1. Install Kernel
   Compile the kernel(4.17) with pblk option. And update the kernel of Guest PC(Qemu).
2. Install nvme-cli in guest pc.
   Download nvme-cli from git server. When I use the ‘apt-get’ command, I worked but the version was very low(v0.5). So download and compile the program.
3. initialize and create nvme with lightnvm and using pblk

   sudo nvme lnvm init -d nvme0n1
   sudo nvme lnvm create -d nvme0n1 -n mynvme -t pblk -b 0 -e 7

when creating the lnvm, start lun and end lun setup must be according to the number of pu(parallel unit.)

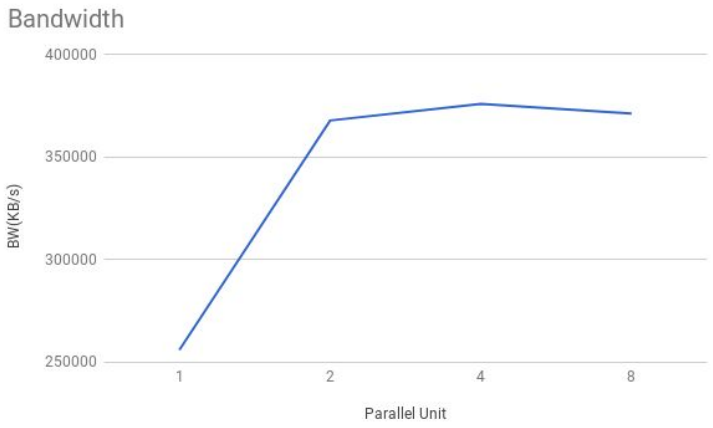
4. FIO Benchmark

Options

```
--rw=randwrite \
 --bs=4096 \
 --ioengine=libaio \
 --numjobs=1 \
 --iodepth=256 \
 --direct=1 \
 --thread \
 --write_iops_log=${BLOCKSIZE}-${RW}-${REPEAT} \
 --log_avg_msec=1000 \
 --size=$((1024*1024*1204))
```

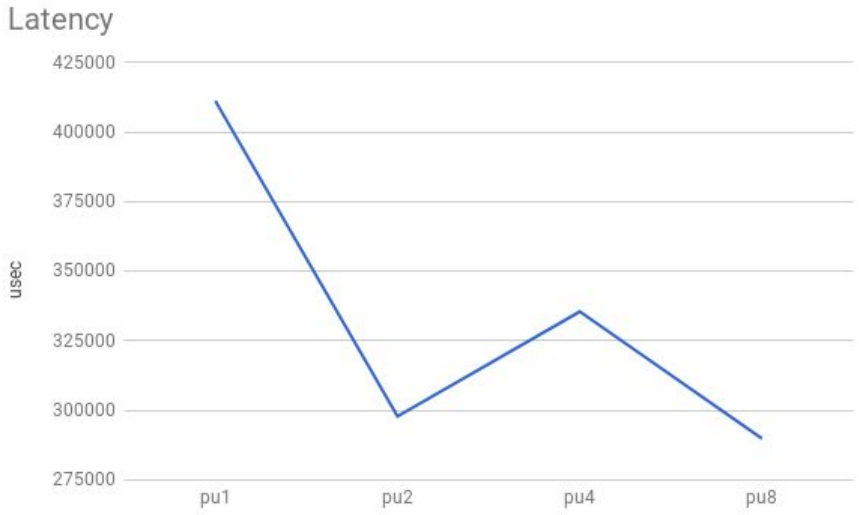
I ran the fio, while changing the pu number from 1 to 8. I ran each benchmark ten times. and I observed the bandwidth and the latency.

5. Result (Random Write - Bandwidth)



This graph is about Bandwidth for each parallel unit. The bandwidth value is average of test without the lowest value and the highest value. As the parallel unit increases , the bandwidth increase slightly. But the growth rate is smaller than I expected. And there is also other problem with Standard Derivation. Lager parallel unit cause larger standard Dev. I think the bandwidth stability get worse when the pu increase.

6. Result(Random-Write Latency)



Latency is reduced, when I increase the pu number. But it is not proportion to pu number.

7. ETC

I cannot know why the bandwidth does no increase while the pu number is increased. I think that the fio thread option will increase the bandwidth. I changed it to 4 thread and 8 pu,but there is no change at the bandwidth.

I also test Seq. read. but the result different from my expect. The bandwidth is not increase while increase the pu. I also increase the thread and jobnumber. but the result in not changed.