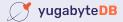
# Exploring Linux Memory Usage and Disk IO performance

Frits Hoogland Yugabyte



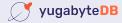
## **Topic: disk IO and disk IO performance**

- 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.



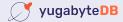
## **Disk IO and memory**

- 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.



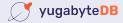
## Where does buffered IO go?

- 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.



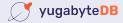
## Okay: but where does buffered IO go?

- 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.
  - <u>https://dev.to/yugabyte/what-is-free-memory-in-linux-18km</u>

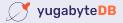
#### MemAvailable

- 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!

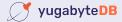
- Tests done on Amazon EC2:
  - c5.large VM (2000/4000 IOPS, 594/82 MBPS)
  - EBS: GP3 250M (3000 IOPS, 125 MBPS)
- I am not running into my bursting limits so concrete:
  - IOPS: 3000
  - MBPS: 125
- EC2 VM limits page:<u>https://docs.aws.amazon.com/AWSEC2/latest/</u> <u>UserGuide/ebs-optimized.html</u>
  - Not easy to find.

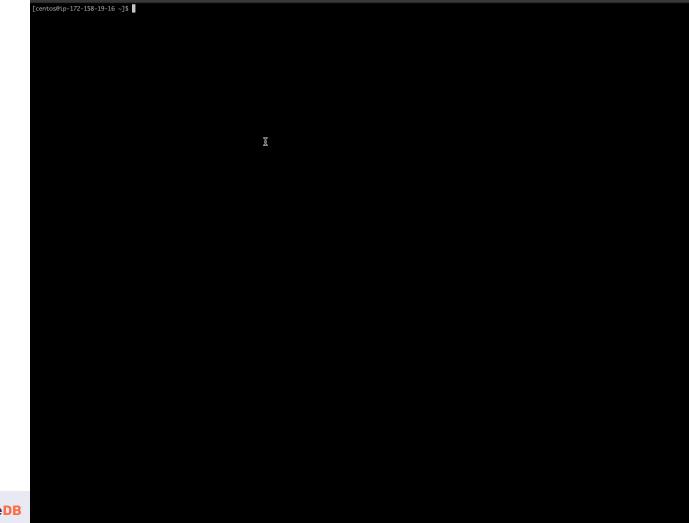


## Read: 2G

- 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
```



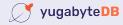


centos@ip-172-158-19-16:~

yugabyteDB

000

0 • •				
[centos@ip-172-158-19-16 ~]\$ sudo suc "echo 1 > /proc/sys/vm/drop_cod [centos@ip-172-158-19-16 ~]\$ ./etmemory-rust/target/release/eatmemory -a available memory = 3290 MB				
total memory : 3664 MB, free memory : 3376 MB, u	used memory : 161 MB			
total swap : 0 MB, free swap : 0 MB, u	used swap : 0 MB			
[centos@ip-172-158-19-16 ~]\$ [				



0	centos@ip-172-158-19-16:~	
<pre>[centos@ip-172-158-19-16 ~]\$ s [centos@ip-172-158-19-16 ~]\$ , available memory : 326 total swap : [centos@ip-172-158-19-16 ~]\$ f test: (g=0): rw=randread, bs=( fio-3.7 Starting 1 process Jobs: 1 (f=1): [r(1)][100.9%][ test: (groupid=0, jobs=1): err read: IOPS=Z609, Bw=20.4MtB clat (usec): min=195, max= lat (usec): min=195,</pre>	<pre>udo suc "echo 1 &gt; /proc/sys/vm/drop_caches" /eatmemory-rust/target/release/eatmemory -q 0 MB 4 MB, free memory : 3376 MB, used memory : 161 MB 0 MB, free swap : 0 MB, used swap : 0 MB ioname testfilename /tmp/fiotestioengine syncrw randreadbs 8kinvalidate 0filesize 26 R) 81928-81928, (W) 81928-81928, (T) 81928-81928, ioengine=sync, iodepth=1 r=21.1MiB/s,w=0KiB/s][r=2703,w=0 IOPS][eta 00m:00s] = 0: pid=725: Wed Apr 13 19:01:52 2022 /r385, avg=382.22, stdev=263.41 Z7385, avg=382.22, stdev=263.41 h=[ 269], 10.00th=[ 331], h=[ 318], 50.00th=[ 331], h=[ 420], 90.00th=[ 331], h=[ 420], 90.00th=[ 331], h=[ 420], 90.00th=[ 331], h=[ 420], 90.00th=[ 301], 95.00th=[ 652],</pre>	
70.00th=[ 1373], 80.00t   99.00th=[ 1106], 99.50t   99.99th=[1106], 99.50t   99.99th=[11076] bw ( KiB/s): min=10640, mo logs : min=1330, mo lat (usec) : 250=0.59%, 50 lat (msec) : 251-0.45%, 59 lat (msec) : 251-0.45%, 59 lot (msec) : 251-0.45%, 59 lot exput : us=0-0.45%, 45100 complet : 0-0.0%, 4-100		
<pre>latency : target=0, win Run status group 0 (all jobs):     READ: bw=20.4MiB/s (21.4MB/ Disk stats (read/write):     rvme0n1: ios=Z62238/113, mer</pre>	dow=0, percentile=100.00%, depth=1 s), 20.4MiB/s-20.4MiB/s (21.4MB/s-21.4MB/s), io=2048MiB (2147MB), run=100465-100465msec ge=0/11, ticks=97587/93, in_queue=97655, util=96.99%	
[centos@ip-172-158-19-16 ~]\$		



• 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:

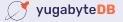
clat (usec): min=195, max=27385, avg=382.22, stdev=263.41

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



## Read: 2G

• Now lets perform the exact same run again



0 • •	centos@ip-172-158-19-16:~
[centos@ip-172-158-19-16 ~]\$ s: [centos@ip-172-158-19-16 ~]\$ available memory : 366 total swap : 366 total swap : ( [centos@ip-172-158-19-16 ~]\$ fi test: (g=0): rw=randread, bs=(fi fio=3.7 Starting 1 process Jobs: 1 (f=1): [r(1)][100.0%][r test: (groupid=0, jobs=1): errs.	do suc "echo 1 > /proc/sys/vm/drop_caches" athmemory-rust/target/release/eathmemory -q MB MB, free memory : 3376 MB, used memory : 161 MB MB, free swap : 0 ME, used swap : 0 ME name test -filename /tmp/fiotest - icengine syncrw randreadbs &kinvalidate 0filesize 26 ) 81928-81928, (W) 81928-81928, (T) 81928-8192B, icengine=sync, icdepth=1 =21.1MiB/s,w=0KiB/s][r=2703,w=0 IOPS][eta 00m:005] 0: pid=725: Wed Apr 13 19:01:52 2022
clat (usec): min=195, max=2 lat (usec): min=195, max=2	s (21.4Ms/s)(2048MtB/100465msec) 7385, avg=382.22, stdev=263.41 7385, avg=382.28, stdev=263.41
30.00th=[ 306], 40.00th   70.00th=[ 379], 80.00th   99.00th=[ 1106], 99.50th   99.90th=[11076] bw ( Kik/s): min=10640, mozi iops : min= 1330, mozi lat (usec) : 250=0.59%, 50% lat (msec) : 2=1.43%, 4=0.6 cpu : us=0.45%, sys IO depths : 1=100.0%, 2=0 submit : 0=0.0%, 4=100 complete : 0=0.0%, 4=100 issued rwts: total=252144	<pre>[ 269], 10.00th=[ 277], 20.00th=[ 283], [ 420], 90.00th=[ 334], 60.00th=[ 351], [ 420], 90.00th=[ 510], 95.00th=[ 652], [ 1237], 99.90th=[ 1647], 99.95th=[ 2999], -23520, per=100.00%, avg=20874.93, stdev=1370.07, samples=200 = 2940, avg=2609.35, stdev=233.77, samples=200 =88.75%, 750-7.09%, 1000-2.07% X, 100-0.02%, 20-0.02%, 50-0.01% -1.95%, ctx=262144, majf=0, minf=36 WS, 4=0.0%, 50-0.0%, 32-0.0%, &gt;=64=0.0% 3%, 8=0.0%, 16-0.0%, 32-0.0%, c4=0.0%, &gt;=64=0.0% 3%, 8=0.0%, 16-0.0%, 32-0.0%, c4=0.0% 3%, 8=0.0%, 16-0.0%, 32-0.0%, c4=0.0%</pre>
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=Z62238/113, merc [centos@ip-172-158-19-16 ~]\$ ▌	e=0/11, ticks=97587/93, in_queue=97655, util=96.99%



#### 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:

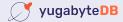
ios=0/0, merge=0/0, ticks=0/0, in\_queue=0, util=0.00%



#### Read: 4G

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

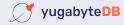
```
fio --name test --filename /tmp/fiotest
    --ioengine sync --rw randread --bs 8k
    --invalidate 0 --filesize 2G
```



		centos@ip-1/2-158-19-16:~	CHZ
	[centos@ip-172-158-19-16 ~]\$		
	<b>2</b>		
	I		
🍼 yugabyteDB			
$\infty$ · · ·			

0 All Rights Reserved

Cent cent	
[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 - 3179 MB	
total memory         :         3664 MB, free memory         :         3267 MB, used memory           total swap         :         0 MB, free swap         :         0 MB, used swap           [CentosEtp-172-158-19-16 ~]\$ []	: 166 MB : 0 MB



-	centos@ip-172-158-19-16:~
	-19-16 ~]\$ sudo suc "echo 1 > /proc/sys/vm/drop_caches" -19-16 ~]\$ ./eatmemory-rust/target/release/eatmemory -q : 3169 MB : 3664 MB, free memory : 3267 MB, used memory : 166 MB
	: 0 MB, free swap : 0 MB, used swap : 0 MB -19-16 ~]\$ fioname testfilename /tmp/fiotestioengine syncrw randreadbs 8kinvalidate 0filesize 4G ındread, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=sync, iodepth=1
Starting 1 process Jobs: 1 (f=1): [rd	; 1)][100.0%][r=18.3MiB/s,w=0KiB/s][r=2342.w=0 IOPS][eta 00m:005]
read: IOPS=2702	jobs=1): err= 0: pid=20803: Thu Apr 14 10:00:21 2022 , BW=21.1MiB/s (22.1MB/s)(4090MiB/193996mssc)
lat (usec): r	nin=219, max=92699, avg=369.32, stdev=214.49 nin=219, max=92699, avg=369.36, stdev=214.49
30.00th=[   70.00th=[	es (usec): 249], 5.00th=[ 262], 10.00th=[ 269], 20.00th=[ 281], 293], 40.00th=[ 306], 50.00th=[ 322], 60.00th=[ 343], 367], 80.00th=[ 412], 90.00th=[ 498], 95.00th=[ 635], 074], 99.50th=[ 1205], 99.90th=[ 1565], 99.95th=[ 1566], 99.95th=[ 1566], 99.95th=[ 1566], 99.95th=[ 1566], 99.95th=[ 1566], 99.95th=[ 1567], 99.95th=[ 1567]
iops : r	621] in=10144, max=24368, per=100.00%, avg=21622.96, stdev=1704.10, samples=387 in= 1268, max= 3046, avg=2702.87, stdev=213.01, samples=387 59=1.27%, 500=88.80%, 750=6.64%, 1000=1.96%
lat (msec) : a cpu : u	=1.30%, 4-0.01%, 10-0.01%, 20-0.01%, 100-0.01% sr=0.29%, sys=1.45%, ctx=524288, majf=0, minf=37 =100.0%, 2-0.0%, 4-0.0%, 8-0.0%, 13-0.0%, >=64=0.0%
submit : ( complete : (	→0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0% →0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0% total=522488,0.0, 0.9 short=0,0.0,0 dropped=0,0,0,0
	arget=0, window=0, percentile=100.00%, depth=1
Run status group ( READ: bw=21.1M	) (all jobs): B/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/n nvme0n1: ios=524 [centos@ip-172-158	247/140, merge=0/17, ticks=190788/67, in_queue=190821, util=98.31%



#### Read: 4G

• This is a summary from the run:

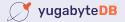
read: IOPS=2702, BW=21.1MiB/s (22.1MB/s) (4096MiB/193996msec)

- 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:~
	16 ~]\$ sudo suc "echo 1 > /proc/sys/vm/drop_caches" 16 ~]\$ ./eatmemory-rust/target/release/eatmemory -q 3179 MB 3664 MB, free memory : 3267 MB, used memory : 166 MB
total swap : [centos@ip-172-158-19-3 test: (g=0): rw=randrea	0 MB, free swap : 0 MB, used swap : 0 MB 16 ~]\$ fioname testfilesize 4G ad, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=sync, iodepth=1
fio-3.7 Starting 1 process	
	100.0%][r=18.3MiB/s,w=0KiB/s][r=2342,w=0 IOPS][eta 00m:005]
test: (groupid=0, jobs	=1): err= 0: pid=20803: Thu Apr 14 10:00:21 2022
	-21.1MiB/s (22.1MB/s)(4096MiB/193996msec)
	19, max=92699, avg=369.32, stdev=214.49
	19. mαx=92699, avg=369.36, stdev=214.49
clat percentiles (	Jsec ;: 5.00th=[ 262], 10.00th=[ 269], 20.00th=[ 281],
	5.vourine_ 262j, zo.vourine_ 269j, zo.vourine_ 261j, 40.vouthe_ 366j, 50.vouthe_ 322j, 60.vouthe_ 343],
	80.00th=[ 412],90.00th=[ 423,90.00th=[ 635].
	99.50th=[ 1205], 99.90th=[ 1565], 99.95th=[ 1680],
99.99th=[ 4621]	
	)144, max=24368, per=100.00%, avg=21622.96, stdev=1704.10, samples=387
	L268, max= 3046, avg=2702.87, stdev=213.01, samples=387
	27%, 500=88.80%, 750=6.64%, 1000=1.96%
	₩, 4=0.01%, 10=0.01%, 20=0.01%, 100=0.01% 200 and 145 at 145 at 145 at 16 at 15 at
	.29%, sys=1.45%, ctx=524288, majf=0, minf=37 .0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
	5.6, $2-0.08$ , $3-0.08$ , $3-0.08$ , $1-0.08$ , $2-0.08$ , $-0.08$ , $-0.08$ , $-0.08$
	(-1)
	=524288,0,0,0 short=0,0,0,0 dropped=0,0,0
	:=0, window=0, percentile=100.00%, depth=1
Run status group 0 (al	; ]005); [22:1M8/5), 21.1MtiB/s-21.1MtiB/s (22.1MB/s-22.1MB/s), io=4096MtiB (4295MB), run=193996-193996msec
READ: DW=21.1M1B/S	22.1MB/S), 21.1MIL/S-21.1MIL/S (22.1MB/S-22.1MB/S), 10=40900/LB (4235MB), FUN=1353900-1353900/SEC
Disk stats (read/write	
	40, merge=0/17, ticks=190788/67, in_queue=190821, util=98.31%
[centos@ip-172-158-19-3	

}



	•••	centos@ip-172-158-19-16:~
	<pre>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=Z702, BW=21.1MiB/s (22.1Mb/s)(4096MiB/193996msec) clat (usec): min=219, max=92699, avg=369.32, stdev=214.49 clat percentiles (usec):</pre>	samples=387
	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=4090	Miß (4295MB), run=193996-193996msec
	<pre>Disk stats (read/write): nvme0n1: ios-524247/140, merge=0/17, ticks=190788/67, in_queue=190821, util=98.3 [centosEy=172-158-19-16 -]5 ftoname testfilename /tmp/fiotestioengine sy test: (g=0): rw=randread, bs=(R) 81928-81928, (W) 81928-81928, (T) 81928-81928, io io=3.7 starting 1 process Jobs: 1 (f=1): [r(1)][99.2%][r=61.0M1B/s,w=0K1B/s][r=7935,w=0 IOPS][teta 00m:01s] test: (g=0upid=0, jobs=1): err= 0: ptd=22616: Thu Apr 14 10:09:11 2022 read: IOPS=4206, BW=32.9M1B/s (34.5W4/s)(4090M1B/124633msec) clet (nsec): min=534, max=305044, avg=237293.03, stdev=228597.81 clet percentiles (nsec): i .004h=[ 1153], 30.004h=[ 1256], 10.004h=[ 1336], i 20.004h=[ 1528], 30.004h=[ 1488], 40.004h=[ 264192], i 50.004h=[ 36592], 90.004h=[ 1483], 40.004h=[ 264192], i 50.004h=[ 36592], 90.004h=[ 1483], 29.004h=[ 329728], i 80.004h=[ 366592], 99.504h=[ 1122304], 99.904h=[1499136], i 99.904hh=[ 370752], 99.504h=[ 1122304], 99.904h=[ 249136], i 99.004hh=[ 366592], 99.004h=[ 7678896] bw ( KiB/s): min=17760, max=6016, avg=4202.43, stdev=1243.08, samples=249 lat (usec) : : 2=31.74%, 4=4.83%, 10=0.18%, 20=0.02%, 250=0.71% lat (usec) : : 2=0-88%, 4=0.01%, 10=0.01%, 50=0.01% cus =-0.28%, sys=1.73%, ctx=331382, m0jf=0, minf=36 lot (msec) : : 2=0-88%, 4=0.01%, 10=0.01%, 50=0.01% cum =: us=-0.00%, 4=100.00%, 8=-0.00%, 15=0.00%, 64=-0.00%, &gt;=64=0.0% submit : : 0=0.00%, 4=100.00%, 8=-0.00%, 16=0.00%, 32=0.00%, 64=-0.00%, &gt;=64=0.0% submit : : 0=0.00%, 4=100.00%, 8=-0.00%, 16=0.00%, 52=0.01% complete : 0=-0.00%, 4=100.00%, 8=-0.00%, 16=0.00%, 52=-0.00% submit : : 0=0.00%, 4=100.00%, 8=-0.00%, 16=0.00%, 52=-0.00% submit : : 0=0.00%, 4=100.00%, 8=-0.00%, 16=0.00%, 52=-0.00% submit : : 0=0=0.00%, 4=100.00%, 8=-0.00%, 16=0.00%, 52=-0.00% submit : : 0=0=0.00%, 4=100.00%, 8=-0.00%, 16=0.00%, 52=-0.00%, 64=-0.00% submit : : 0=0=0.00%, 4=0:00, 8=0.00%, 16=0.00%, 52=-0.00% submit : : 0=0=0.00%, 4=0:00, 0%, 8=-0.00%, 1</pre>	yncrw randreadbs 8kinvalidate 0norandommapfilesize 4G oengine=sync, iodepth=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=4096	3MiB (4295MB), run≡124633-124633msec
yugabyteDB	Disk stats (read/write): nvme0n1: ios=330920/105, merge=0/14, ticks=122124/59, in_queue=122162, util=98.0 [centos@ip-172-158-19-16 ~]\$ ▋	94%

0 All Rights Reserved

#### 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
```

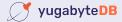


# Reality

#### • 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?

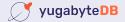


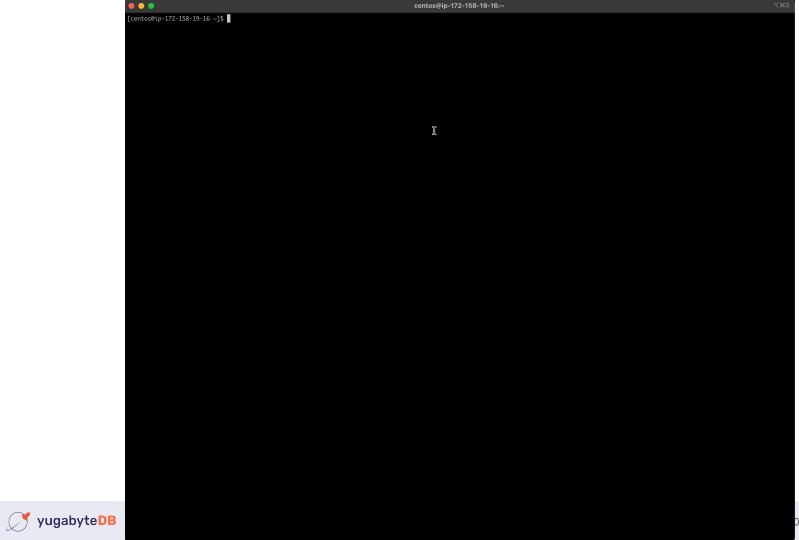
#### eatmemory

• 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

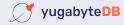
- Credits to original eatmemory.c tool (<u>https://github.com/julman99/</u> <u>eatmemory.git</u>)
- Let's try the same 2G run again!





0 All Rights Reserved

•••							centos@ip-172	2-158-19-16:		
[centos@ip-172-158-19-16 ~]\$ sudo suc "echo 1 > /proc/sys/vm/drop_caches" [centos@ip-172-158-19-16 ~]\$ ./eatmemory-rust/target/release/eatmemory -g										
			emory-rust/tar	get/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		Ø MB		
[centos@ip-172-158-3	19-16	~]\$								



( <b>•</b> )	• •	centos@ip-172-158-19-16:~
[cei ava	ntos@ip-172-158-19-16 ~]\$ sudo suc "echo 1 > /proc/sys/vm/drop_caches" tos@ip-172-158-19-16 ~]\$ ./eatmemory-rust/target/release/eatmemory -q lable memory : 1135 MB 1 memory : 32664 MB, free memory : 1228 MB, used mem	vorv : 2215 MB
toto [centest	11 swap : 0 MB, free swap ttos@tp-12-158-19-16 -]\$ fioname testfilename /tmp/fiotestioengine : (g=0): rw=randread, be:(\$ 81928-81928, (୩) 81928-81928, (T) 81928-81928	p : 0 MB syncrw randreadbs 8kinvalidate 0filesize 2G
Sta	-3.7 *ting 1 process : 1 (f=1): Гr(1)]Г100.0%∏Гr=21.2MiB/s.w=0KiB/s]Гr=2719.w=0 IOPS]Гeta 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)	
	<pre>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):</pre>	
	<pre>Cld pertentities (use(): 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], 99.90th=[10290]</pre>	
10	( S15)501-[L02.20] with ini=13392, max=23744, per=99.99%, avg=21365.87, stdev=1330.25, lops : min=1674, max=2968, avg=2670.73, stdev=166.30, samples=196 it (usec) : 250-0.87%, 500-89.28%, 750-6.43%, 1000-2.02% it (msec) : 2=1.34%, 4-0.03%, 10-0.02%, 20-0.01%, 50-0.01%	samples=196
c	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	
	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	
	status group 0 (all jobs): EAD: bww-20.9MiB/s (21.9MB/s), 20.9MiB/s-20.9MiB/s (21.9MB/s-21.9MB/s), io=20	48MiB (2147MB), run=98136-98136msec
m	< stats (read/write): me0n1: ios=262560/146, merge=0/17, ticks=96004/105, in_queue=96092, util=97. tos®ip-172-158-19-16 ~]\$ ┃	37%



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

• This is a summary from the run:

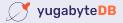
read: IOPS=2671, BW=20.9MiB/s (21.9MB/s) (2048MiB/98136msec)

- 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:~	
[ a t t t f S J	centos@ip-172-158-19-16 - vailable memory : otal memory : otal swap : centos@ip-172-158-19-16 - est: (g=0): rw-randread, H ito-3.7 tarting 1 process obs: 1 (f=1): [r(1)][100. read: IOPS-2671, 8W-20.5 clat (usec): min=173, r clat percentiles (usec): ni.04(bH=[257], 40 1 70.06H=[257], 40 1 70.06H=[257], 40 1 99.99H=[10220] bw ( KiB/S): min=13392 iops : min= 1674 lat (usec) : 250-0.87% lat (usec) : 250-0.87% lat (usec) : 250-0.87% lot depths : 1=100.0%, 4 complete : 0=-0.0%, 4 complete : 0=-0.0%, 4 complete : 0=-0.0%	<pre>[\$ sudo su c "echo 1 &gt; /proc/sys/vm/drop_caches" [\$ ./eatmemory-rust/target/release/eatmemory -q 1135 M8 0 M8, free memory : 1228 M8, used memory : 2215 M8 0 M8, free swap : 0 M8, used swap : 0 M8 [\$ fionome testfilename /tmp/fictestioengine sync.rw randreadbs 8kinvalidate 0filesize 2G ps=(R) 81928-81928, (W) 81928-81928, (T) 81928-81928, ioengine=sync, iodepth=1 %[[r=21.2Mi8/s,w=0Ki8/s][r=2719,w=0 IOPS][eta 00m:00s] err= 0: pid=26975: Thu Apr 14 10:36:21 2022 AMI8/s (21.3M8/s)[048Mi8/98136msec) max=24206, avg=373.64, stdew=218.70 pi. 00th=[ 265], 10.00th=[ 273], 20.00th=[ 285], 00th=[ 120], 50.00th=[ 285], 20.00th=[ 247], 00th=[ 121], 99.00th=[ 1614], 99.55th=[ 2409], ; max=23744, per=99.9%, avg=2165.87, stdew=130.25, samples=196 ; pia=299.28%, 759=6.43%, 1000=2.0% i=0.03%, 10=0.02%, 20=0.01%, 5=0-01% ; sym=4.98, x0=0.01%, 5=0-01% ; sym=4.98, x0=0.01%, 5=0-01% ; sym=4.96, x0=0.01%, 5=0-01% ; sym=4.91%, sym=4.01% ; sym=4.91%, sym=4.01%, sym=4.00% ; sym=4.91%, sym=4.01% ; sym=4.91%, sym=4.01%, sym=4.00% ; sym=4.91%, sym=4.01%, sym=4.01% ; sym=4.91%, sym=4.01% ; sym=4.91%, sym=4.01% ; sym=4.91%, sym=4.01%, sym=4.01% ; sym=4.91%, sym=4.01%, sym=4.01% ; sym=4.91%, sym=4.01% ; sym=4.91%, sym=4.01% ; sym=4.91%, sym=4.01% ; sym=4.91%, sym=4.01% ; sym=4.91%, sym=4.01% ;</pre>	
D	visk stats (read/write):	MB/s), 20.9MiB/s-20.9MiB/s (21.9MB/s-21.9MB/s), io=2048MiB (2147MB), run=98136-98136msec merge=0/17, ticks=96004/105, in_queue=96092, util=97.37%	



		centos@ip-172-158-19-16:~
	<pre>Starting 1 process Jobs: 1 (f-1): [r(1)][00.0%][r=21.2MiB/s,w=0KiB/s][r=2719,w=0 IOPS][eta 00 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 clat (usec): min=173, max=24206, avg=373.65, stdev=218.70 clat (usec): min=173, max=24206, avg=373.65, stdev=218.70 clat percentiles (usec): 1 1.00th=[ 251], 5.00th=[ 265], 10.00th=[ 273], 20.00th=[ 285], 1 30.00th=[ 297], 40.00th=[ 310], 50.00th=[ 1432], 90.00th=[ 347], 1 90.00th=[ 371], 80.00th=[ 412], 90.00th=[ 149], 95.00th=[ 644], 1 99.00th=[ 1074], 99.50th=[ 1221], 99.90th=[ 1614], 99.95th=[ 2449], 1 99.99th=[10220] bw ( KiKs): min=1332, max=23744, per=99.99%, avg=21365.87, stdev=133 iops : z1=100.0%, z=0.0%, 10=0.0%, 15=0.0%, 51=0.0%, s=64.0%, s=64.0%, s=64. submit : usr=0.31%, sys=1.49%, ctx=262145, mgif=0, minf=36 IO depths : l=100.0%, z=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, s=64. complete : 0=0.0%, 4=100.0%, 8=0.0%, 15=0.0%, 32=0.0%, 64=0.0%, s=64. complete : 0=0.0%, 4=100.0%, 8=0.0%, 15=0.0%, 32=0.0%, 64=0.0%, s=64. complete : 0=0.0% (atlo0.0%, 2=0.00th] [0.00th] [0.00th] Rum status group 0 (all jobs): Path. w=20 0thb/c (21.00M/c) 20 0thb/c 20 0thb/c (21.00M/c (21.00M/c) 21.00M/c) [0.00th] Path. avg=20 0thb/c (21.00M/c) 20 0thb/c (20.00tb/c) (21.00M/c) (</pre>	0.25, samples=196 196 0.0% -0.0% -0.0%
	<pre>READ: bw=20.9MtB/s (21.9MB/s), 20.9MtB/s-20.9MtB/s (21.9MB/s-21.9MB/s), Disk stats (read/write): nyme0h1: ios=262560/146, merge=0/17, ticks=96004/105, in_queue=96092, ut: (centosE)-172-158-19-16 -]5 fioname testfilename /tmp/fiotestioer test: (g=0): rw=randread, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-81 fio-3.7 Starting 1 process Jobs: 1 (f=1): [r(1)][100.0%][r=31.2MtB/s,w=0KtB/s][r=3988,w=0 IOPS][eta 00 test: (groupid=0, jobs=1): err= 0: pid=32758: Thu Apr 14 11:10:09 2022 read: IOPS=3525, BW=27.5MtB/s (28.9MB/s)(2048MtB/74366msec) clat (nsec): min=518, max=28349k, avg=283194.64, stdex=256978.87 lat (nsec): min=518, max=28349k, avg=283194.64, stdex=256978.87 lat (nsec): min=518, max=28349k, avg=283124.64, stdex=256979.92 clat percentiles (nsec): i .004th=[ 1768], 5.004th=[ 1320], 10.004th=[ 1448], i 20.004th=[ 1768], 5.004th=[ 2312536], 70.004th=[ 24672], i 50.004th=[ 1056766], 9.504th=[ 1202424], 99.904th=[ 1581056], i 90.004th=[ 1056766], 99.504th=[ 1204224], 99.904th=[ 1581056], i 90.004th=[ 10567663], 99.504th=[ 1204224], 99.904th=[ 1581056], i 90.004th=[ 10567663], 99.504th=[ 1204224], 99.904th=[ 1581056], i 90.004th=[ 3505.635, r50-6.05%, 1000-0.17% lat (usec) : : 2=22.806%, 4=3.41%, 10=0.31%, 20=0.01%, 50=0.91% lat (usec) : : 2=22.805, 4=3.41%, 10=0.31%, 20=0.01%, 50=0.01% lat (usec) : : 2=1.23%, 4=0.02%, 10=0.01%, 12=0.02%, 25=0.99% lat (usec) : : 2=1.23%, 4=0.02%, 10=0.01%, 10=0.00%, 64=0.00%, s=64 submit : u=r=0.27%, sys=1.63%, r50=0.01%, 10=0.00%, 64=0.00%, s=64 submit : u=r=0.27%, sys=1.63%, r50=0.00%, 10=0.00%, 64=0.00%, s=64 submit : u=r=0.27%, sys=1.63%, r50=0.00%, 32=0.00%, 64=0.00%, s=64 submit : u=r=0.274, 4=000, 8=0.00%, 10=0.00%, 64=0.00%, s=64 submit</pre>	il=97.37% ngine syncrw randreadbs &kinvalidate 0norandommapfilesize ZG 1928, ioengine=sync, iodepth=1 2m:00s] 2m:00s] 20.47, samples=148 148 0.0% -0.0%
🖉 yugabyteDB	Run status group Ø (all jobs): READ: bw=27.5MiB/s (28.9MB/s), 27.5MiB/s-27.5MiB/s (28.9MB/s-28.9MB/s), Disk stats (read/write): nrme0n1: ios=192264/64, merge=0/6, ticks=72937/55, in_queue=72978, util=6 [centos@ip-172-158-19-16 ~]\$ ■	

0 All Rights Reserved

#### 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,  $3000 \iff 3525$ ).
- 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%

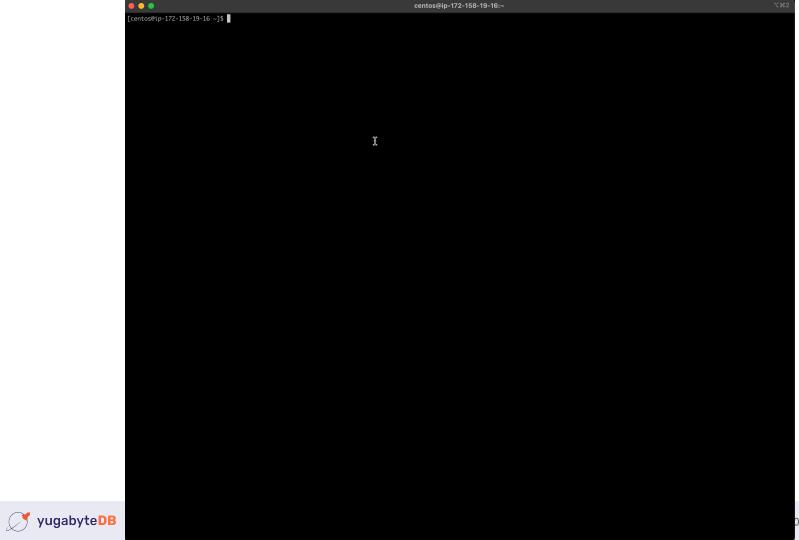


### Write: 2G

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

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





0 All Rights Reserved

### 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;

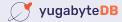
ios=0/61815

issued rwts: total=0,262144



# Write: 2G -- write details

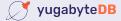
- Why aren't all writes cached, like all reads were?
  - Writes are special!
  - Writes cannot be discarded like reads can, they **must** be written first.
  - Writes can/should not exhaust available memory.
  - Therefore: vm.dirty\_background\_ratio, vm.dirty\_ratio, others.
  - Ratio is taken from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

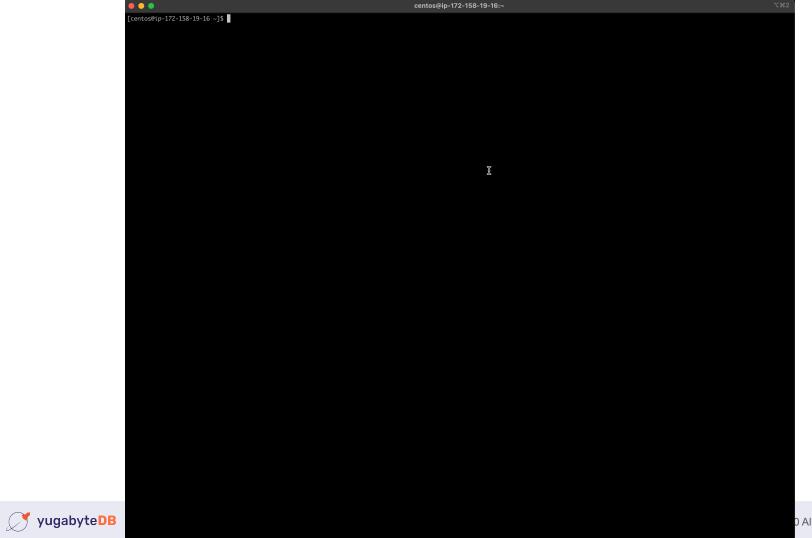


#### Write: 500M

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

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





0 All Rights Reserved

#### Write: 500M

• This is a summary from the run:

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

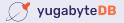
- My limits are 125 MBPS and 3000 IOPS.
- IOPS = 193000, MBPS = 1506.
- Reason; no write (throttling):

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

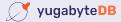
- The writes so far were also conducted with no memory used.
- Let's occupy 50% and perform the same tests again.



# 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
```



<u> </u>	•	

centos@ip-172-158-19-16:~

7.3

[centos@ip-172-158-19-16 ~]\$

I

🍼 yugabyteDB

0 All Rights Reserved

	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 m	emory		2217 MB		
total swap		0 MB, free swap		0 MB, used s	wap		Ø MB		
[centos@ip-172-158-	19-16 ~	-]\$							

I

yugabyteDB

..... centos@ip-172-158-19-16:~ [centos@ip-172-158-19-16 ~]\$ ./eatmemory-rust/target/release/eatmemory -q available memory : 1018 MB 2217 MB total memory 3664 MB, free memory 134 MB, used memory 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: (q=0): rw=randwrite, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=sync, iodepth=1 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%, ava=52686.61, stdev=52446.86, samples=79 : 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% : 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

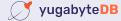
• This is a summary from the run:

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

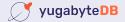
• Why? Available: 1018 MB, vm.dirty ratio: 30% = 305MB



# 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
```



entos@ip-172-158-19-16:~

[centos@ip-172-158-19-16 ~]\$ [

C**%2** 



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

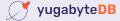
• This is a summary from the run:

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:

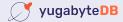
ios=17/20846

• 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  $\approx$  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



- 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.

