

Bitfusion Flexdirect vs Network Relationship Experiments (tf_cnn_benchmark)

1. HARDWARE

| | |
|------------------|--|
| Host | Dell PowerEdge T320 Hosts 4 Intel(R) Xeon(R) CPU E5-2440 v2 @ 1.90GHz |
| GPU | GeForce GTX 1050 Ti |
| Interconnections | VMXNET 3 |

2. SOFTWARE

| | |
|--------------------|------------------------------|
| ESXi | 6.7.0 |
| FlexDirect Version | FlexDirect Version fd-1.11.7 |
| OS | Ubuntu 16.04 |
| Cuda | 9.0 |
| Tensorflow-Gpu | 1.12.0 |
| Nvidia Driver | 430.26 |

3. TRAINING DATA

tf_cnn_benchmark: https://github.com/tensorflow/benchmarks/tree/master/scripts/tf_cnn_benchmarks

```
$ python tf_cnn_benchmarks.py --num_gpus=1 --batch_size=32 --model=resnet50 --variable_update=parameter_server
```

4. TEST CASE

FlexDirect Server At: 10.110.125.113 with a 1050 TI GPU

| # | TestCase | Description | FlexDirect Client At |
|---|---------------|---|----------------------|
| 0 | Baseline | 1. Run directly on the FlexDirect Server using GPU Passthrough 2. This test case server as the baseline for later test | None |
| 1 | Local | 1. FlexDirect server and client are on the same VM | 10.110.125.113 |
| 2 | Same ESXi | 1. FlexDirect server and client are on different VM but on same ESXi | 10.110.125.80 |
| 3 | Same LAN | 1. FlexDirect server and client on different VM but in the same LAN | 10.110.124.237 |
| 4 | Different LAN | 1. FlexDirect server and client on different VM and in different LAN | 10.184.75.87 |

5. BANDWIDTH TEST

| | | | |
|----------------|---------------|----------------|----------------|
| | 10.110.125.80 | 10.110.125.113 | 10.110.124.237 |
| 10.110.125.113 | 10 Gb/sec | | 943 Mb/sec |
| 10.110.125.80 | | 7.19 Gb/sec | |
| 10.110.124.237 | | 531 Mb/sec | |

6. RESULT (Without Network Tuning)

Batch_size = 32

| TestCase# | 0 (Standard) | 1 | 2 | 3 | 4 |
|-------------------|-----------------|---|------|------|------|
| Performance Ratio | 1 | 1 | 0.88 | 0.05 | Fail |

Batch_size = 16

| TestCase# | 0(Standard) | 1 | 2 | 3 | 4 |
|-------------------|-------------|---|------|------|------|
| Performance Ratio | 1 | 1 | 0.65 | 0.03 | Fail |

| TestCase# | Trained Images Per Second (images/s) | Max GPU utilization | GPU memory usage |
|-----------|--|---|----------------------------|
| 0 | 54.22 (batch_size = 32) 48.83 (batch_size = 16) | 100% (batch_size = 32) 98% (batch_size = 16) | 96.5% |
| 1 | 54.30 48.79 | 99% 98% | 94.3% |
| 2 | 47.98 31.94 | 97% 92% | 94.3% |
| 3 | 2.9 1.7 | 17% 9% | 94.3% |
| 4 | FAIL (connection time out) | FAIL (connection time out) | FAIL (connection time out) |

7. RESULT(With Network Tuning)

7.1. Network Tuning Options I Tried (Based on <https://www.vmware.com/techpapers/2011/best-practices-for-performance-tuning-of-latency-s-10220.html>)

| Options |
|--|
| Set the Power Management Mode to Maximum Performance |
| Processor Settings: set Turbo Mode to enabled |
| Processor Settings: set C States to disabled |
| Disable physical NIC interrupt moderation on the ESXi host |
| Disable virtual interrupt coalescing for VMXNET 3 virtual NICs as follows. |
| Set VM Latency Sensitivity |
| Stopping the guest firewall (iptables) |

Disabling SELinux

Batch_size = 32

| TestCase# | 0 (Standard) | 1 | 2 | 3 | 4 |
|-------------------|-----------------|---|------|------|------|
| Performance Ratio | 1 | 1 | 0.94 | 0.04 | Fail |

Batch_size = 16

| TestCase# | 0(Standard) | 1 | 2 | 3 | 4 |
|-------------------|-------------|---|------|------|------|
| Performance Ratio | 1 | 1 | 0.89 | 0.03 | Fail |

| TestCase# | Trained Images Per Second (images/s) | Max GPU utilization | GPU memory usage |
|-----------|---|----------------------------|----------------------------|
| 1 | 54.11 (batch_size = 32) | 99% (batch_size = 32) | 94.3% |
| | 48.77 (batch_size = 16) | 98% (batch_size = 16) | |
| 2 | 51.15 | 97% | 94.3% |
| | 43.54 | 94% | |
| 3 | 2.2 | 14% | 94.3% |
| | 1.3 | 5% | |
| 4 | FAIL (connection time out) | FAIL (connection time out) | FAIL (connection time out) |

8. CONCLUSION

- 1) Setting VM Latency Sensitivity has the most positive effect on decreasing latency
- 2) After tuned,
 - a) Test Case 1 should not and did not improve performance since it is local
 - b) Test Case 2 has improve greatly by 6%(batch_size=32) and 35%(batch_size=16)
 - c) Test Case 3 has not improve, the reason maybe the low bandwidth connection between two hosts
 - d) Test Case 4 still failed, maybe due to the distance between client(local) and server(nimbus) - too far away