EANTC Independent Test Report

Huawei Cloud AI Fabric

Ultra-High-Speed Ethernet Benchmarking

June 2018

HUAWEI

EANTC

Introduction

Enterprises and cloud providers increasingly need to process large-scale big data, artificial intelligence workloads. High-performance computing (HPC) infrastructures required to handle such tasks need to excel in compute, storage, and networking performance. While compute and storage metrics depend on hardware development (CPUs and SSDs), the networking performance can be improved using advanced configs based on standard hardware.

Test Highlights

- ➔ The Huawei AI Fabric solution reduced request latency in Intel MPI Benchmarking scenarios, between 0.2 % for small request sizes and 44 % for very large request sizes
- ➔ AI Fabric reduced latency in Distributed File System (DFS) scenarios up to 15 % for large I/O depth requests
- ➔ The solution integrates 100GbE and 25GbE SmartNICs, leaf and spine switches successfully
- ➔ Advanced implementations of RDMA over Converged Ethernet and Explicit Congestion Notification enable lossless, resilient transport

In parallel computing environments, the network performance determines the efficiency of message passing interfaces (MPI) that transfer data between compute tasks.

Huawei commissioned EANTC to verify the networking efficiency of Huawei's *Cloud AI Fabric*. The solution includes data center switches and smart network interface cards (NICs) based on 100GigabitEthernet (100GbE) and 25GigabitEthernet (25GbE).

The Huawei solution uses Remote Direct Memory Access (RDMA) over Converged Ethernet on top of the IP/UDP protocol, also called RoCEv2. RDMA enables zero-copy transfer of network data directly to the memory, bypassing the CPU and other compute components and thus increasing the efficiency.

One major architectural issue of IP/UDP messaging on top of standard Ethernet is its queuing behavior. Under load, packets get queued and eventually dropped once queues overflow. In such cases, a higher layer in the application needs to handle retransmission of lost messages, which would decrease performance.

EANTC verified Huawei's AI Fabric data transfer solution which uses Explicit Congestion Notification (ECN) to optimize the queuing behavior, reduce latency and minimize loss of IP packets in transit.

Overview of Test Areas And Results

At EANTC, we performed a set of industry-accepted Intel MPI Benchmarks to verify the Huawei solution's ability to forward data traffic efficiently. To rate its efficiency, we compared the ECN-optimized variant with a non-optimized configuration of the same components. As a key performance indicator, we verified whether the data forwarding latency would decrease during our tests once dynamic ECN got enabled.

For network reliability as critical point for data center operators, we tested the network convergence to verify that the Huawei solution was resilient against network link failure while running HPC applications.

Additionally, we performed Distributed File System (DFS) benchmarks to verify the ability of the Huawei solution to handle file transfers over the optimized network infrastructure efficiently. Due to the development of SSDs, storage performance has reached and exceeded network speed. We verified how the *AI Fabric Ultra-High-Speed Ethernet* performed in this situation.

Huawei Hardware and Software

Role	Hardware	Software
Leaf	CE6865	8.180 (V200R005C00SPC100)
Spine	CE8850-64CQ	8.180 (V200R005C00SPC100)
Spine	CE12800E	8.180 (V200R005C00SPC100)
Compute node	RH2288 V2	Ubuntu 17.10
Storage node	RH2288 V3	Ubuntu 17.10
HPC node	XH321 V3	CentOS 7.3.1611
SSD	ES3600P V3	ES3600P V3

Test Bed

The main test object was the top-of-rack (TOR) switch. Huawei constructed a complete HPC scenario with eight top-of-rack switches as leaves. The main architectural goal was to achieve maximum full-mesh scale without increasing latency.

For load-balancing and network redundancy, two spine routers were each connected to all eight leaf switches.

This setup resulted in a total of 32 100GbE links network capacity towards the servers, implemented by the Huawei AI fabric.

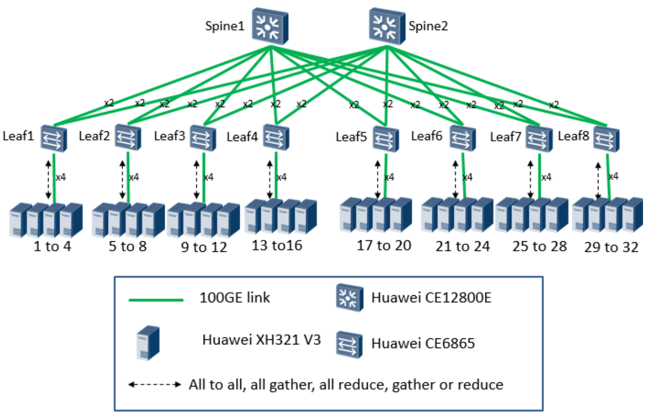

Figure 1: HPC Test Setup

Huawei used standard x86 servers to run the benchmarking tests. The model used was the Huawei XH321 V3.

HPC Performance Tests

HPC defines the use of super computers and parallel processing techniques for running complex computational calculations. HPC technology focuses on developing parallel processing and systems by incorporating parallel computational techniques. The MPI (Message Passing Interface) is a standardized and portable message-passing standard to focus on parallel computing architectures. Intel MPI benchmarks provide a set of MPI performance measurements for communication operations for a range of message sizes. The Intel benchmarks are available as open source software to enable independent benchmarking tests.

We verified performance stability and resiliency using Intel MPI benchmarks. The latency in this test is defined as the time taken to complete a computation task.

Baseline Latency Test

To obtain reliable reference data of the 32-server environment, we measured the latency by conducting five selected Intel MPI benchmark tests. We performed each test in two solution configurations:

- First without dynamic ECN enabled to obtain the baseline values; This non-optimized configuration allowed the ECN function to react to congestion through static gate values. Once set to pre-configured value, it remained unchanged over test runtime.
- Then with dynamic ECN enabled to compare it with baseline latency. The feature under test provided the ability to adjust the gate value dynamically according to the congestion state.

IMB Test Case	Description	Maximum Message size [Bytes]
AllToAll	Q*8 MB, in the case of np number of processes, every process inputs x*np bytes (x for each process) and receives X*np bytes (x from each process)	524 288
All Gather	(Q+1)*4 MB, every process inputs x bytes and receives the gathered x*np bytes, where np is the number of processes (x=4MB)	4 194 304
Gather	8MB, every process inputs x bytes and receives the gathered x*np bytes, where np is the number of processes (x=8M). The root of the operation is changed round robin	524 288
All Reduce	8MB, it reduces a vector of length l = x/size of (float) float items (x = 8M)	524 288
Reduce	8MB, it reduces a vector of length l = x/size of (float) float items (x = 8M) The root of the operation is changed round robin	4 194 304
Number of repetitions per run: 5		
Q: Number of active process X: Maximum size of the passing messages		

Table 1: IMB Test Setup

Each test was repeated five times to obtain representative latency data. The Table 1 on page 3 describes the parameter settings.

Features enabled

The server network interface cards (NICs) Mellanox CX4 100GE NIC were configured with special settings for RDMA. The NIC has the task to process the RDMA over Converged Ethernet (RoCEv2) protocol, encapsulate RDMA data into Ethernet frames and transmit them over the Ethernet network. RDMA is a technology for high-performance computing servers that enables high speed data exchange from NIC directly to the memory avoiding CPU intervention.

To avoid high delay due to packet loss and queueing, the switching layer requires congestion control mechanisms. The Huawei engineers enabled PFC (Priority-based Flow-Control) as queue-based flow control on the Huawei leaf switches. When the outbound port queue of the switch gets congested, the

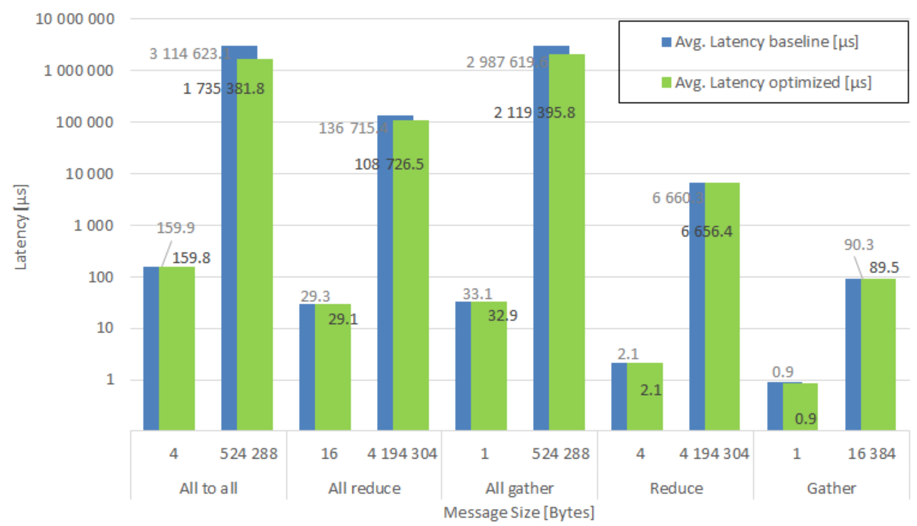

Figure 2: Average Latency Of Intel MPI Benchmarks, Regular And Dynamic ECN-Enabled

PFC-enabled solution is able to notify the upstream device to derate this queue without affecting other queues. This setting provided prerequisites for switching congestion control.

The ECN (Explicit Congestion Notification) flag in the Differentiated Services header of IP packets is enabled once switch congestion occurs. For AI Fabric solution, ECN configuration is dynamic; there is no pre-configured static gate value. It is usually automatically adjusted through learning. When ECN-marked IP packets arrive at the destination server, the server's NIC triggers a CNP (Congestion Notification Packet) message to the peer to warn the source server of network congestion.

We performed in total 107 runs, each repeated for baseline test and measurement when dynamic ECN was enabled. We analyzed the minimum, average and maximum latency results. We obtained 72 results carrying expected reduction of latency when dynamic ECN was enabled.

Minimum vs. Maximum Latency

We observed consistent minimum and maximum latency across five repetitions of each test configuration. Results are summarized in figure 2.

The overall variation of latency results between minimum and maximum values were as expected, specifically in the all to all, all gather and all reduce test cases.

The results confirmed our expectations: Task load and message sizes are proportional to switch congestion state. The heavier loaded the tasks or the message sizes, the higher the link utilization, and congestion is more likely to occur. The table below shows an example of the AllToAll test where message sizes were related to ECN counters and link utilization. The statistics showed that the ECN was triggered when switch congestion occurred.

Server-gathered Statistics	Value per message size	
	16 Bytes	524 288 Bytes
Number of ECN markers sent by a single leaf	0	403 192
Average traffic rate over spines [Gbit/s]	32	116 800

Table 2: Server-gathered Statistics

We observed that both minimum and maximum latency reduced when dynamic ECN was enabled.

- AllReduce decreased the latency by 0.2 µs (0.7%) from 29.3 µs to 29.1 µs providing the highest delta of latency among all tasks for small message sizes.
- The average latency of large message sizes significantly reduced. In the AllToAll test, latency decreased by 1.4 s (44.3%) to 1.7 s when dynamic ECN was enabled.

In addition, we observed the system under test status while performing the benchmark tests. CPU utilization was 13 % on the spine, and memory was filled to 11 % (1 GByte of 8 GByte). The CPU utilization was 9 % on the leaf, and memory usage was 27 % (1 GB or 4 GB). The Figure 3 shows the ECN marker that was set in the IP packet when congestion occurred.

Summary

- Latency increased linearly proportional to message size. This behavior did not change when dynamic ECN was enabled as expected.
- Overall latency decreased when dynamic ECN was enabled.
- 72 tests of 107 performed test runs showed reduction of average latency when dynamic ECN was enabled. In cases when the network is not congested at all, there is nothing to optimize.
- The average latency reduced significantly in heavy tasks for big message sizes when dynamic ECN was enabled.

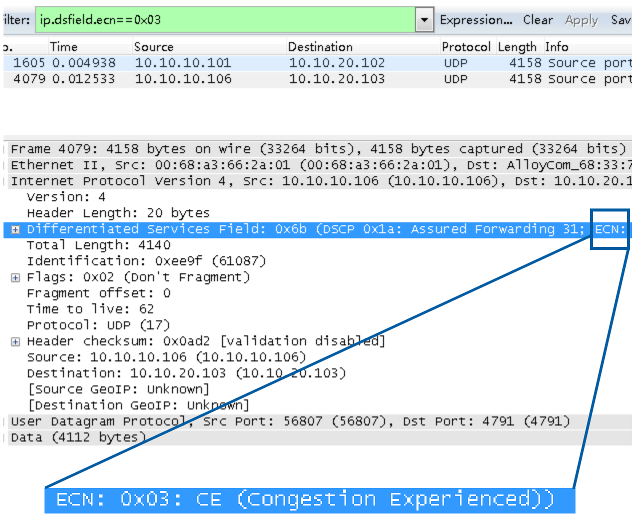

Figure 3: ECN (Explicit Congestion Notification) Marker In IP Packet

The following table depicts more details of some example cases. For instance, the average latency of the AllGather test with 256 Bytes request size got reduced from 1011 μs to 762 μs, an improvement of 24.6 %.

No.	Avg. latency [μs] Without dynamic ECN	Avg. latency [μs] With dynamic ECN	Delta [μs]	%
AllGather (256 Bytes)				
1	1 011.2	762.2	249.0	24.6
AllReduce (524,288 Bytes)				
2	13 438.0	9 192.3	4 246.1	31.6
AllToAll (262,144 Bytes)				
3	1 416 474.8	919 207.7	497 267.1	35.1

Table 3: Examples Of Compared Latency Values

Summary

- The ECN-optimized average latency values were evenly distributed in all intervals from 10 μs to 10 s.

Figure 4 shows the latency measurements of the AllGather tests. The average latency decreased in the test runs with all message sizes, from 0 to 2^{19} (524,288 Bytes). The highest latency 2.987 s decreased by 0.8868 s (29 %) to 2.119 s after dynamic ECN was enabled. The results AllToAll and AllReduce were similar and are not displayed here.

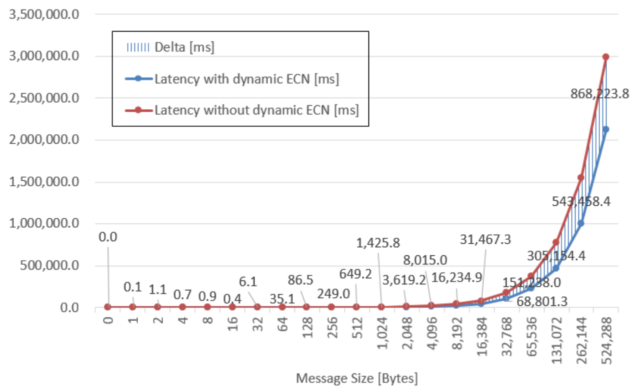

Figure 4: IMB Average Latency [μs] For AllGather Tasks

Summary

- Full data collection delivered more seamless latency reduction, while latency of partial data operation slightly decreased when dynamic ECN was enabled.

Mix of RoCEv2 and TCP Background Traffic

We tested the system against additional TCP background traffic to verify that the Huawei ECN-enabled AI Fabric could handle switch congestion in mixed data center traffic. We expected that RoCEv2 traffic would not be affected by the TCP background traffic.

To verify this claim, we installed an open-source software traffic generator (iPERF) to generate TCP traffic. To prevent RoCEv2 traffic from being affected on the same machine, we divided the servers into two groups: 16 servers were dedicated to run the Intel MPI benchmark, and another 16 servers were running iPERF to send TCP traffic at 20 Gbit/s (see Figure 5) per server.

The average latency of the baseline tests was slightly lower than running IMB over 32 servers as described in the previous test.

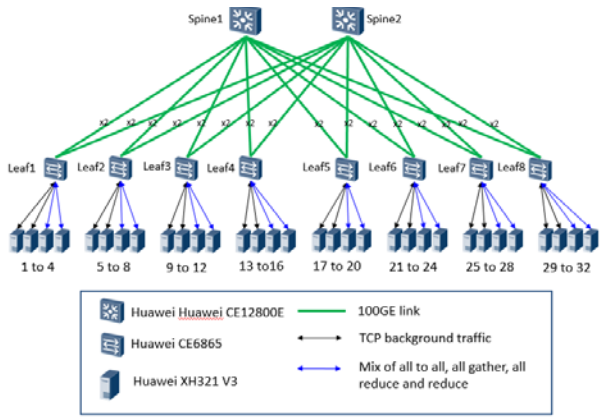

Figure 5: Mixed HPC/TCP Test Scenario

For example, AllToAll (524,288 Bytes) reached an average 0.86 s latency over 16 servers; the same test configuration had taken 3.11 s latency over 32 servers. This was expected, as a 16-server environment requires much less data exchange as a 32-server environment for the AllToAll environment. In any case, this did not affect the test methodology itself — to observe differences between dynamic ECN disabled and enabled.

Size [Bytes]	Average Latency		Delta	%
	Without dynamic ECN	With dynamic ECN		
AllToAll				
2	91.6 μs	91.5 μs	0.1 μs	<0.1
1 048 576	2 119 ms	1 667 ms	451 ms	21.3
AllReduce				
8	28.7 μs	28.7 μs	<0.1	<0.1
4 194 304	120 ms	103 ms	17 ms	14.3
AllGather				
1	31.2 μs	31.2 μs	<0.1	<0.1
1 048 576	2 757 ms	2 037 ms	720 ms	26.1
Gather				
2	0.8 μs	0.8 μs	<0.1	<0.1
1 048 576	5 181 ms	5 167 ms	14 ms	0.3
Reduce				
32	1.9 μs	1.8 μs	0.1 μs	0.4
262 144	194.0 μs	193.6 μs	0.4 μs	0.2

Table 4: IMB Latency Results Overview (When Running TCP Background Traffic)

During the benchmarking run, we observed up to 264 Gbit/s traffic (measured via switch CLI) over 2 x 8 x 100GbE links between spines and leaves. No loss was shown in the drop counter; Intel MPI benchmark traffic entered a high-priority queue while TCP traffic got assigned to a low-priority queue on each leaf switch.

The overall average latency decreased as expected when dynamic ECN was enabled.

Convergency

We tested network convergence to verify the ability of the system under test to recover from network failures.

We pulled out the cables between Spine 1 and Leaf 1 during an Intel MPI benchmark run. We measured the latency values, comparing them to the normal latency measured in previous test cases. We expected that RoCEv2 traffic would continue to operate after the failover.

Each server in a group of four servers connected over one leaf switch ran a different Intel MPI benchmark. This was to find out the overall impact of convergence on different RoCEv2 flows. In addition, TCP traffic was sent via iPERF as well, as described in the previous test case.

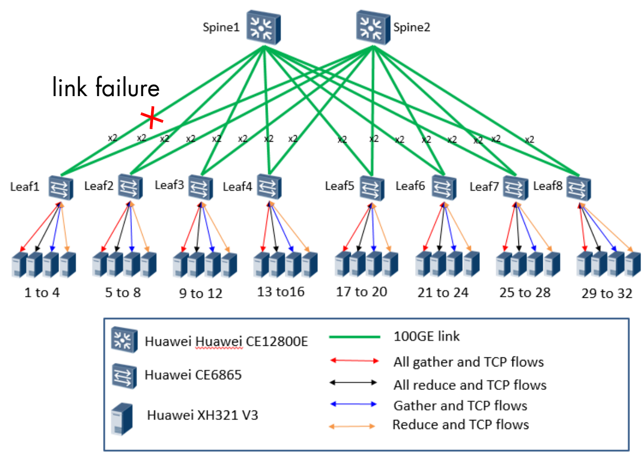

Figure 6: Convergency Topology

Initially, we generated IPv4 flows at 10 Gbit/s to measure the out of service time. However, the IPv4 traffic showed loss once it has been started with the test traffic together. This was expected: PFC was enabled on the same servers running test traffic; it dynamically adjusted the bandwidth for test traffic to avoid loss. While the IPv4 traffic kept at a constant rate, however under heavy congestion, it could not concur at certain point and showed loss. Since non-PFC application was not focus of the test, we removed the traffic generator from the test.

We specifically picked out AllToAll results from the four groups of benchmarks that were running in parallel, and show the impact of switching over on the average latency (see Figure 7) during link fail over.

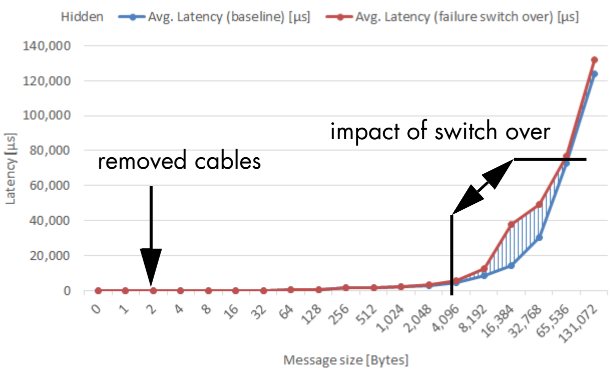

Figure 7: Avg. Latency During Fail Over — AllToAll

After removal of the cables, the latency remained closely tied to the baseline delay until having completed the small message size operations. Subsequently it increased, showing in the results graph (see Figure 7 above) at large-byte operations, indicating the impact of the switching process. It lasted for a few big message sizes then returned back to normal, indicating of the end of the switch over. The last measurements points of very large message sizes did not exactly overlap with the baseline because two links had been removed from the switch. The same computational effort was kept with fewer resources, in effect reducing the network latency.

Summary

- HPC continued to operate well after failover and recovery from link failure.

Distributed File System (DFS) Latency

As a second use case, we measured the latency of remote file access in the data center. The storage latency defines the time taken to complete a storage operation. The common performance indicator is the number of Input/Output Operation per Second (IOPS) which are directly influenced by the latency. The DFS service latency includes a variety of factors across data center including server processing times, switching delay and network delay.

To measure the impact of dynamic ECN (Explicit Congestion Notification) on latency, we evaluated the storage performance in two scenarios with and without dynamic ECN. Each test was repeated three times. We compared the values and calculated the delta.

Huawei set up a separate test bed following the same spine-leaf architecture as before. The leaf and spine nodes were implemented with Huawei CE6865 and Huawei CE8850-64CQ. The storage unit of the data center included four Huawei XH321 V3 servers, each connected with one 25 GbE port. This resulted a total of 100 Gbit/s physical bandwidth from the storage servers; only two Huawei ES3600P V3 SSDs (Solid State Drives) per server were needed to saturate this bandwidth. Huawei informed us that the SSDs had 1.7 GBytes/s write-speed and 2.9 GBytes/s read-speed.

For the computation unit, 11 compute nodes were set up by the Huawei team, each connected to the leaf nodes via one 25 GbE interface.

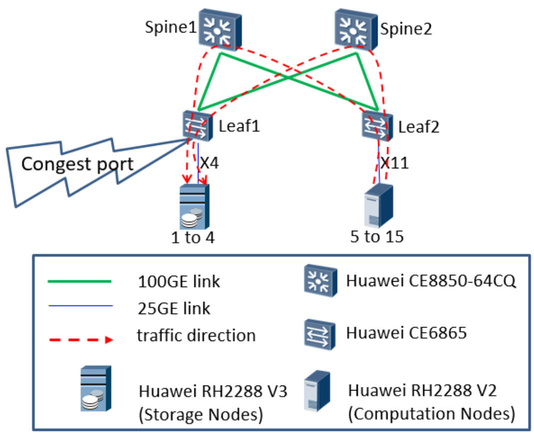

Figure 8: Storage Performance Test Bed

We used the open source tool FIO (Flexible I/O Tester, fio-2.1.10) to emulate desired I/O workload under Linux. This tool runs simple, non-optimized file access to saturate the available bandwidth of 25 Gbit/s traffic on each server. Five key parameters characterize the traffic model over network:

Parameter	Tested Values
Block size	4 KB or 4 MB
I/O depth	1, 32 and 128
Run time	100 s
Target device	1 (to emulate central storage) and 4 (to emulate distributed storage access)
I/O type	random write

Table 5: FIO Test Setup

The I/O type was chosen as random write because random write access actually created network congestion across the 11:4 nodes as required for the test. Read access would just have caused SSD storage processing and had less impact on the network performance.

The first baseline tests (dynamic ECN disabled) reflected the default network state of the Huawei AI fabric architecture. Under this precondition, the server NIC had been configured with ROCEv2 (RDMA over Converged Ethernet) to accommodate the speed caused by congestion, and the AI fabric also set with PFC to create queues. These settings smooth the traffic state of the AI Fabric to adapt the bandwidth generated by FIO to the available physical bandwidth. We recorded the storage performance including IOPS, bandwidth, latency and tail latency.

Then we enabled dynamic ECN. During the test, we observed that the AI fabric triggered ECN messages for buffered packets as expected.

We observed up to 84 ms average delay reduction when dynamic ECN was enabled (I/O Depth 128), equal to 14 % of the 599 ms baseline delay without dynamic ECN. The corresponding IOPS, performed by 11 computation nodes, reached up to 248.8 per node, with up to 1.0 GBytes/s write throughput. All other I/O Depths showed performance optimization as well; the latency results with I/O Depth of 1 improved by 20 % and the results with I/O Depth of 32 improved by 13 %.

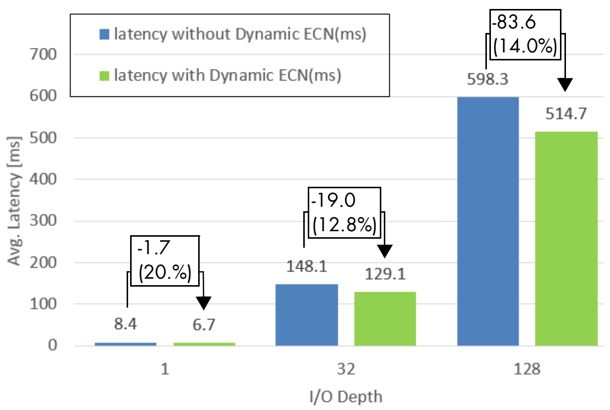

Figure 9: FIO Average Delay (ms), 4M Random Write, 4 Target Devices

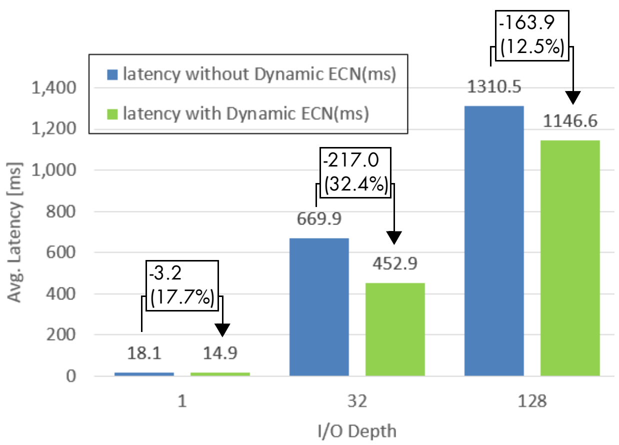

Figure 10: Average Tail Delay (ms), 4M Random Write, 4 Target Devices

The tail delays improved as well for all I/O Depths. For the depth of 32, there was a 32 % latency improvement compared with the baseline tail delay without dynamic ECN. Tests with I/O Depth 1 and 128 showed tail latency reductions between 3 ms and 164 ms.

I/O Depth	Average delay [μs]		Delta [μs]	%
	Without dynamic ECN	With dynamic ECN		
4 target devices, 4 Kbytes				
1	37.7	36.7	1.0	2.7
32	215.4	203.9	11.5	5.3
128	820.3	816.3	4.0	0.5
1 target device, 4 Kbytes				
1	38.5	38.5	0.0	0.0
32	588.2	546.7	41.5	7.1
128	2 371.8	2 138.8	233.0	9.8
1 target device, 4 Mbytes				
1	18.0	16.6	1.3	7.2
32	600.8	511.0	89.7	15.0
128	2 649.3	2 416.8	232.5	8.7

Table 6: FIO Latency Results Overview

I/O Depth	Avg. tail delay [μs]		Delta [μs]	%
	Without dynamic ECN	With dynamic ECN		
4 target devices, 4 Kbytes				
1	42.7	42.4	0.3	0.7
32	262.1	220.9	41.2	15.7
128	1 006.9	847.8	159.1	15.8
1 target device, 4 Mbytes				
32	883.6	717.3	166.3	18.8
128	2 347.5	2 034.1	232.5	8.8

Table 7: FIO Tail Latency Results Overview

I/O Depth	IOPS		Delta [Number of IOPS]	%
	Without dynamic ECN	With dynamic ECN		
4 target devices, 4 Mbytes				
1	118	148	29	19.9
32	216	247	30	12.4
128	215	248	33	13.5
1 target device, 4 Mbytes				
1	55	59	4	7.4
32	53	62	8	14.1
128	54	62	8	13.0
4 target devices, 4 Kbytes				
1	26 100	26 500	400	1.4
32	152 600	164 500	11 900	7.2
128	160 000	162 600	2 600	1.6
1 target device, 4 Kbytes				
1	25 365	25 374	9	<0.1
32	54 276	58 504	4 227	7.2
128	54 131	59 913	5 782	9.7

Table 8: IOPS Results Overview

Furthermore, we evaluated the latency behavior when TCP traffic was added. The latency reduced as expected when dynamic ECN was enabled.

Summary

- The latency development was linear. The greater the message size (and I/O depth), the higher the latency value. This behavior did not change when dynamic ECN was enabled as expected.
- Latency reduced significantly for large I/O depth of big message sizes.
- Overall latency reduced when dynamic ECN was enabled.

Conclusion

The Huawei's AI Fabric Ultra-High-Speed Ethernet passed EANTC's high performance data center benchmarking tests. It supported message passing and file storage use cases without packet loss and handled delay-sensitive applications efficiently. In both the HPC as well as DFS benchmarking tests, the dynamic ECN-enabled Huawei AI fabric handled a mix of data center traffic successfully and reduced average network latency as expected. In addition, the solution proved to be resilient against network link failures.

About EANTC

EANTC (European Advanced Networking Test Center) is internationally recognized as one of the world's leading independent test centers for telecommunication technologies.

Based in Berlin, the company offers vendor-neutral consultancy and realistic, reproducible high-quality testing services since 1991. Customers include leading network equipment manufacturers, tier 1 service providers, large enterprises and governments worldwide. EANTC's Proof of Concept, acceptance tests and network audits cover established and next-generation fixed and mobile network technologies.

EANTC AG
Salzufer 14, 10587 Berlin, Germany
info@eantc.com, http://www.eantc.com/
[v1.1 20180716]