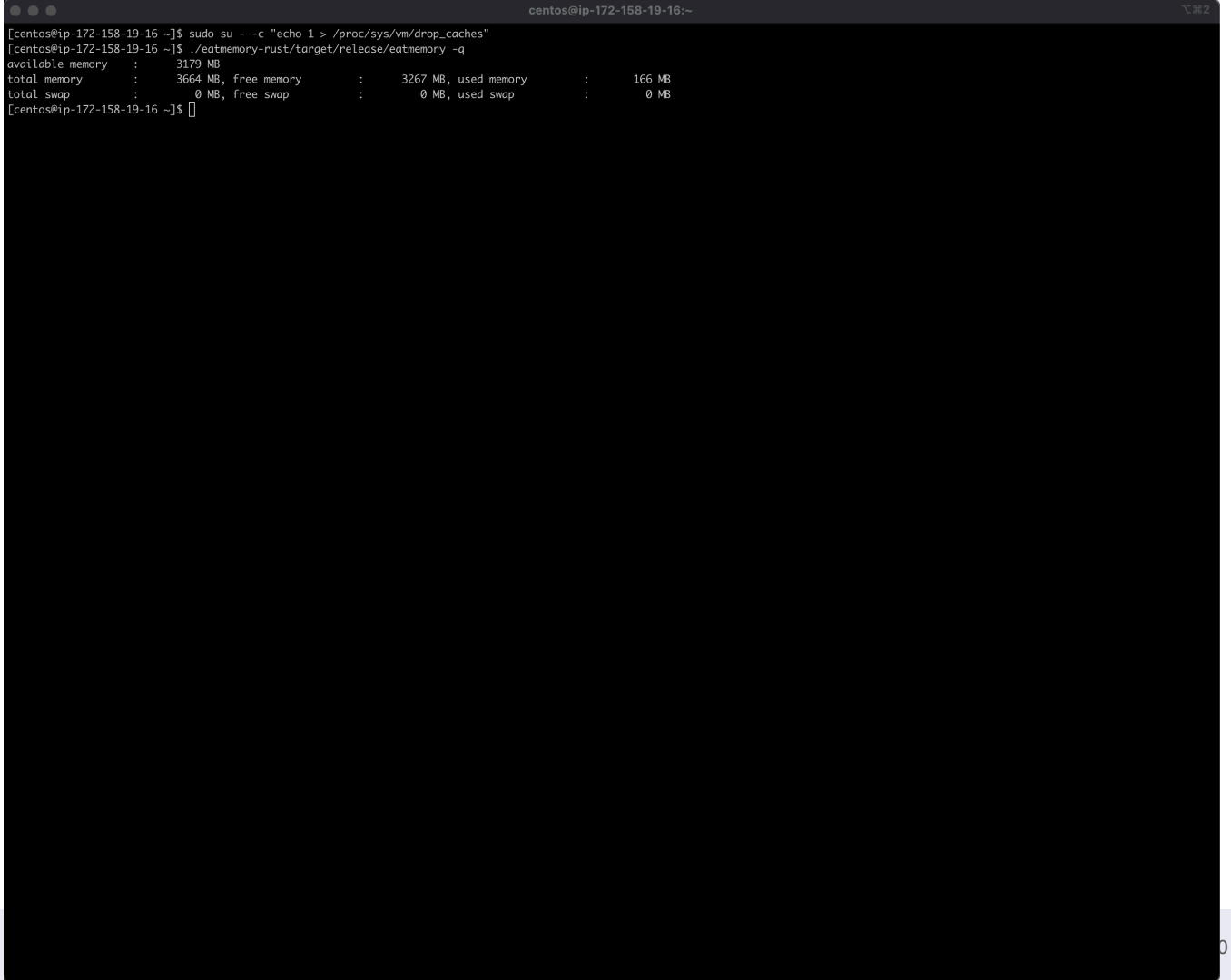
```
ios=0/0, merge=0/0, ticks=0/0, in_queue=0, util=0.00%
```

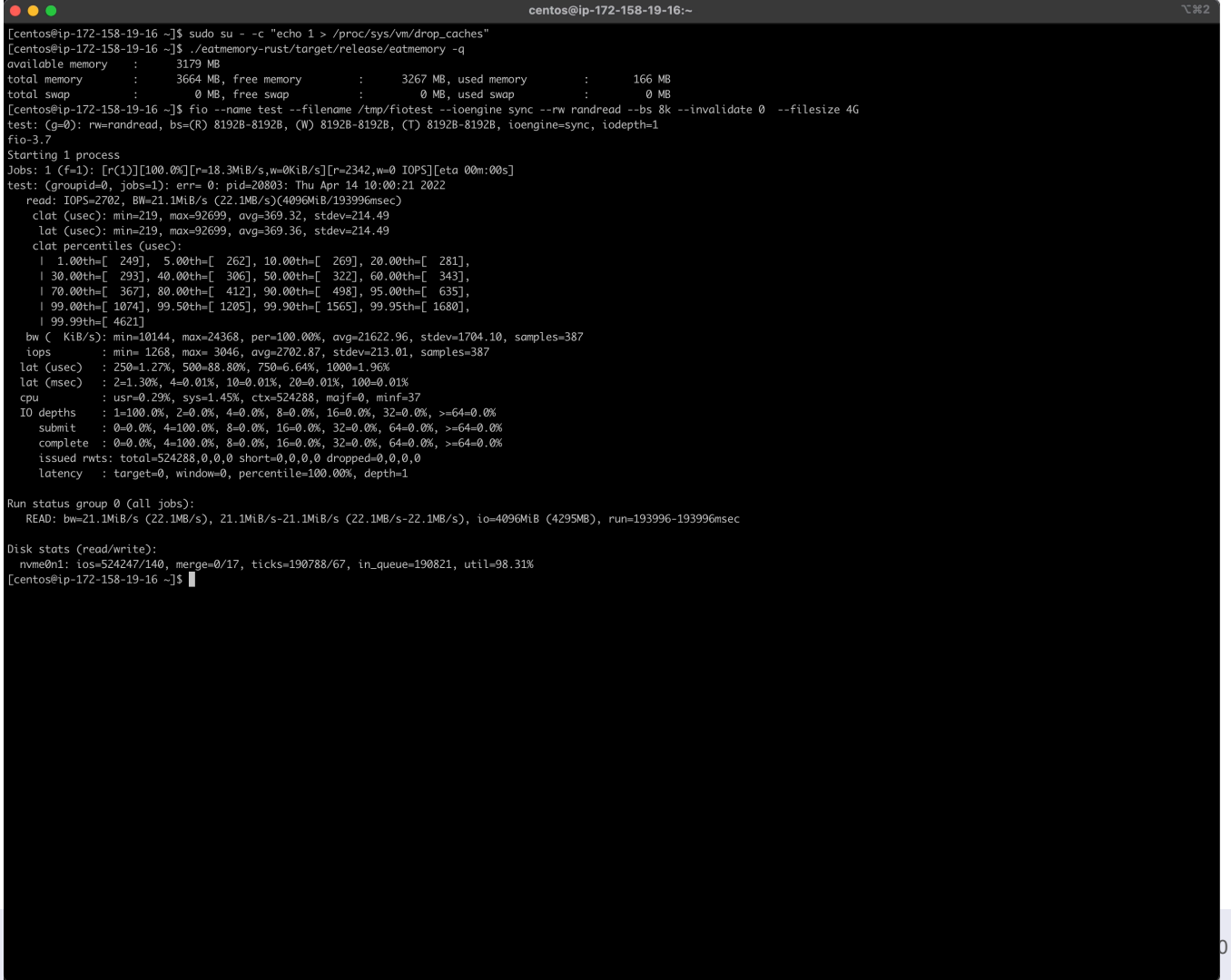
Read: 4G

- Drop the page cache
- Validate available memory
- Run a fio **read** test reading 4G:

```
fio --name test --filename /tmp/fiotest
    --ioengine sync --rw randread --bs 8k
    --invalidate 0 --filesize 2G
```







Read: 4G

- This is a summary from the run:

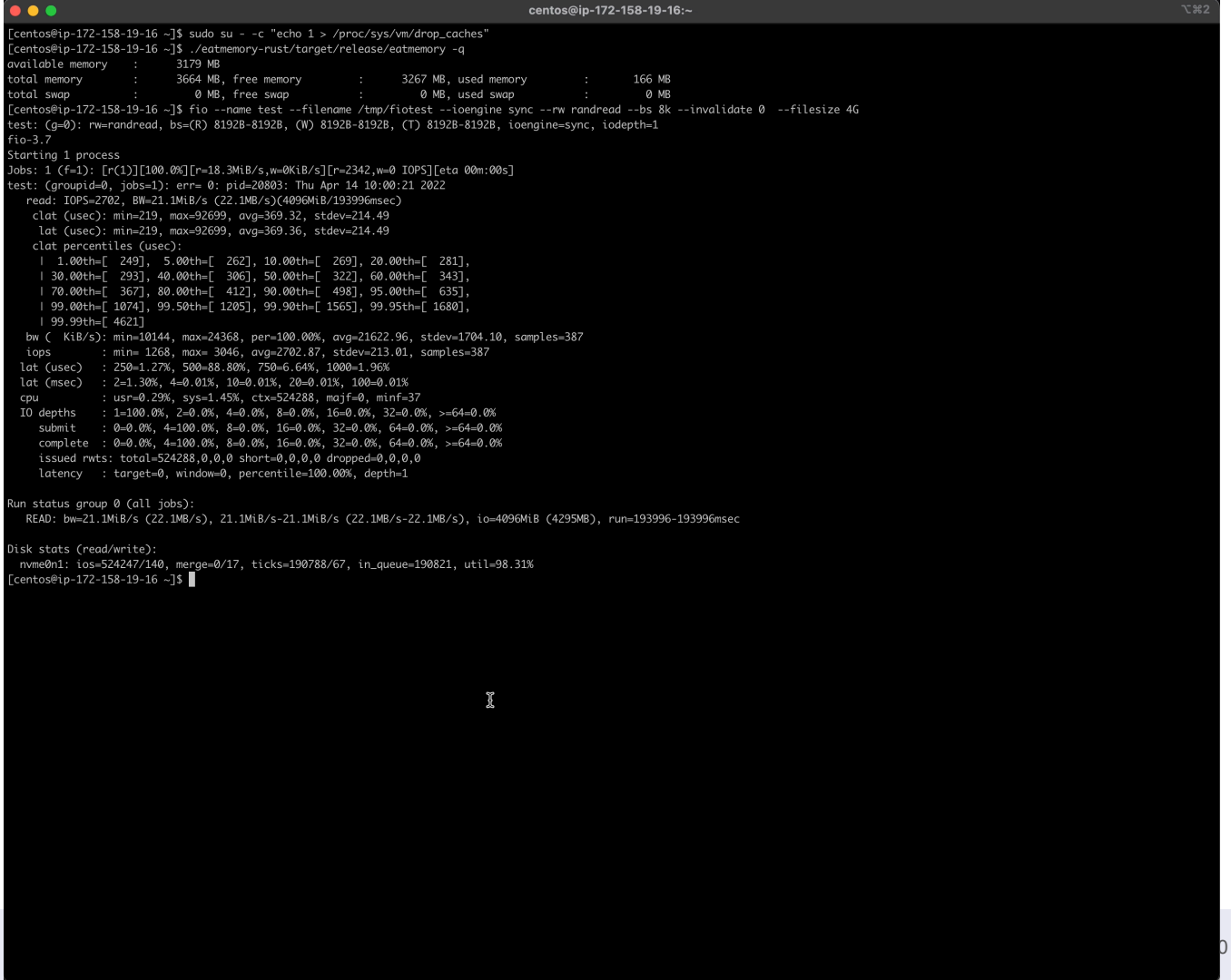
```
read: IOPS=2702, BW=21.1MiB/s (22.1MB/s)(4096MiB/193996msec)
```

- My limits are 125 MBPS and 3000 IOPS.
- IOPS rate identical to 2G run, indicates being latency bound again.
- Time and disk physical IOs roughly doubled, as expected.

Read: 4G

- Now lets perform the exact same run again

- Caveat: I had to slightly alter the fio statement.
  - Add option `--norandommap`
  - This prevents every 8k IO offset from being touched exactly once.



```
centos@ip-172-158-19-16:~                                                    ⌥⌘2
Starting 1 process
Jobs: 1 (f=1): [r(1)][100.0%][r=18.3MiB/s,w=0KiB/s][r=2342,w=0 IOPS][eta 00m:00s]
test: (groupid=0, jobs=1): err= 0: pid=20803: Thu Apr 14 10:00:21 2022
   read: IOPS=2702, BW=21.1MiB/s (22.1MB/s)(4096MiB/193996msec)
    clat (usec): min=219, max=92699, avg=369.32, stdev=214.49
     lat (usec): min=219, max=92699, avg=369.36, stdev=214.49
    clat percentiles (usec):
     |  1.00th=[  249],  5.00th=[  262], 10.00th=[  269], 20.00th=[  281],
     | 30.00th=[  293], 40.00th=[  306], 50.00th=[  322], 60.00th=[  343],
     | 70.00th=[  367], 80.00th=[  412], 90.00th=[  498], 95.00th=[  635],
     | 99.00th=[ 1074], 99.50th=[ 1205], 99.90th=[ 1565], 99.95th=[ 1680],
     | 99.99th=[ 4621]
   bw (  KiB/s): min=10144, max=24368, per=100.00%, avg=21622.96, stdev=1704.10, samples=387
   iops        : min= 1268, max= 3046, avg=2702.87, stdev=213.01, samples=387
  lat (usec)   : 250=1.27%, 500=88.80%, 750=6.64%, 1000=1.96%
  lat (msec)   : 2=1.30%, 4=0.01%, 10=0.01%, 20=0.01%, 100=0.01%
  cpu          : usr=0.29%, sys=1.45%, ctx=524288, majf=0, minf=37
  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 rwts: total=524288,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=21.1MiB/s (22.1MB/s), 21.1MiB/s-21.1MiB/s (22.1MB/s-22.1MB/s), io=4096MiB (4295MB), run=193996-193996msec

Disk stats (read/write):
  nvme0n1: ios=524247/140, merge=0/17, ticks=190788/67, in_queue=190821, util=98.31%
[centos@ip-172-158-19-16 ~]$ fio --name test --filename /tmp/fiotest --ioengine sync --rw randread --bs 8k --invalidate 0  --norandommap --filesize 4G
test: (g=0): rw=randread, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=sync, iodepth=1
fio-3.7
Starting 1 process
Jobs: 1 (f=1): [r(1)][99.2%][r=61.0MiB/s,w=0KiB/s][r=7935,w=0 IOPS][eta 00m:01s]
test: (groupid=0, jobs=1): err= 0: pid=22616: Thu Apr 14 10:09:11 2022
   read: IOPS=4206, BW=32.9MiB/s (34.5MB/s)(4096MiB/124633msec)
    clat (nsec): min=513, max=30504k, avg=237260.41, stdev=228596.70
     lat (nsec): min=534, max=30504k, avg=237293.03, stdev=228597.81
    clat percentiles (nsec):
     |  1.00th=[   1176],  5.00th=[   1256], 10.00th=[   1336],
     | 20.00th=[   1528], 30.00th=[   1848], 40.00th=[ 264192],
     | 50.00th=[ 284672], 60.00th=[ 305152], 70.00th=[ 329728],
     | 80.00th=[ 366592], 90.00th=[ 448512], 95.00th=[ 552960],
     | 99.00th=[ 970752], 99.50th=[1122304], 99.90th=[1499136],
     | 99.95th=[1613824], 99.99th=[2768896]
   bw (  KiB/s): min=17760, max=64128, per=99.90%, avg=33619.46, stdev=9944.64, samples=249
   iops        : min= 2220, max= 8016, avg=4202.43, stdev=1243.08, samples=249
  lat (nsec)   : 750=0.01%, 1000=0.01%
  lat (usec)   : 2=31.74%, 4=4.83%, 10=0.18%, 20=0.02%, 250=0.71%
  lat (usec)   : 500=55.67%, 750=4.56%, 1000=1.38%
  lat (msec)   : 2=0.88%, 4=0.01%, 10=0.01%, 20=0.01%, 50=0.01%
  cpu          : usr=0.28%, sys=1.73%, ctx=331382, majf=0, minf=36
  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 rwts: total=524288,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=32.9MiB/s (34.5MB/s), 32.9MiB/s-32.9MiB/s (34.5MB/s-34.5MB/s), io=4096MiB (4295MB), run=124633-124633msec

Disk stats (read/write):
  nvme0n1: ios=330920/105, merge=0/14, ticks=122124/59, in_queue=122162, util=98.04%
[centos@ip-172-158-19-16 ~]$
```

Read: 4G

- This is a summary from the run:

```
read: IOPS=4206, BW=32.9MiB/s (34.5MB/s)(4096MiB/124633msec)
```

- My limits are 125 MBPS and 3000 IOPS.
- IOPS rate increased, because of caching
- Still had to do a lot of IO:

```
ios=330920/105
issued rwts: total=524288,0
```

Reality

- Let's take a look at the memory figures again:

```
[centos@ip-172-158-19-16 ~]$ ./eatmemory-rust/target/release/eatmemory -q
available memory     :      3179 MB
total memory         :      3664 MB, free memory         :      3267 MB, used memory         :       166 MB
total swap           :         0 MB, free swap           :         0 MB, used swap           :         0 MB
```

- Having 166MB used is not a realistic scenario.
- A server would typically have an application running!
- Which is what reads that data to serve it, right?
- What if we occupy 50% of memory?

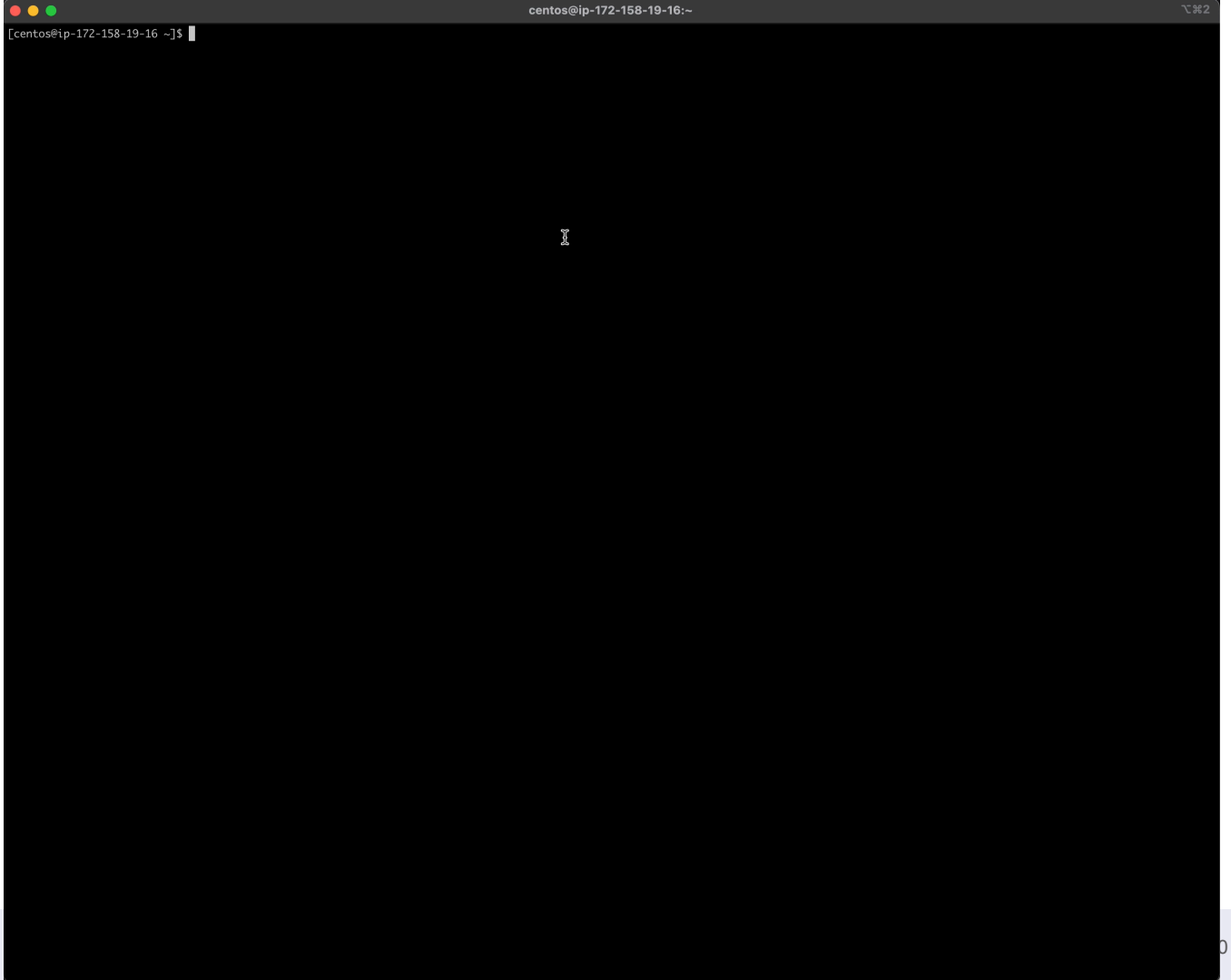
eatmemory

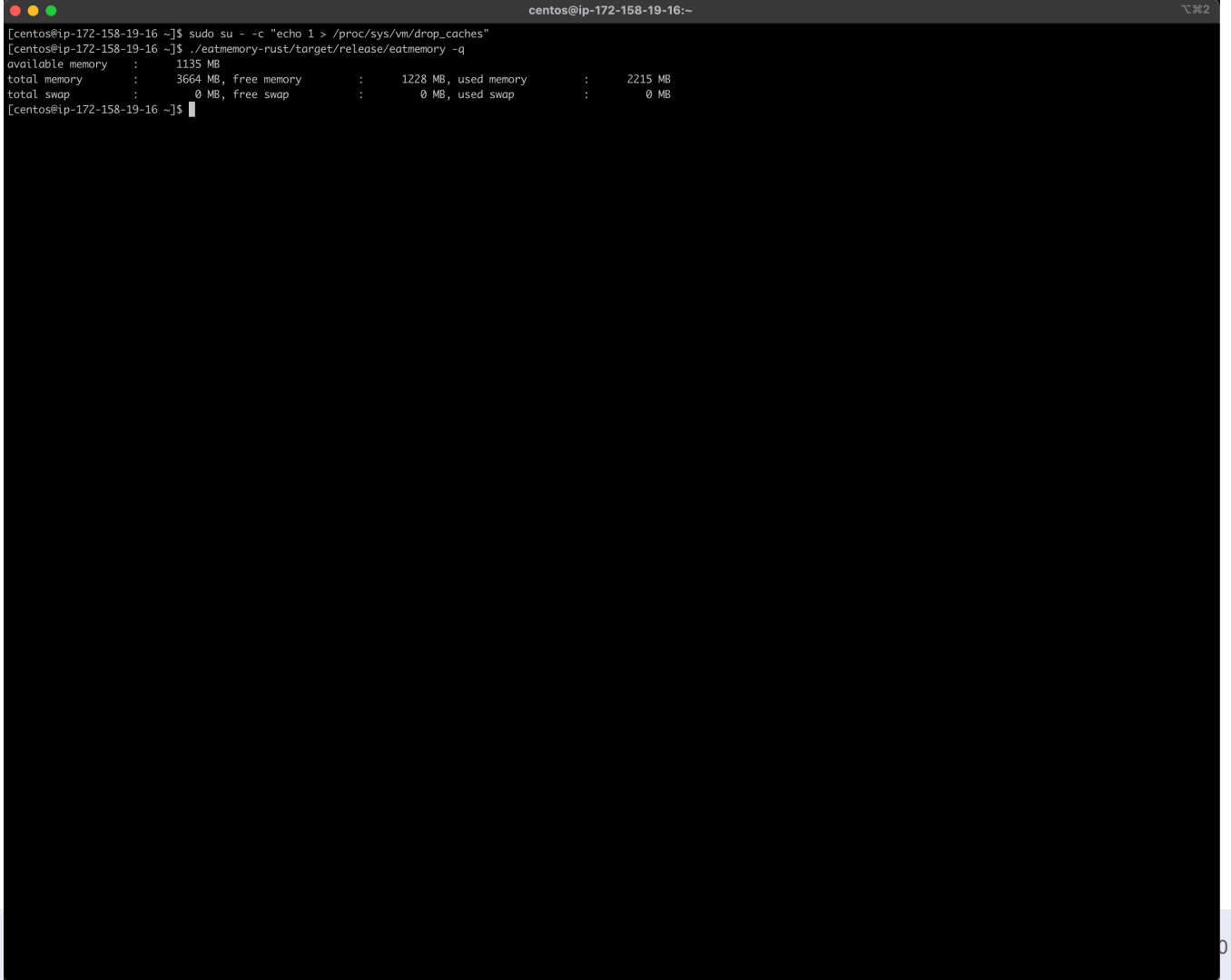
- I build a tool that can do that: eatmemory

```
[centos@ip-172-158-19-16 ~]$ ./eatmemory-rust/target/release/eatmemory -s 2000
done. press enter to stop and deallocate
```

- Credits to original eatmemory.c tool (https://github.com/julman99/eatmemory.git)

- Let's try the same 2G run again!





```
[centos@ip-172-158-19-16 ~]$ sudo su - -c "echo 1 > /proc/sys/vm/drop_caches"
[centos@ip-172-158-19-16 ~]$ ./eatmemory-rust/target/release/eatmemory -q
available memory    :      1135 MB
total memory        :      3664 MB, free memory         :      1228 MB, used memory         :      2215 MB
total swap          :         0 MB, free swap           :         0 MB, used swap           :         0 MB
[centos@ip-172-158-19-16 ~]$ fio --name test --filename /tmp/fiotest --ioengine sync --rw randread --bs 8k --invalidate 0  --filesize 2G
test: (g=0): rw=randread, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=sync, iodepth=1
fio-3.7
Starting 1 process
Jobs: 1 (f=1): [r(1)][100.0%][r=21.2MiB/s,w=0KiB/s][r=2719,w=0 IOPS][eta 00m:00s]
test: (groupid=0, jobs=1): err= 0: pid=26975: Thu Apr 14 10:36:21 2022
   read: IOPS=2671, BW=20.9MiB/s (21.9MB/s)(2048MiB/98136msec)
    clat (usec): min=173, max=24206, avg=373.64, stdev=218.70
     lat (usec): min=173, max=24207, avg=373.69, stdev=218.70
    clat percentiles (usec):
     |  1.00th=[  251],  5.00th=[  265], 10.00th=[  273], 20.00th=[  285],
     | 30.00th=[  297], 40.00th=[  310], 50.00th=[  326], 60.00th=[  347],
     | 70.00th=[  371], 80.00th=[  412], 90.00th=[  498], 95.00th=[  644],
     | 99.00th=[ 1074], 99.50th=[ 1221], 99.90th=[ 1614], 99.95th=[ 2409],
     | 99.99th=[10290]
   bw (  KiB/s): min=13392, max=23744, per=99.99%, avg=21365.87, stdev=1330.25, samples=196
   iops        : min= 1674, max= 2968, avg=2670.73, stdev=166.30, samples=196
  lat (usec)   : 250=0.87%, 500=89.28%, 750=6.43%, 1000=2.02%
  lat (msec)   : 2=1.34%, 4=0.03%, 10=0.02%, 20=0.01%, 50=0.01%
  cpu          : usr=0.31%, sys=1.49%, ctx=262145, majf=0, minf=36
  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 rwts: total=262144,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=20.9MiB/s (21.9MB/s), 20.9MiB/s-20.9MiB/s (21.9MB/s-21.9MB/s), io=2048MiB (2147MB), run=98136-98136msec

Disk stats (read/write):
  nvme0n1: ios=262560/146, merge=0/17, ticks=96004/105, in_queue=96092, util=97.37%
[centos@ip-172-158-19-16 ~]$
```

Read: 2G / 50% of 4G memory taken

- This is a summary from the run:

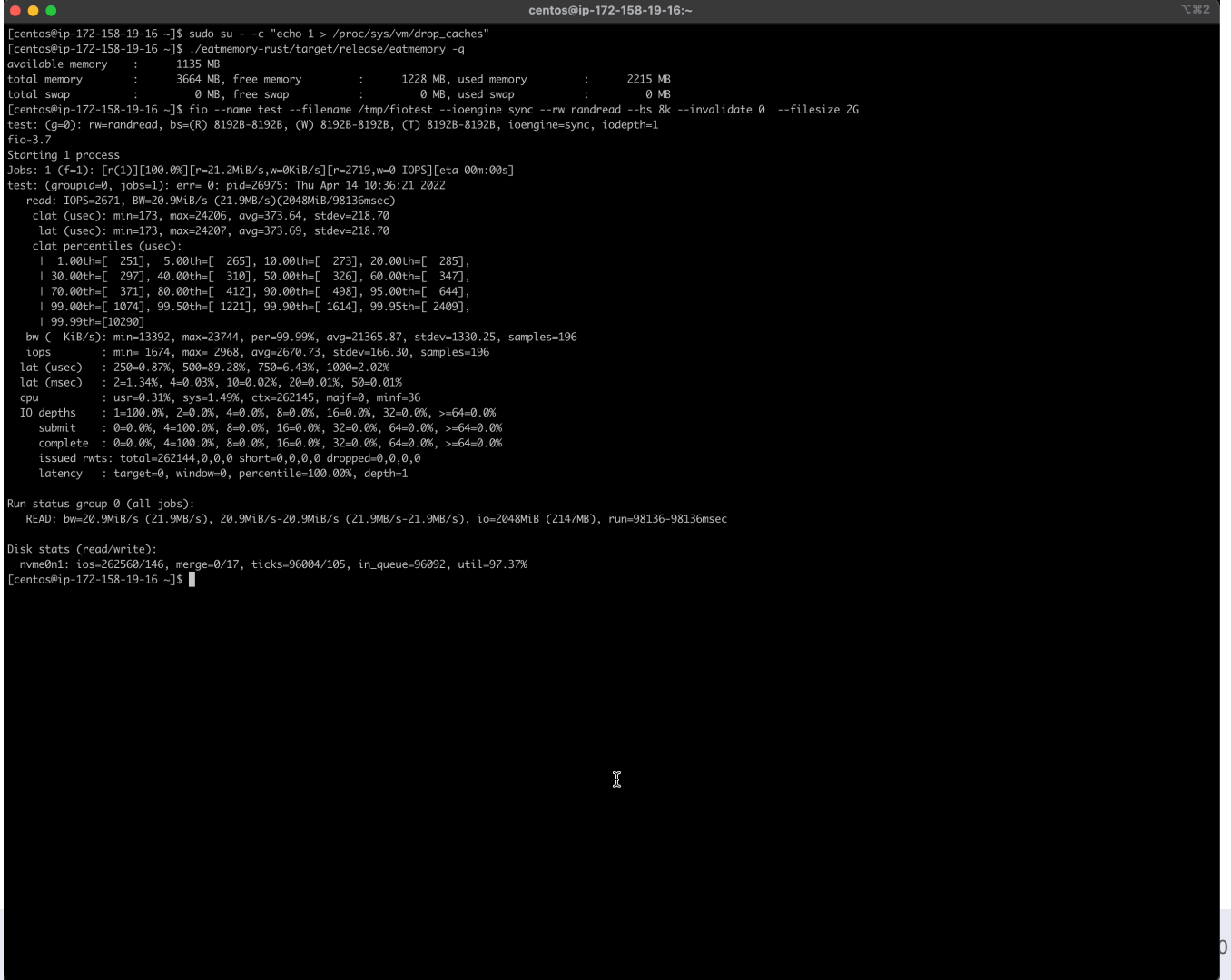
```
read: IOPS=2671, BW=20.9MiB/s (21.9MB/s)(2048MiB/98136msec)
```

- My limits are 125 MBPS and 3000 IOPS.

- Time is slightly less (98136 vs. 100465), but generally equal.
- Because despite the memory allocation, there was no change: bound by IO.

Read: 2G / 50% of 4G memory taken

- Now lets perform the same run again
  - Add option `--norandommap`



```
centos@ip-172-158-19-16:~                                                    ⌥⌘2
Starting 1 process
Jobs: 1 (f=1): [r(1)][100.0%][r=21.2MiB/s,w=0KiB/s][r=2719,w=0 IOPS][eta 00m:00s]
test: (groupid=0, jobs=1): err= 0: pid=26975: Thu Apr 14 10:36:21 2022
   read: IOPS=2671, BW=20.9MiB/s (21.9MB/s)(2048MiB/98136msec)
    clat (usec): min=173, max=24206, avg=373.64, stdev=218.70
     lat (usec): min=173, max=24207, avg=373.69, stdev=218.70
    clat percentiles (usec):
     |  1.00th=[  251],  5.00th=[  265], 10.00th=[  273], 20.00th=[  285],
     | 30.00th=[  297], 40.00th=[  310], 50.00th=[  326], 60.00th=[  347],
     | 70.00th=[  371], 80.00th=[  412], 90.00th=[  498], 95.00th=[  644],
     | 99.00th=[ 1074], 99.50th=[ 1221], 99.90th=[ 1614], 99.95th=[ 2409],
     | 99.99th=[10290]
   bw (  KiB/s): min=13392, max=23744, per=99.99%, avg=21365.87, stdev=1330.25, samples=196
   iops        : min= 1674, max= 2968, avg=2670.73, stdev=166.30, samples=196
  lat (usec)   : 250=0.87%, 500=89.28%, 750=6.43%, 1000=2.02%
  lat (msec)   : 2=1.34%, 4=0.03%, 10=0.02%, 20=0.01%, 50=0.01%
  cpu          : usr=0.31%, sys=1.49%, ctx=262145, majf=0, minf=36
  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 rwts: total=262144,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=20.9MiB/s (21.9MB/s), 20.9MiB/s-20.9MiB/s (21.9MB/s-21.9MB/s), io=2048MiB (2147MB), run=98136-98136msec

Disk stats (read/write):
  nvme0n1: ios=262560/146, merge=0/17, ticks=96004/105, in_queue=96092, util=97.37%
[centos@ip-172-158-19-16 ~]$ fio --name test --filename /tmp/fiotest --ioengine sync --rw randread --bs 8k --invalidate 0  --norandommap --filesize 2G
test: (g=0): rw=randread, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=sync, iodepth=1
fio-3.7
Starting 1 process
Jobs: 1 (f=1): [r(1)][100.0%][r=31.2MiB/s,w=0KiB/s][r=3988,w=0 IOPS][eta 00m:00s]
test: (groupid=0, jobs=1): err= 0: pid=32758: Thu Apr 14 11:10:09 2022
   read: IOPS=3525, BW=27.5MiB/s (28.9MB/s)(2048MiB/74366msec)
    clat (nsec): min=516, max=28349k, avg=283194.64, stdev=256978.87
     lat (nsec): min=538, max=28349k, avg=283232.46, stdev=256979.92
    clat percentiles (nsec):
     |  1.00th=[   1208],  5.00th=[   1320], 10.00th=[   1448],
     | 20.00th=[   1768], 30.00th=[ 261120], 40.00th=[ 284672],
     | 50.00th=[ 301056], 60.00th=[ 321536], 70.00th=[ 350208],
     | 80.00th=[ 395264], 90.00th=[ 485376], 95.00th=[ 618496],
     | 99.00th=[1056768], 99.50th=[1204224], 99.90th=[1581056],
     | 99.95th=[1728512], 99.99th=[5079040]
   bw (  KiB/s): min=11744, max=34736, per=99.96%, avg=28190.05, stdev=4400.47, samples=148
   iops        : min= 1468, max= 4342, avg=3523.75, stdev=550.06, samples=148
  lat (nsec)   : 750=0.01%, 1000=0.01%
  lat (usec)   : 2=22.80%, 4=3.41%, 10=0.31%, 20=0.02%, 250=0.99%
  lat (usec)   : 500=63.35%, 750=6.05%, 1000=1.79%
  lat (msec)   : 2=1.23%, 4=0.02%, 10=0.01%, 20=0.01%, 50=0.01%
  cpu          : usr=0.27%, sys=1.63%, ctx=192535, majf=0, minf=36
  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 rwts: total=262144,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=27.5MiB/s (28.9MB/s), 27.5MiB/s-27.5MiB/s (28.9MB/s-28.9MB/s), io=2048MiB (2147MB), run=74366-74366msec

Disk stats (read/write):
  nvme0n1: ios=192264/64, merge=0/6, ticks=72937/55, in_queue=72978, util=98.01%
[centos@ip-172-158-19-16 ~]$
```

Read: 2G / 50% of 4G memory taken

```
read: IOPS=3525, BW=27.5MiB/s (28.9MB/s)(2048MiB/74366msec)
```

- My limits are 125 MBPS and 3000 IOPS.
- This gone beyond the limits (IOPS, 3000 <> 3525).
- Time difference with previous 2nd 2G run: 74.3 <> 0.4 second (!)

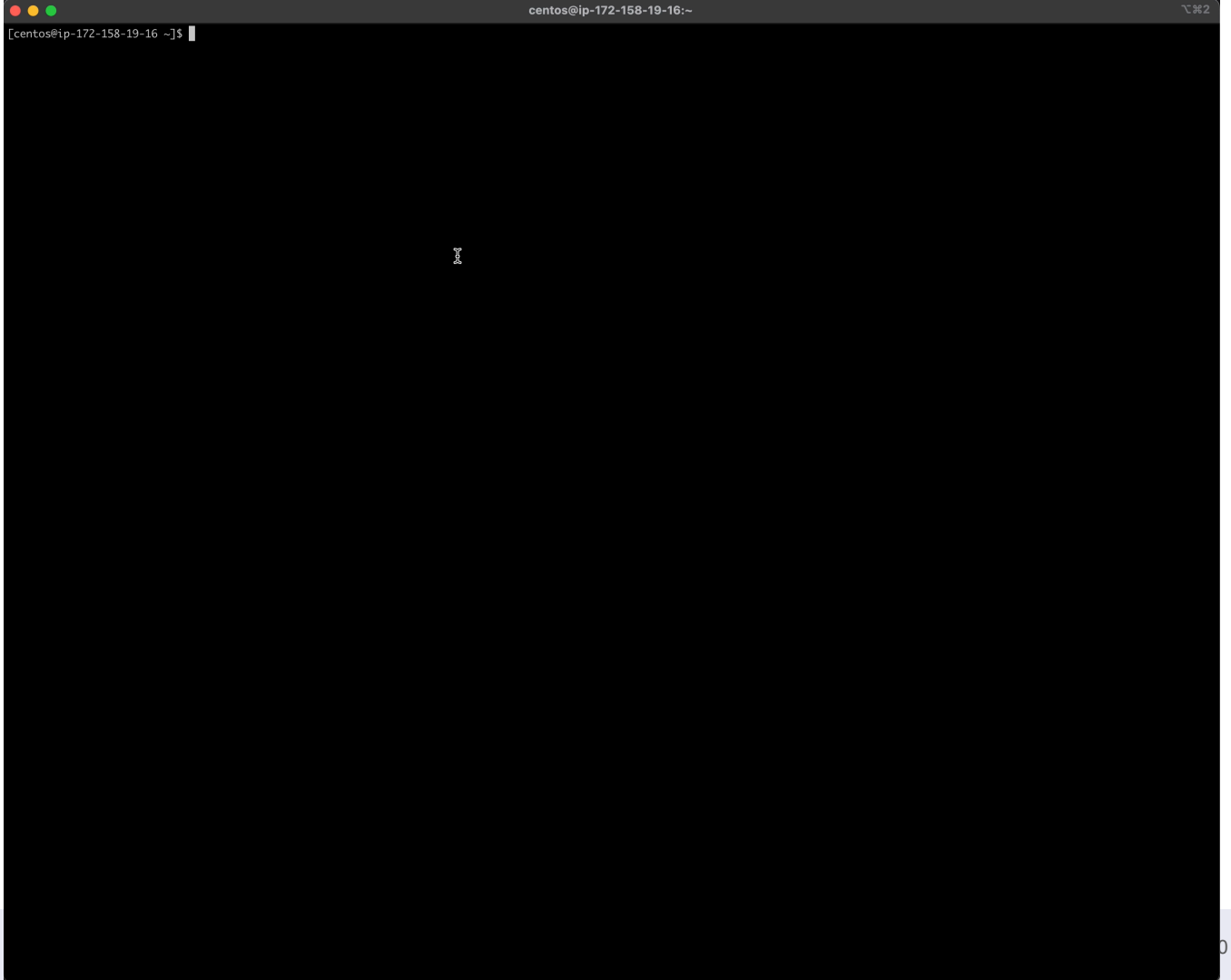
- Reason: physical IO had to be performed:

```
ios=192264/64, merge=0/6, ticks=72937/55, in_queue=72978, util=98.01%
```

Write: 2G

- Validate available memory
- Run a fio **write** test writing 2G:

```
fio --name test --filename /tmp/fiotest
    --ioengine sync --rw randwrite --bs 8k
    --filesize 2G
```



Write: 2G

- This is a summary from the run:

```
IOPS=22.1k, BW=173MiB/s (181MB/s)(2048MiB/11840msec)
clat (usec): min=2, max=18861, avg=44.34, stdev=597.77
```

- My limits are 125 MBPS and 3000 IOPS.
- IOPS = 22100, which is significantly more than 3000 IOPS.
- Reason: only 24% was written;

```
ios=0/61815
issued rwts: total=0,262144
```

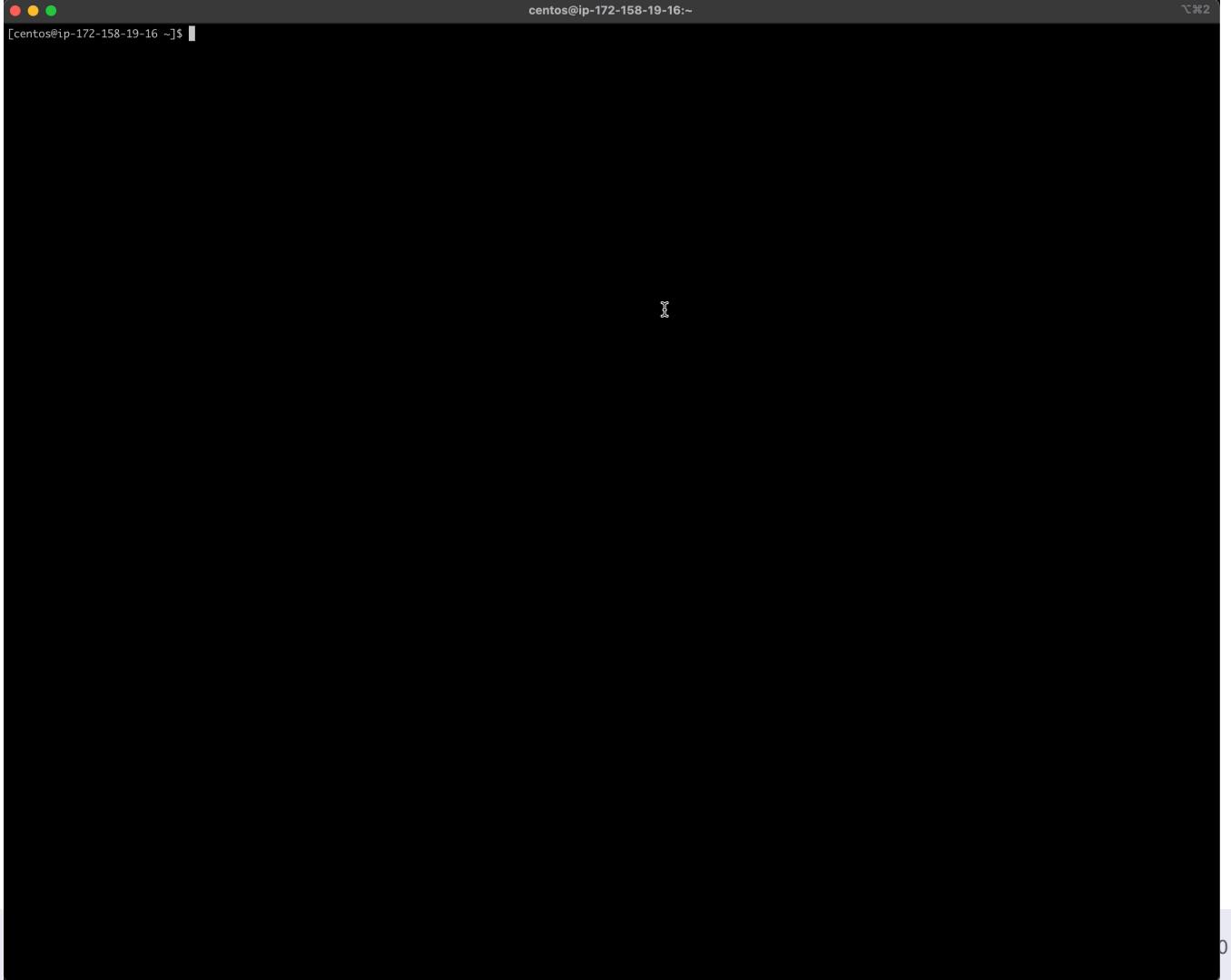
Write: 2G -- write details

- Why aren't all writes cached, like all reads were?
  - Writes are special!
  - Writes cannot be discarded like reads can, they **must** be written first.
  - Writes can/should not exhaust available memory.
  - Therefore: `vm.dirty_background_ratio`, `vm.dirty_ratio`, others.
  - Ratio is taken from *available memory*, unlike popular believe of total mem.
  - https://dev.to/fritshooglandyugabyte/linux-buffered-write-latency-10mc
- In linux, processes performing buffered writes do not actually write to disk.
  - Produce dirty pages, and get throttled (wait in write()) to balance.

Write: 500M

- Validate available memory
- Run a fio **write** test writing 500M:

```
fio --name test --filename /tmp/fiotest
    --ioengine sync --rw randwrite --bs 8k
    --filesize 500M
```



Write: 500M

- This is a summary from the run:

```
IOPS=193k, BW=1506MiB/s (1579MB/s)(500MiB/332msec)
```

- My limits are 125 MBPS and 3000 IOPS.
- IOPS = 193000, MBPS = 1506.
- Reason; no write (throttling):

```
ios=0/0, merge=0/0, ticks=0/0, in_queue=0, util=0.00%
```

- Why? Available: 3072 MB, `vm.dirty_ratio`: 30% = 922MB

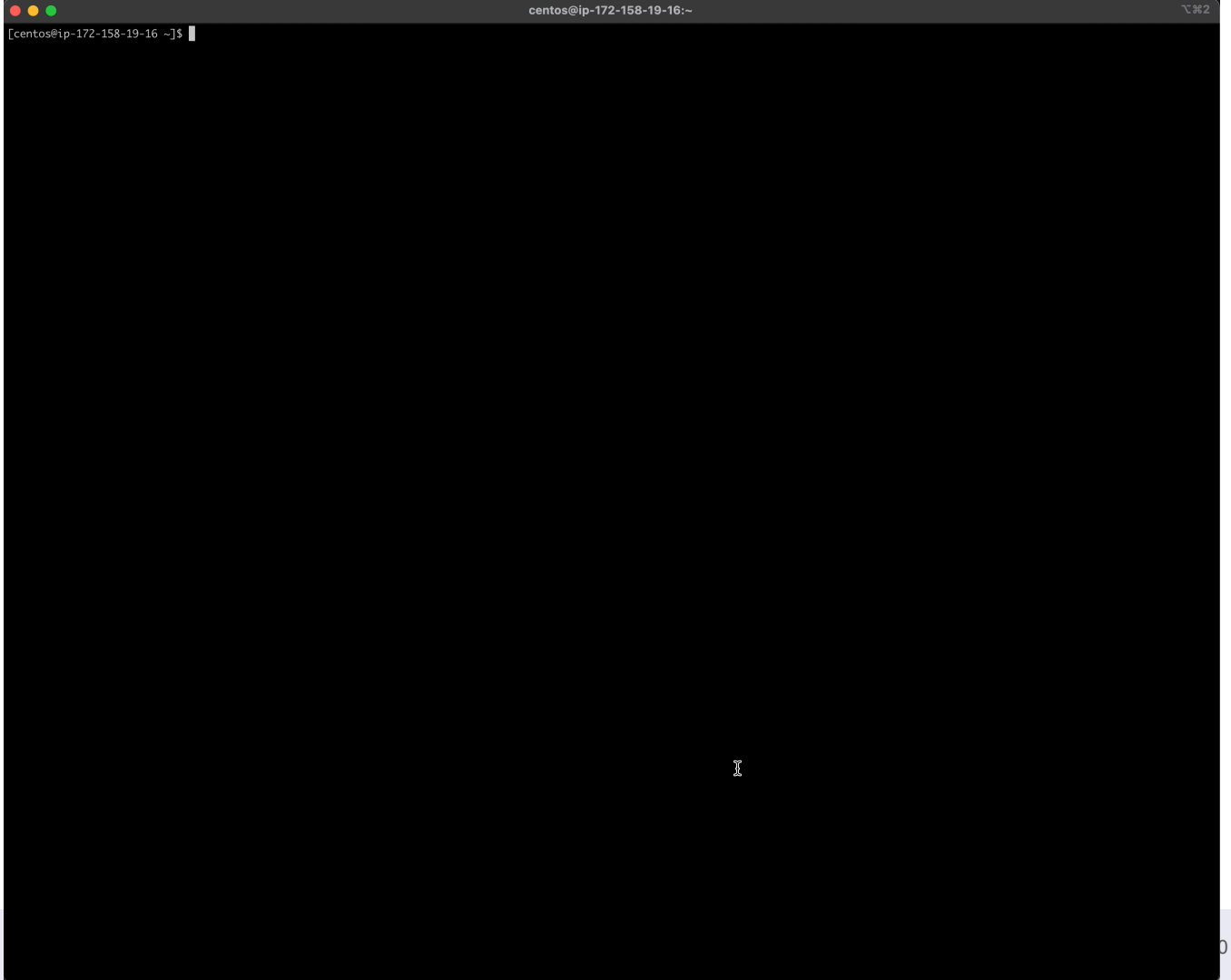
Reality

- The writes so far were also conducted with no memory used.
- Let's occupy 50% and perform the same tests again.

Write: 2G / 50% of 4G memory taken

- Validate available memory
- Run a fio **write** test writing 2G:

```
fio --name test --filename /tmp/fiotest
    --ioengine sync --rw randwrite --bs 8k
    --filesize 2G
```



```
centos@ip-172-158-19-16:~
[centos@ip-172-158-19-16 ~]$ ./eatmemory-rust/target/release/eatmemory -q
available memory    :      1018 MB
total memory        :      3664 MB, free memory         :       134 MB, used memory         :      2217 MB
total swap          :         0 MB, free swap           :         0 MB, used swap           :         0 MB
[centos@ip-172-158-19-16 ~]$ fio --name test --filename /tmp/fiotest --ioengine sync --rw randwrite --bs 8k --filesize 2G
test: (g=0): rw=randwrite, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=sync, iodepth=1
fio-3.7
Starting 1 process
Jobs: 1 (f=1): [w(1)][100.0%][r=0KiB/s,w=43.1MiB/s][r=0,w=5516 IOPS][eta 00m:00s]
test: (groupid=0, jobs=1): err= 0: pid=14088: Thu Apr 14 12:22:25 2022
  write: IOPS=6581, BW=51.4MiB/s (53.9MB/s)(2048MiB/39832msec)
    clat (usec): min=2, max=15892, avg=151.04, stdev=920.26
     lat (usec): min=2, max=15892, avg=151.09, stdev=920.26
    clat percentiles (usec):
     |  1.00th=[    4],  5.00th=[    4], 10.00th=[    4], 20.00th=[    4],
     | 30.00th=[    4], 40.00th=[    4], 50.00th=[    4], 60.00th=[    5],
     | 70.00th=[    5], 80.00th=[    5], 90.00th=[    6], 95.00th=[    8],
     | 99.00th=[ 5866], 99.50th=[ 5866], 99.90th=[ 5866], 99.95th=[ 5866],
     | 99.99th=[ 9765]
   bw (  KiB/s): min=31872, max=451584, per=100.00%, avg=52686.61, stdev=52446.86, samples=79
   iops        : min= 3984, max=56448, avg=6585.82, stdev=6555.86, samples=79
  lat (usec)   : 4=52.09%, 10=45.08%, 20=0.33%, 50=0.01%, 250=0.01%
  lat (usec)   : 1000=0.01%
  lat (msec)   : 10=2.49%, 20=0.01%
  cpu          : usr=0.28%, sys=3.17%, ctx=6580, majf=0, minf=33
  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 rwts: total=0,262144,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
  WRITE: bw=51.4MiB/s (53.9MB/s), 51.4MiB/s-51.4MiB/s (53.9MB/s-53.9MB/s), io=2048MiB (2147MB), run=39832-39832msec

Disk stats (read/write):
  nvme0n1: ios=216/151456, merge=0/37180, ticks=2139/1488942, in_queue=1491142, util=99.45%
[centos@ip-172-158-19-16 ~]$
```

Write: 2G / 50% of 4G memory taken

- This is a summary from the run:

```
IOPS=6581, BW=51.4MiB/s (53.9MB/s)(2048MiB/39832msec)
```

- My limits are 125 MBPS and 3000 IOPS.
- IOPS = 6581, MBPS = 51 (vs . 22100 IOPS, 173 MBPS no mem pressure)
- Reason; write throttling:

```
ios=216/151456
```

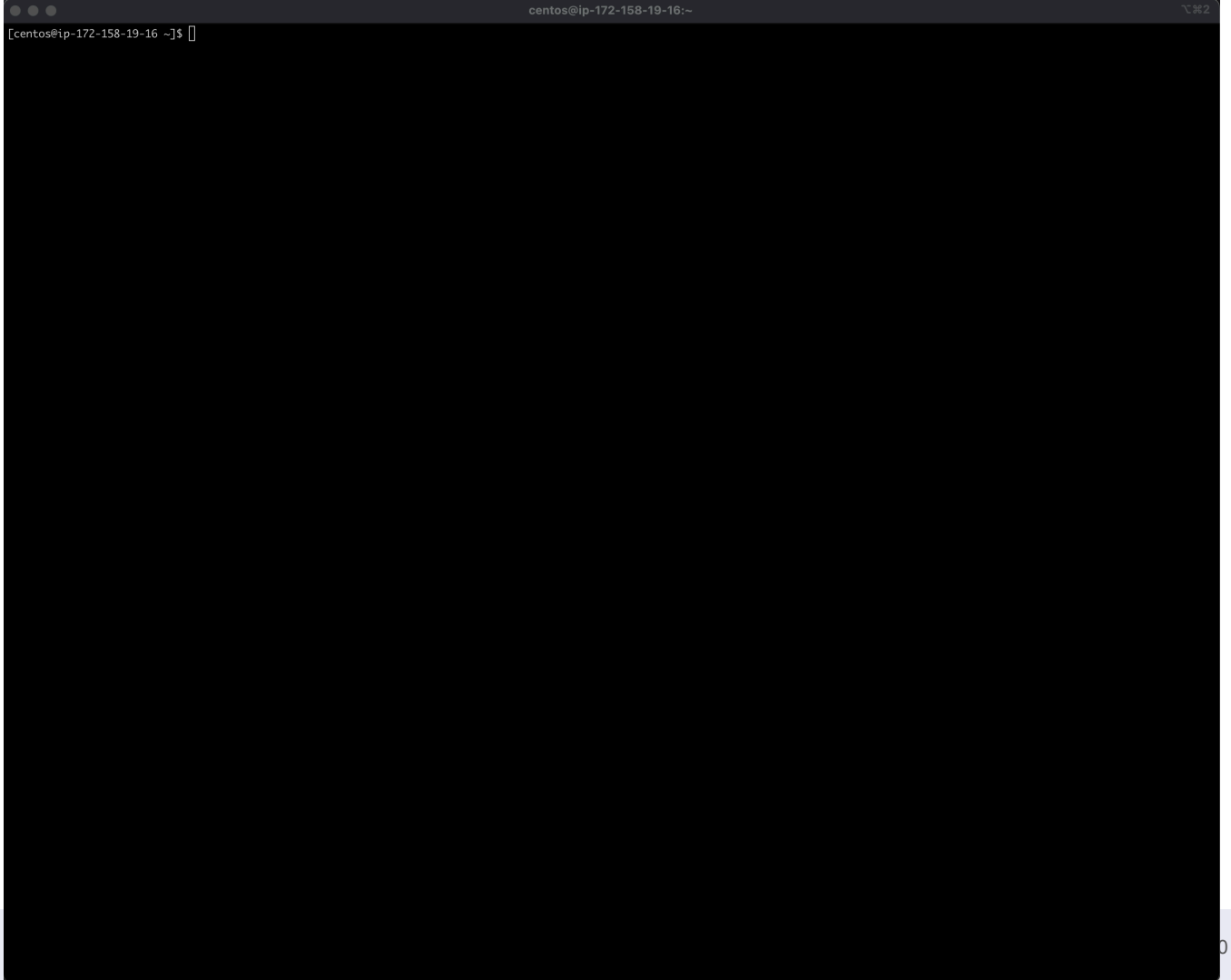
- Why? Available: 1018 MB, `vm.dirty_ratio`: 30% = 305MB

Write: 500M / 50% of 4G memory taken

- How about writing 500M? That was really fast previously?

- Run a fio **write** test writing 500M:

```
fio --name test --filename /tmp/fiotest
    --ioengine sync --rw randwrite --bs 8k
    --filesize 500M
```



Write: 500M / 50% of 4G memory taken

- This is a summary from the run:

```
IOPS=25.1k, BW=196MiB/s (206MB/s)(500MiB/2549msec) (332ms)
```

- My limits are 125 MBPS and 3000 IOPS.
- IOPS = 25100, MBPS = 196 (vs . 193000 IOPS, 1506 MBPS no mem pressure)
- *Despite feeling fast, performance was severely impacted!!*
- Reason; write throttling:

```
ios=17/20846
```

- Why? Available: 1023 MB, `vm.dirty_ratio`: 30% = 307MB

Conclusion

- If you are using buffered IO, do you rely on caching for performance?
- Are you keeping track of Available Memory?
- Available memory ≈ memory *acting as/available for* cache + regular alloc.

- Understand the differences between read and write cache properties:
  - Data must be read before it can be cached and reused.
  - A variable proportional limit is imposed on # dirty buffers.
    - Kernel applies write throttling when # dirty pages increases.

Conclusion

- You have to understand your *active dataset*
  - Which consists of reads and writes.
  - The cache effectivity is relative to available memory.

- This means you might seem to suffer random IO performance issues.
  - Which can be caused by either:
    - Change in the active dataset.
    - Change in available memory

PS

- The tests were performed on linux without swap.
  - Buffered IO needs buffers which is a memory allocation.
  - It competes with regular IO allocations.

- Therefore, linux will evaluate available memory using a LRU mechanism.
  - Therefore, bursts of IO buffers usage
  - Could push seldom used mapped allocations to swap.