HPEVM latency profiling

Monday, March 11, 2019 10:25 AM

Target:

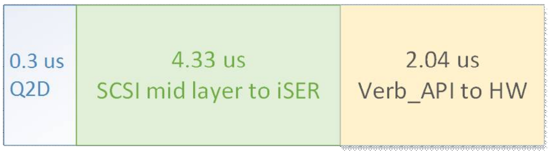
Breakdown the average 4K pagefault latency reported from lm_bench.
We want to know the mysterious behind the 4us that a page fault over RDMA needs to pay

Experiments and steps:

By blktrace, we got the following conclusions: [Blktrace Experiment]

a. The maximum data per IO request is 128K.
b. The average latency for an IO request is around 210 microsecond.
c. There is about 98% time consumption on D2C. The BIO layer only consume 2% of the execution time.
d. HW could handle two request parallelly.
e. The average 4K pagefault latency in total BIO latency is 210/32(128K data)/2(Handle parallelly) = 3.3 microsecond.
f. The one BIO latency is 6.5 microsecond.

By using lb_read_lat, we got the 2.04 microsecond latency from Verb API to HW. [lb_read_lat and wireshark related]



By using Wireshark, we got the following conclusions: [lb_read_lat and wireshark related]

a. The total amount of data transmit is matched to the IO requested.
b. The data transmit by 1K network packages.
c. No effect on modify the MTU number in both sides.

By using ftrace, we got the following conclusions: [Ftrace and mmap sample code]

a. Execution flow from BIO to HW: iscsi_queuecommand [scsi-mid-layer] ->iscsi_iser_task_init,iscsi_iser_pdu_alloc,iscsi_iser_task_xmit [ib_iser]
b. Execution flow from HW to BIO: ib_poll_handler [ib_core] -> __ib_process_cq [ib_core] -> iser_task_rsp [ib_core] -> rdma_port_get_link_layer [ib_core]() -> iscsi_iser_recv [ib_iser]
c. Detect time interval between mlx5_eq_int [mlx5_core] is about 120 microsecond.

By using mmap sample code, we got the following conclusion: [Ftrace and mmap sample code]

a. The average IO latency(Q2C) is about 210 microseconds in 4K, 64K, and 128K page fault.
b. The function pairs number is IO request + 1.

size	IO request	Ftrace function pair
4K	1	2
64K	1	2
128K	3	4
256K	3	4
512K	6	7
1M	9	10
10M	81	82

c. Estimate 1K package deliver time in HCA:
4K D2C: 0.000199981
64K D2C: 0.000229380
Diff time / diff packages = 29.399 (microsecond) / 64 -4 (packages) = 0.49 (microsecond/1K package)

Back to the lm_bench, the experiment is to count the latency between scsi mid-layer to ib_core layer. [Ftrace and mmap sample code] [THE LATEST STEP]

~~The average latency between scsi mid-layer to ib_core layer for one IO request is about 1.69*2 = 3.38 microsecond for 128K data.~~
The iscsi_queuecommand => 1.3 us
The ib_poll_handler => 1.18 us
The 4K average latency between scsi and ib_core is (1.3 + 1.18) / (128K / 4K) = 0.08

Experiments for ramdisk: [Ftrace and mmap sample code]

Method 1: 1.2245 microsecond Using this minus the time of tmpfs will get the 0.28 us.
This method is to measure the ext4 file system overhead by format the ramfs file as ext4. Detail steps in [Ftrace and mmap sample code]

Method 2: 1.5 microsecond
This method is to create /dev/ram0 and measure the performance.

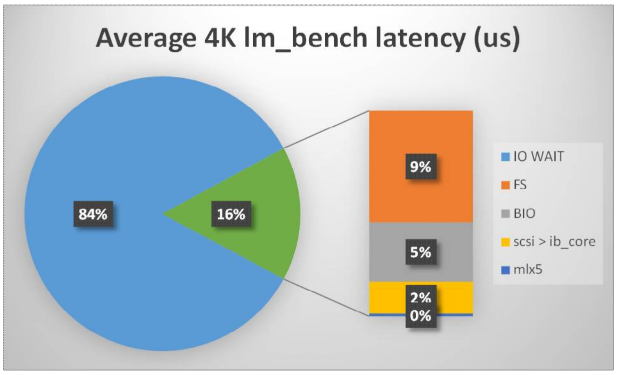
Results for 1G data in different storages:

Method	lmbench 4K(us)	Elapsed (s)	Percent of CPU	IO WAIT(s)	4K IO WAIT (us)	Page fault in stacks(us)
	From lmbench	From time	From time	From time	IO WAIT(s)/4K pages numbers	page fault exclusive IO WAIT(s) / 4K pages number
rdma	3.26	16.42	0.43	9.36	2.75	0.52

4K pages numbers 106509*32 [IO request number*(128k/4K)] = 3408288
Page fault in software stacks: includes FS, BIO, scsi mid-layer, ib_iser, ib_core, and mlx5_core.

IO WAIT	FS	BIO	scsi > ib_core	mlx5 (Predict)
2.75	0.28	0.15	0.08	0.007

IOWAIT is from
FS is from
BIO is from Q2D and divided by 2 because parallel handling
Scsi > ib_core is from
mlx5 is time of "Page fault in stacks" - the time of the above 4 items.
Excel: https://hpe-my.sharepoint.com/:x:/p/andy_liang/ETBvh5ijz_VMt0s6XxDFgTsBISx0wFRHjlOIV4Xub4v68A?e=Sc2Wpi



Experiments for latency on server side by ftrace:

Client side - IRQ = 23 us
HW handle to server tgtd = 11 us
Tgtd to HW handle = 110 us
HW handle to client = 7 us