Test 1 - cluster test

Description

The testing use **two cluster**, one with cgroup block io limitation for writing(**5MB/s**) speed and another is unlimited. The cgroup io limitation just for QE and QD processes.

A python script used to monitor QE/QD processes and refresh the processes per second. If there are some QE/QD processes been found, put those processes to specific cgroup(have limited block io).

The testing data generated by another python script. The script opens 20 connection to master and uses `Faker` library to produce fake data and insert it to gpdb, it just produce

```
CREATE TABLE xxxx () WITH (appendoptimized={random from true or false});
```

and

```
INSERT INTO xxx (columns) VALUES (xxxx);
```

Configuration

Four VMs been createed via vSphere to run the testing, and each VM has same configuration:

- 16-core cpus
- 64G RAMs
- 50G system partition
- 100G data partition(**only used by GPDB**)

Cluster info:

- cluster with cgroup limitation: 2 hosts
- cluster without cgroup limitation: 2 hosts

And the tools used in this testing:

- `iotop`: monitor read and write speed of processes
- `iostat`: monitor actually disk io
- `blktrace`: monitor lifecycle of block io requests

Cgroup configuration

block io limitation of **test cgroup**:

```
dev_of_data_partition wbps=5MB rbps=5MB
```

Result

After running the testing for 2 clusters around **2 hours**, I get the collected data from cluster.

Table size

The test database named `testdb`, let's check the size of `testdb`:

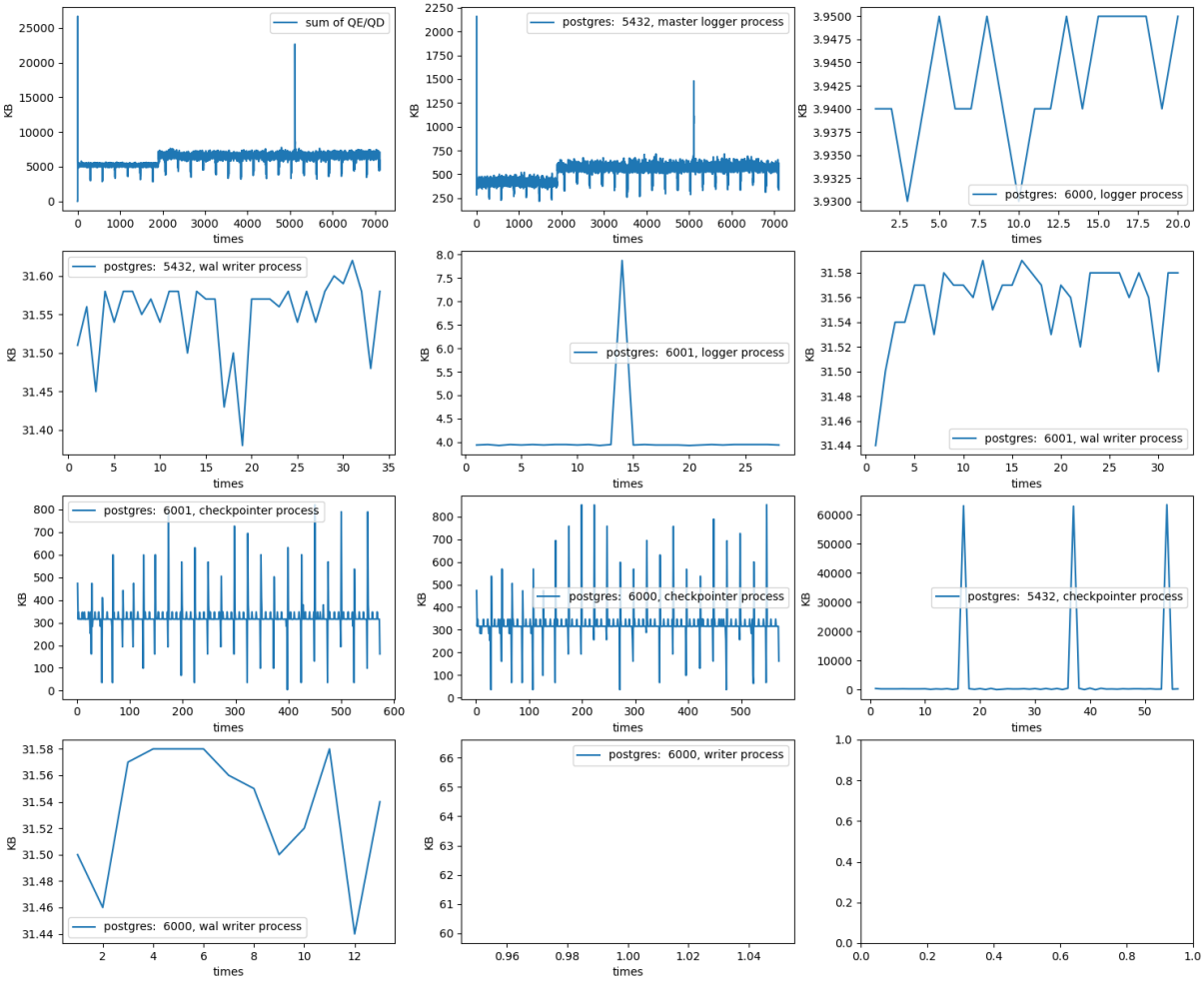
- cluster with cgroup limitation: **1215MB**
- cluster without cgroup limitation: **4170MB**

iotop datas

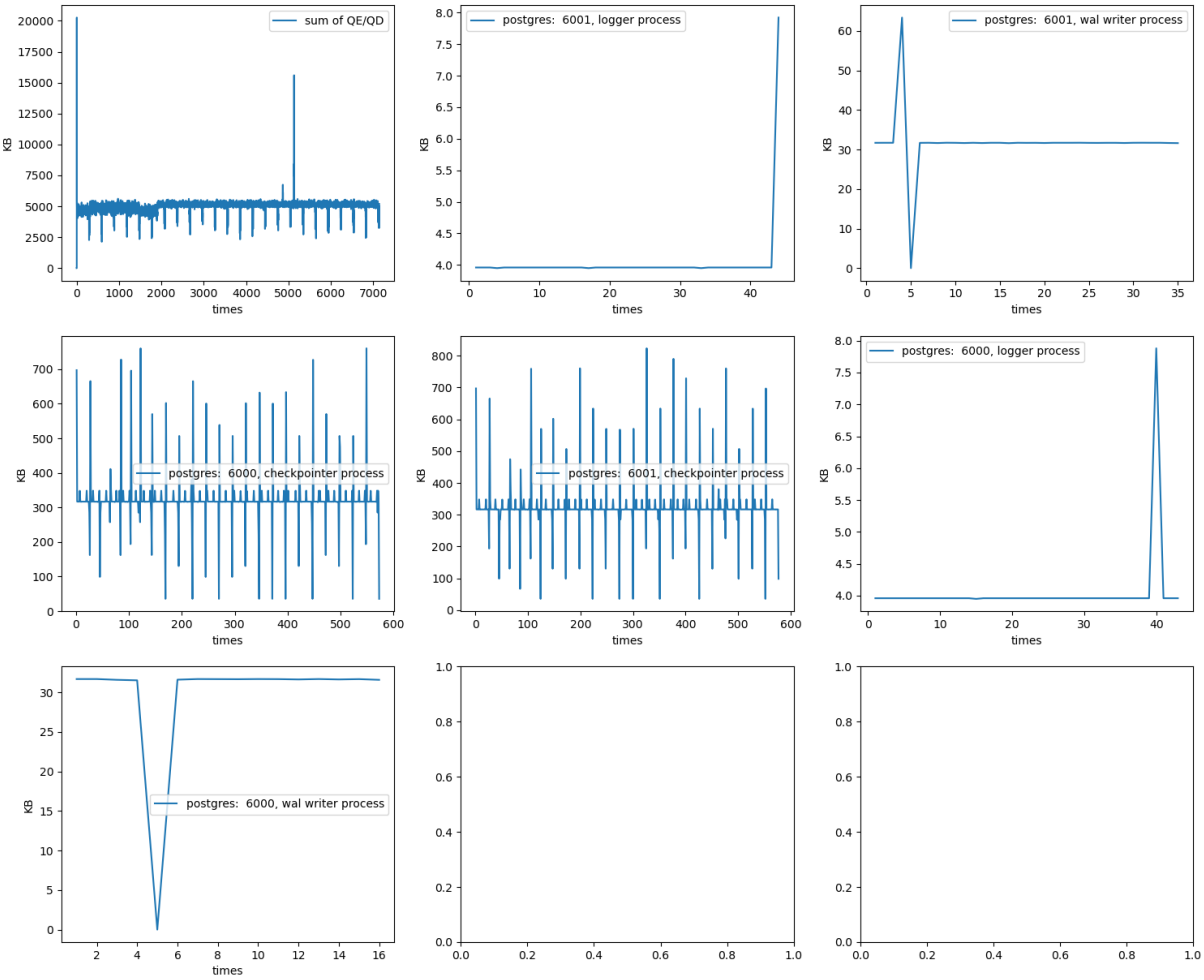
The following figures are `write speed` of every postgres related processes on every host:

Master with cgroup limitation

iotop of master with cgroup

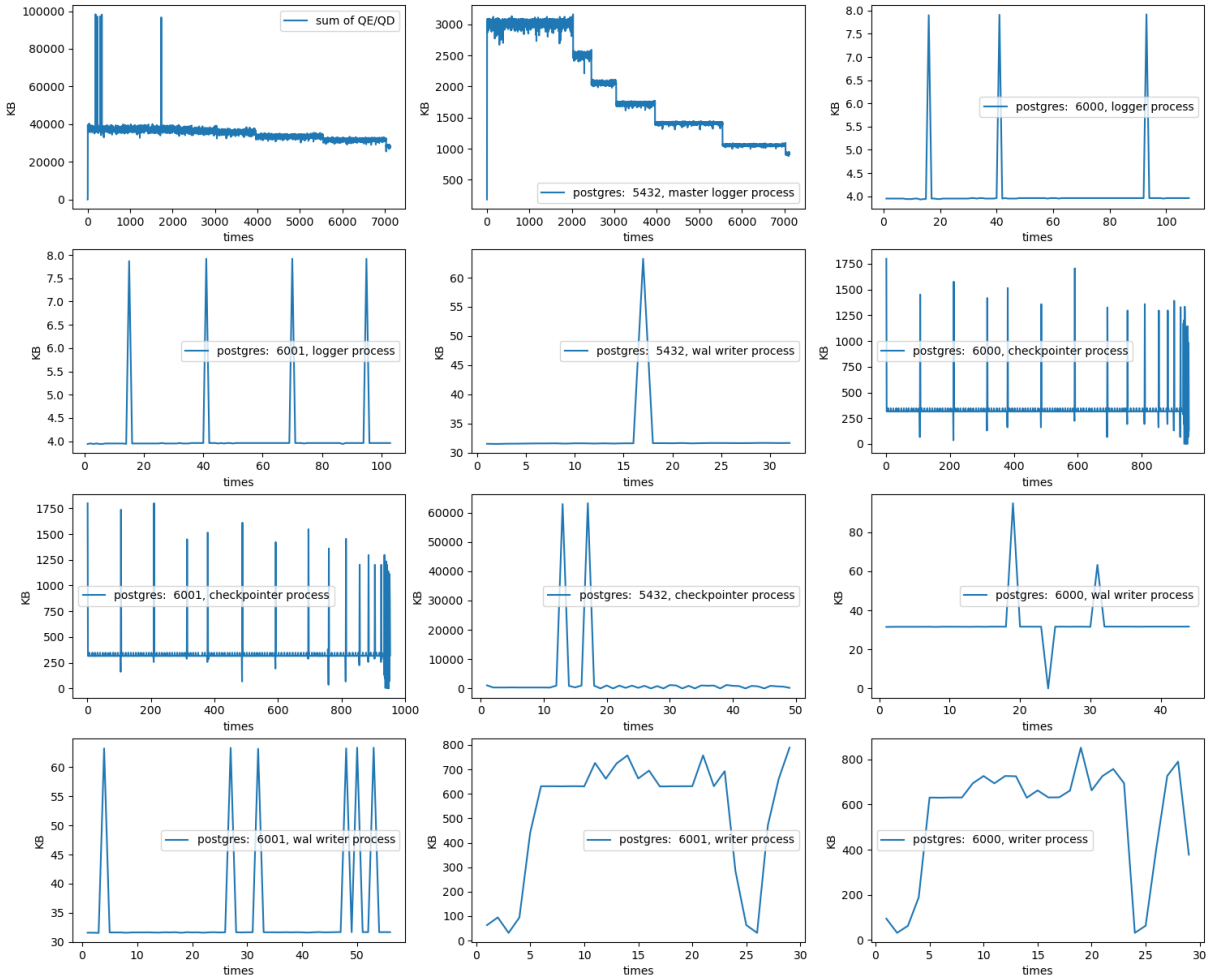


Segment with cgroup limitation

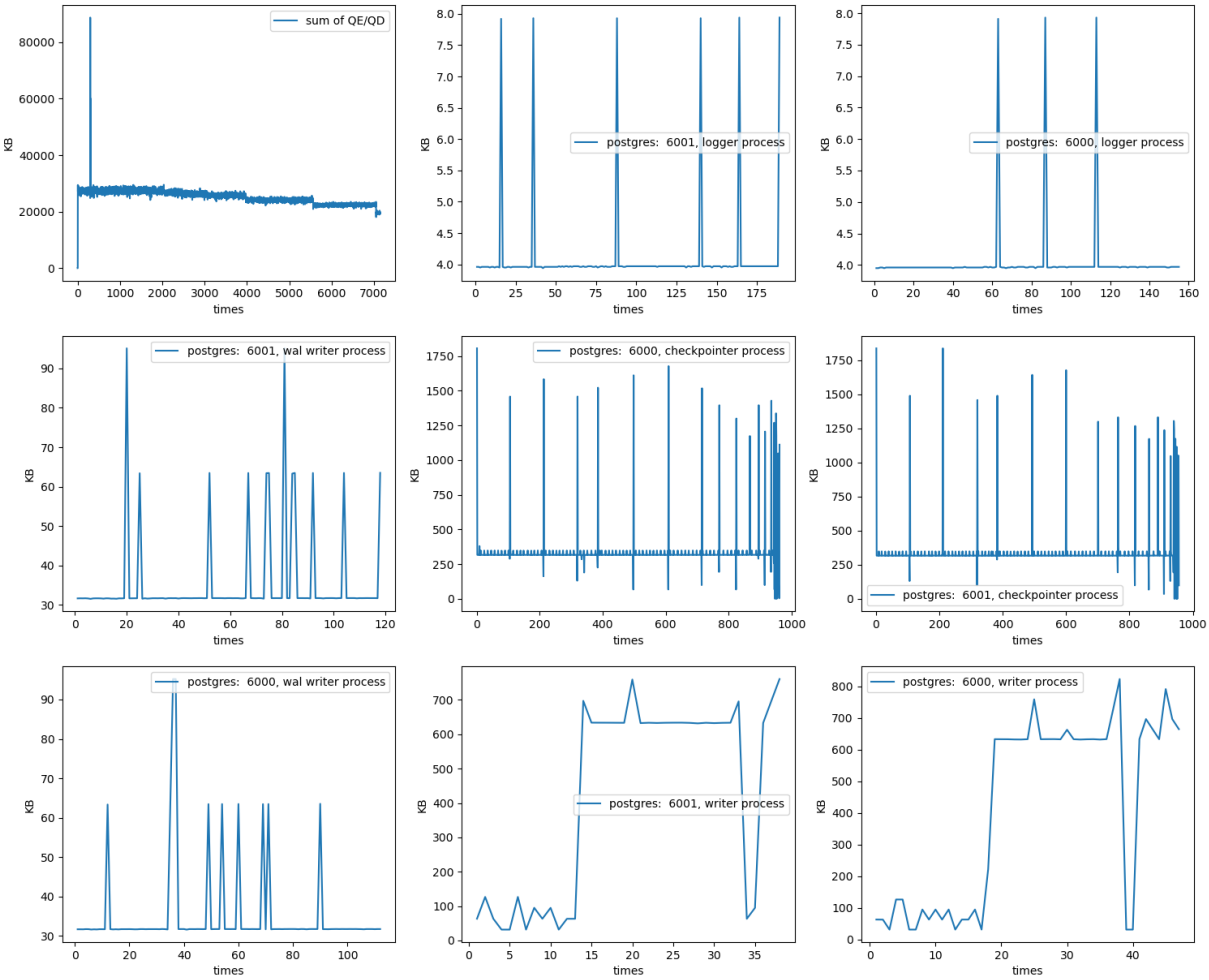


Master without cgroup limitation

iotop of master without cgroup



Segment without cgroup limitation



As you can see, the **write speed** of all QE/QD in 1 second is:

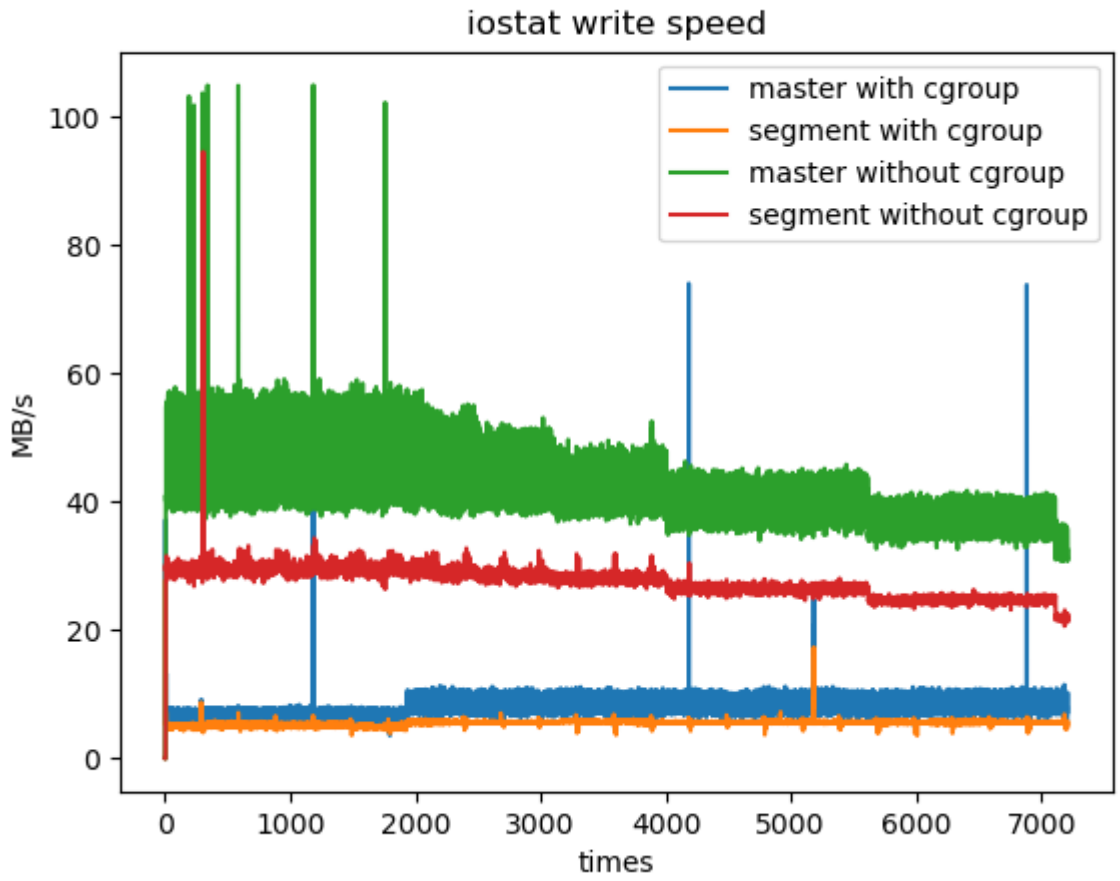
- cluster without cgroup limitation: around 30000 KB/s(30MB/s) in most time
- cluster with cgroup limitation: around 5000 KB/s(5MB/s) in most time

But in the cluster with cgroup limitation, there are several points which have high bandwidth. The first point at the start of x-axis, because the script used to move QE/QD processes to specific cgroup just run once every second, so there is a little delay to apply cgroup limitation to QE/QD processes. And other points with high bandwidth maybe because other processes have high output at this point.

And other processes just used a little write bandwidth compare to QE/QD.

iostat

iostat just show actually disk io bandwidth:



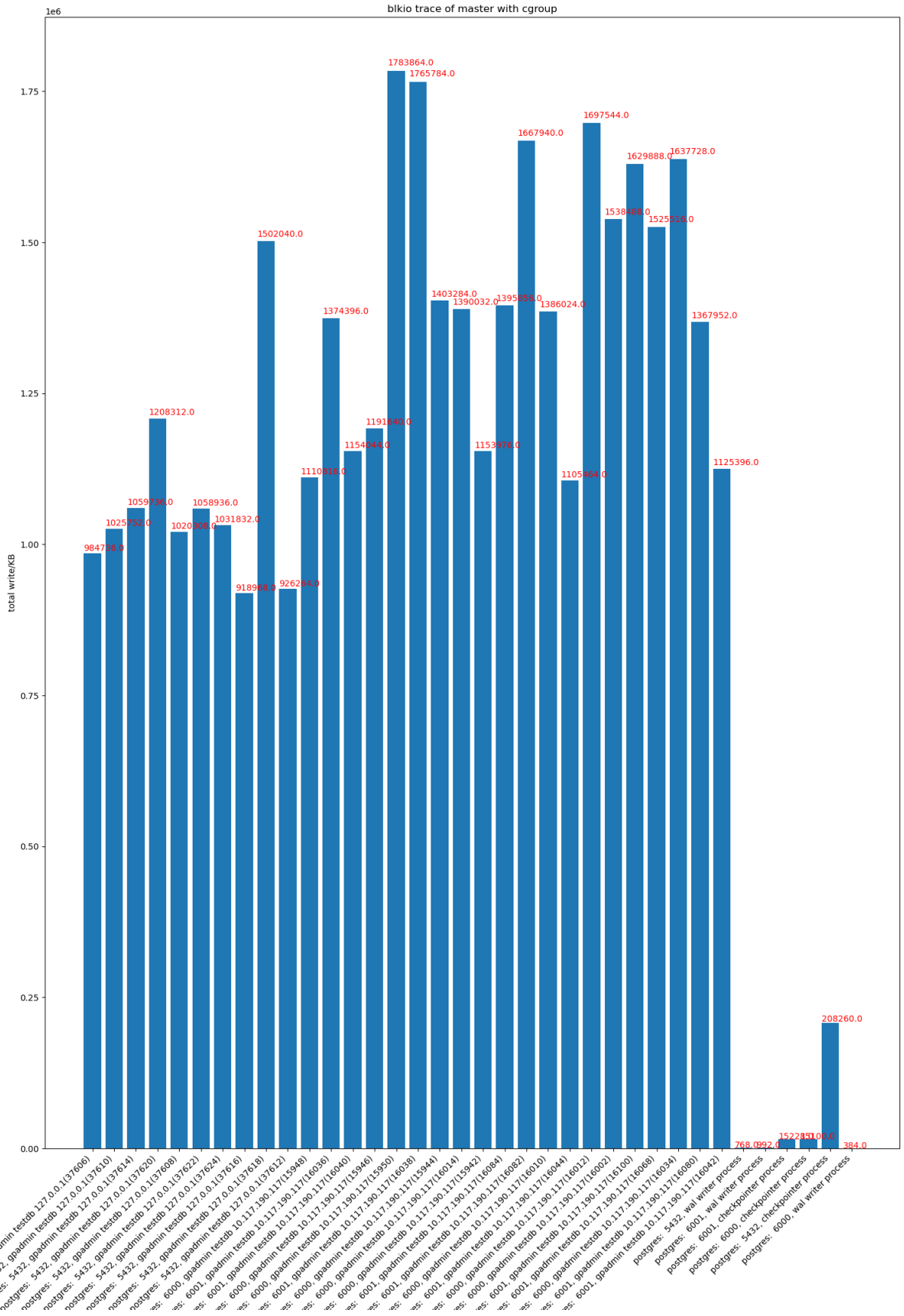
As you can see, whole disk io bandwidth just be filled by QE/QDs.(in the above **iotop** section)

blktrace

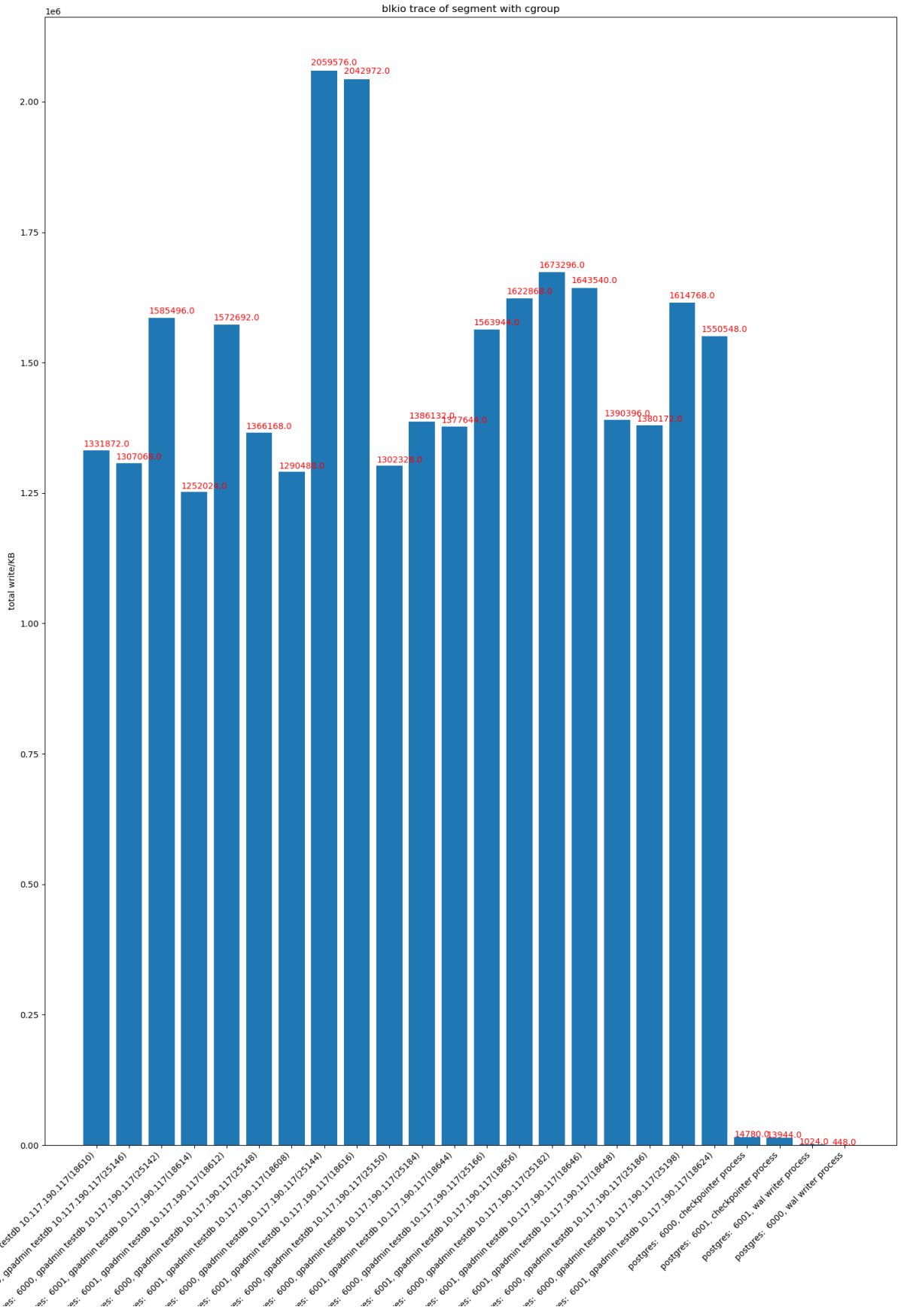
Blktrace used to monitor actually block io request received by the general block layer(after pagecache) of linux kernel.

Following figures show the total io requests of every postgres related processes received by general block layer:

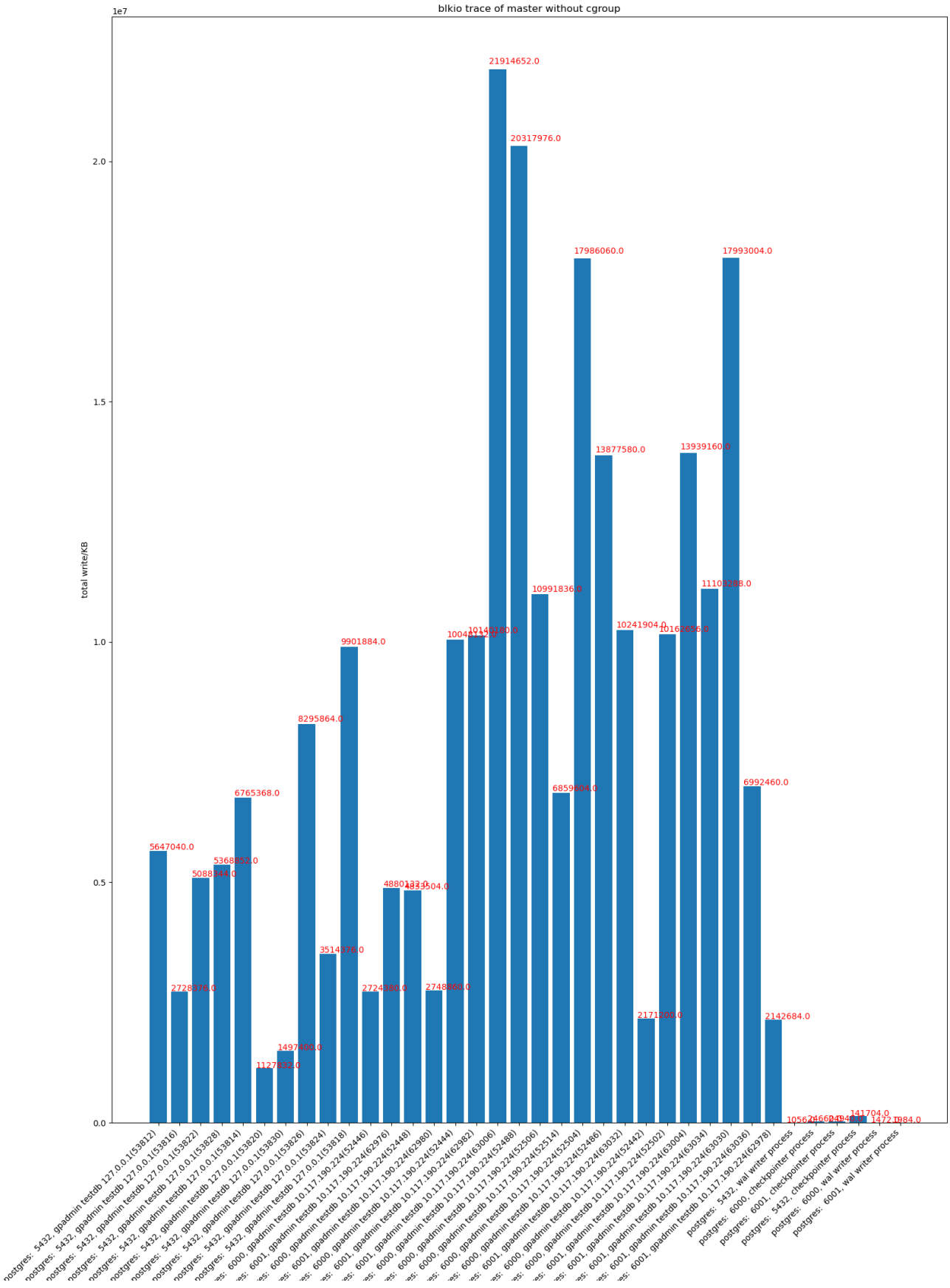
Master with cgroup limitation



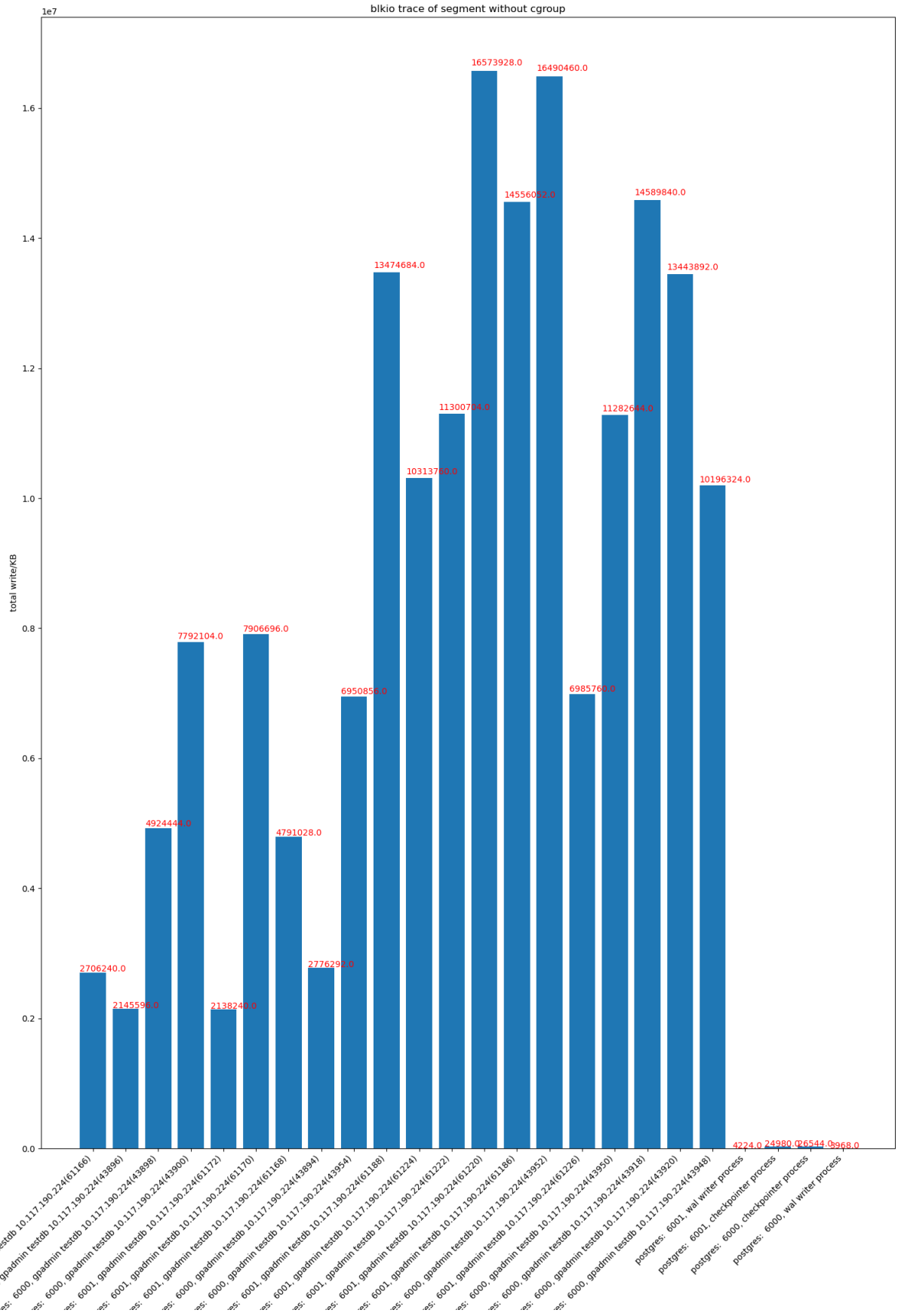
Segment with cgroup limitation



Master without cgroup limitation



Segment without cgroup limitation



So, whether in masters or segments, most io requests issued by QE/QDs, and most disk bandwitdth use by QE/QDs. And when limite io bandwidth of QE/QDs, other processes(like checkpointer) seems

can have more bandwith(compare `Master with cgroup limitation` and `Master without cgroup limitation`).

Summary

Limite block io bandwidth of QE/QDs can reduce a big part of whole bandwidth of cluster, and give more bandwidth to other processes. So, limit block io bandwidth for some specific QE/QDs(for example, some unimportant queries) and give more bandwidth to emergency queries is meaningful.

Steps for reproducing testing

Testing related code is located at my repo: https://github.com/RMTT/gpdb-cgroup-io-testing.git and following the `README` to reproduce testing.