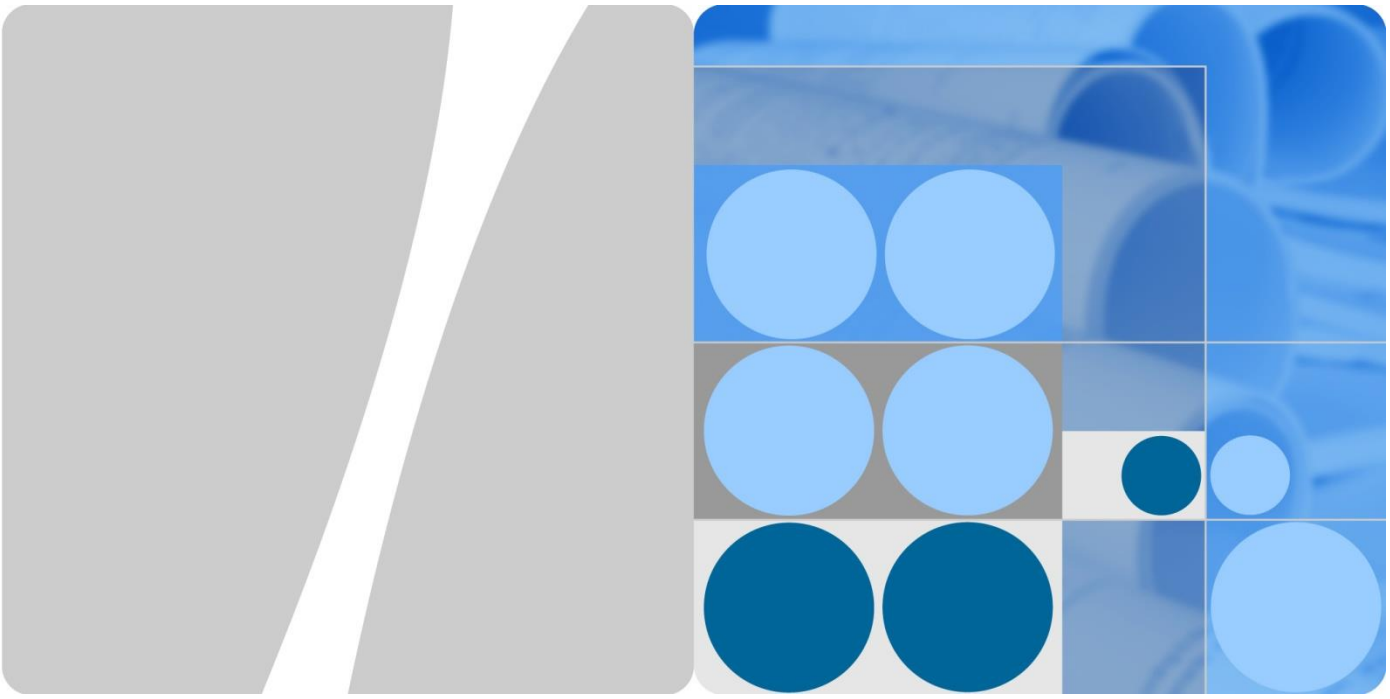


CloudEngine AI Fabric Technology White Paper

Issue 01

Date 2018-07-10

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.huawei.com

Email: support@huawei.com

Contents

1 Current Situation of AI Data Centers

Traditional data centers use Ethernet technology to form a multi-hop symmetric network architecture and use the TCP/IP network protocol stack to transmit data. As the data center traffic volume increases, applications such as Artificial Intelligence (AI) and Virtual Reality (VR) are constantly emerging. This makes it increasingly difficult for traditional Ethernet technology carrying TCP/IP traffic to meet requirements. Therefore, current AI data center networks (DCNs) are starting to shift towards high-speed, reliable, and stable link technologies and network protocols.

InfiniBand (IB) is a computer-networking communications standard used in high-performance computing. It features high throughput and low latency. It is used for data interconnect both among and within computers. InfiniBand is also used as a direct or switched interconnect between servers and storage systems, as well as an interconnect between storage systems. InfiniBand uses a switched fabric topology, as opposed to the earlier method that used Ethernet as a shared medium. With InfiniBand, all transmissions begin or end at a channel adapter. To be specific, each processor contains a host channel adapter (HCA) and each peripheral has a target channel adapter (TCA). These adapters can also exchange information for security or quality of service (QoS). Unlike the traditional TCP/IP protocol stack, InfiniBand has its own network layer and transport layer protocols. Also, it does not have standard network programming interfaces. Instead, its programming interface is the Verbs interface released by OpenFabrics Alliance, and this interface is not compatible with the socket interface. InfiniBand is the most commonly used interconnection technology of supercomputers, due to its high transmission rate.

Table 1-1 InfiniBand transmission performance

Item	SDR	DDR	QDR	FDR-10	FDR	EDR	HDR
Signaling rate (Gbit/s)	2.5	5	10	10.3125	14.0625	25	50
Theoretical effective throughput (Gbit/s), per 1x	2	4	8	10	13.64	24.24	50
Speeds for 4x links (Gbit/s)	8	16	32	40	54.54	96.97	200

Item	SDR	DDR	QDR	FDR-10	FDR	EDR	HDR
Speeds for 12x links (Gbit/s)	24	48	96	120	163.64	290.91	600
Adapter latency (microseconds)	5	2.5	1.3	0.7	0.7	0.5	0.5

The main InfiniBand device manufacturers include Mellanox and Intel.

RoCEv1/RoCEv2: Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE) technology is developed based on InfiniBand technology, which has high speed and high resource utilization. However, its entire system is incompatible with the traditional TCP/IP Ethernet network and requires special NICs and switches. Based on existing Ethernet standards, RoCE uses InfiniBand standards to implement highly-efficient transmission at the network layer (RoCEv1) and transport layer (RoCEv2). Only RoCE-enabled NICs are required, and network-side hardware does not need to be changed. RoCE implements high-speed and efficient transmission in the Ethernet environment. At present, the number one RoCE NIC manufacturer is Mellanox.

Omni-Path: Omni-Path is a high-speed interconnection technology proposed by Intel for High Performance Computing (HPC). It has a transmission rate of 100 Gbit/s and a lower latency than the Infiniband, and uses dedicated network devices. The number one manufacturer is Intel.

Table 1-2 Comparisons of different network technologies

Technology	Interface Rates (Gbit/s)	Latency	Key Technology	Advantage	Disadvantage
TCP/IP over Ethernet	10, 25, 40, 50, 56, 100, or 200	500-1000 ns	TCP/IP Socket programming interface	Wide application scope, low price, and good compatibility	Low network usage, poor average performance, and unstable link transmission rate
Infiniband	40, 56, 100, or 200	300-500 ns	InfiniBand network protocol and architecture Verbs programming interface	Good performance	Large-scale networks not supported, and specific NICs and switches required
RoCE/RoCEv2	40, 56, 100, or 200	300-500 ns	InfiniBand network layer or transport layer and Ethernet link layer Verbs programming	Compatibility with traditional Ethernet technologies, cost-effectiveness, and good performance	Specific NICs required

Technology	Interface Rates (Gbit/s)	Latency	Key Technology	Advantage	Disadvantage
			interface		
Omni-Path	100	100 ns	OPA network architecture Verbs programming interface	Good performance	Single manufacturer and specific NICs and switches required

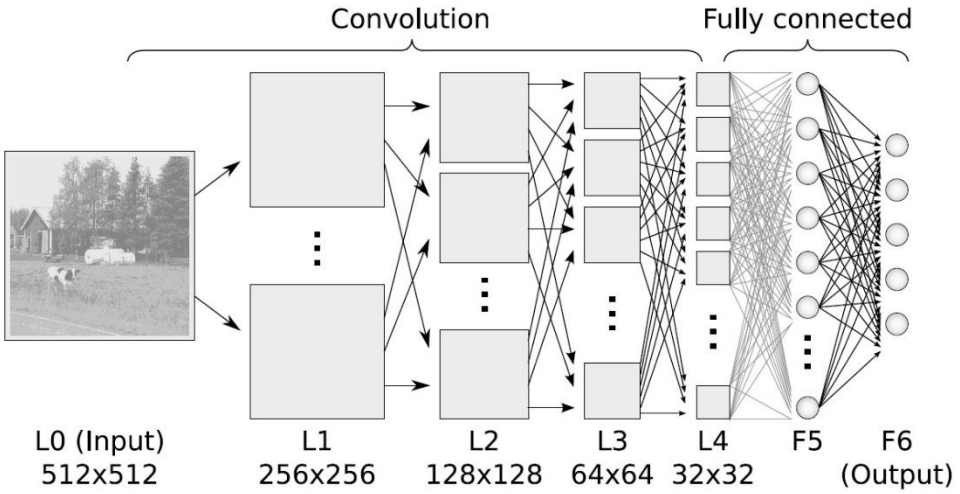
In 2016, of the US public cloud service providers Amazon, Google, and Microsoft, only Microsoft Azure used InfiniBand technology to provide high-speed network interconnection. However, with the increasing number of AI applications, this situation is changing greatly. Facebook's latest machine learning open-source platform—Big Basin—uses 100 Gbit/s InfiniBand for network interconnection. Baidu also uses 100 Gbit/s InfiniBand network devices in its machine learning environment. In response to the constant emergence of newly developed dedicated AI computing chips, the expansion of computing scale, and the growing maturity mainstream machine learning and development platforms, a growing number of AI data centers are using high-performance network solutions other than TCP/IP over Ethernet.

2 New Requirements for DCNs by AI Technology

2.1 Characteristics of AI Computing

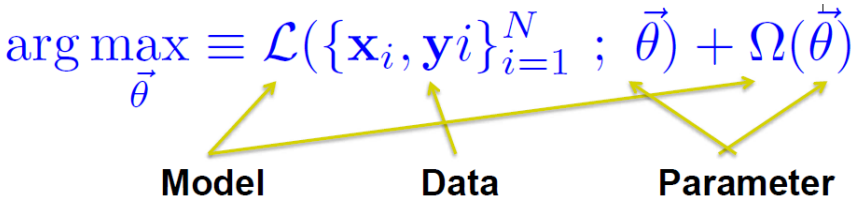
Traditional data center services (web, video, and file storage) are transaction-based and the calculation results are deterministic. For such tasks, there is no correlation or dependency between single calculation and network communication, and the occurrence time and duration of the entire calculation and communication are random. AI computing is based on target optimization and iterative convergence is required in the computing process, which causes high spatial correlation in the computing process of AI services and temporally similar communication modes.

Figure 2-1 Iterative machine learning network model



A typical AI algorithm refers to an optimization process for a target. The computing scale and features mainly involve models, input data, and weight parameters.

Figure 2-2 Machine learning algorithm



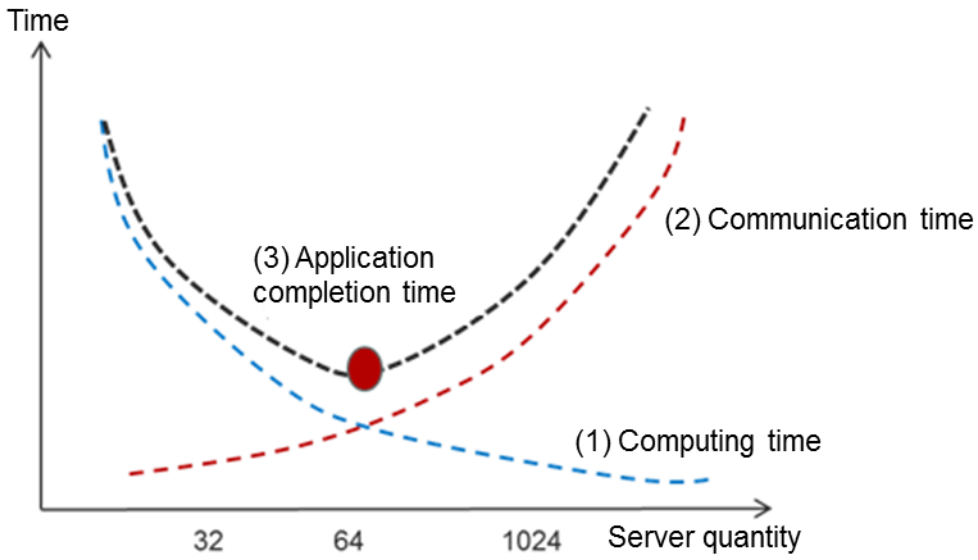
--Data source: Eric Xing, CMU

To solve the Big Data problem, the computing model and input data need to be large (for a 100 MB node, the AI model for 10K rules requires more than 4 TB memory), for which a single server cannot provide enough storage capacity. In addition, because the computing time needs to be shortened and increasingly concurrent AI computing of multiple nodes is required, DCNs must be used to perform large-scale and concurrent distributed AI computing.

2.2 Requirements of Distributed AI Computing for DCNs

As the number of AI algorithms and AI applications continue to increase, and the distributed AI computing architecture emerges, AI computing has become implemented on a large scale. To ensure enough interaction takes place between such distributed information, there are more stringent requirements regarding communication volume and performance. Facebook recently tested the distributed machine learning platform Caffe2, in which the latest multi-GPU servers are used for parallel acceleration. In the test, computing tasks on eight servers resulted in insufficient resources on the 100 Gbit/s InfiniBand network. As a result, it proved difficult to achieve linear computing acceleration of multiple nodes. The network performance greatly restricts horizontal extension of the AI system.

Figure 2-3 Balance between communication time and computing time

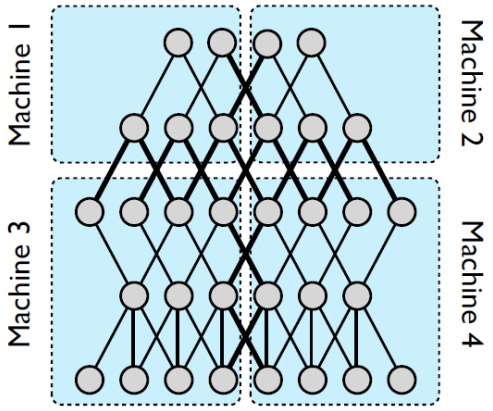


2.2.1 Implementation of Distributed AI Computing

Distributed AI computing has the following two modes: model parallel computing and data parallel computing.

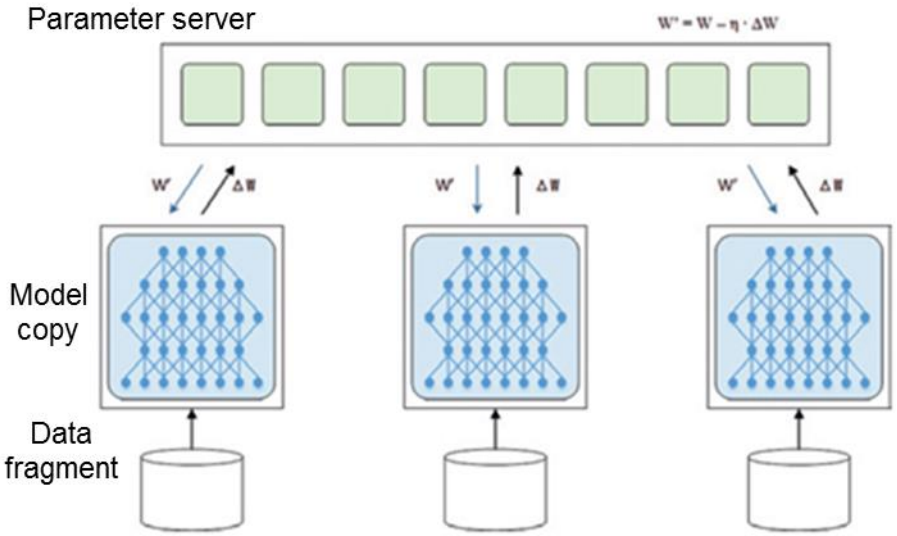
For model parallel computing, each node computes one part of the algorithm. After computing is complete, all data fragmented across models needs to be transferred to other nodes, as shown in Figure xx.

Figure 2-4 Model parallel training



For parallel data computing, each node loads the entire AI algorithm model. Multiple nodes can calculate the same model at the same time, but only part of the input data is input to each node. When a node completes a round of calculation, all relevant nodes need to aggregate updated information about obtained weight parameters, and then obtain the corresponding globally updated data. Each weight parameter update requires that all nodes upload and obtain the information synchronously.

Figure 2-5 Data parallel training



2.2.2 Communication Requirements of Distributed AI Data Centers

The increasing scale of distributed AI computing and improving AI computing hardware capabilities pose high requirements on basic performance of AI DCNs.

High Bandwidth and Low Static Latency

The test results of several typical AI computing models indicate that distributed AI computing is high-performance computing with centralized computing and communication. For some AI models, 100 MB to 200 MB of traffic is generated every 20 ms on the latest GPU, which is equivalent to instantaneous burst traffic of up to 80 Gbit/s. Therefore, AI data centers must ensure that the link bandwidth is as high as possible and the link latency is as low as possible. Otherwise, the communication time is more than the calculation time, which severely affects the concurrency and completion time of distributed AI computing.

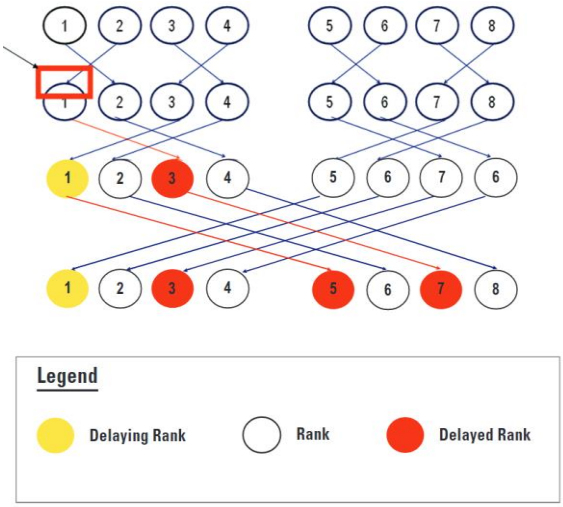
Table 2-1 AI computing (data source: benchmarking state of the art deep learning software tools)

AI Computing Model	Communication Volume (16-Bit Precision)	Computing Capacity (FLOPS)	Single Iteration Time (Seconds)	
			Single GPU (K80, V100)	CPU (32-Thread)
AlexNet	120 MB	63 millions	0.07, < 0.02	1
ResNet50	1.5 MB	3.8 billions	0.5, < 0.1	8
FCN-S	236 MB	55 millions	0.07, < 0.02	1

Low dynamic latency and zero packet loss

The high-bandwidth and low-latency DCN with only physical links cannot meet requirements of large-scale and highly concurrent AI applications. In the iteration process of distributed AI computing, a large amount of burst traffic is generated within milliseconds. In addition, because a parameter server (PS) architecture is used to update parameter weights of the new model for data parallelization, the Incast traffic model at a fixed time is easily formed. In this case, packet loss, congestion, and load imbalance occur on the network. As a result, the Flow Completion Time (FCT) of some data flows is too long. Distributed AI computing is synchronous. If few flows are delayed, more computing processes are affected. Consequently, the completion time of the entire application is delayed.

Single-point delay causes the overall performance to deteriorate (data source: Mellanox)

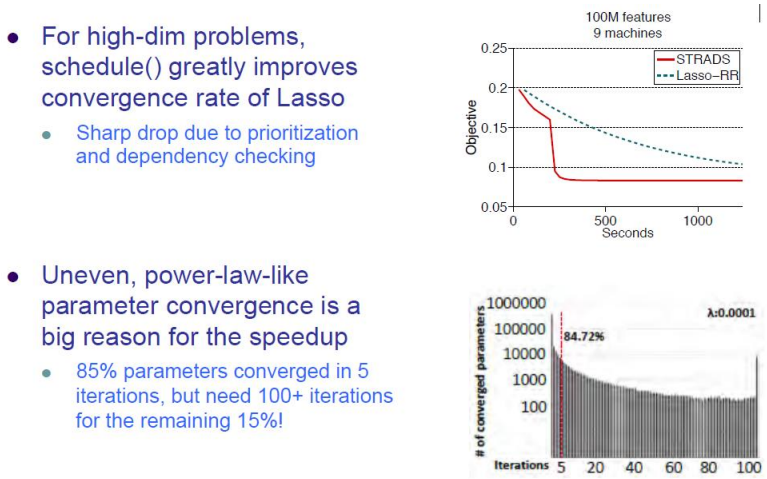


Therefore, the fine-grained design is required for the DCN that is oriented towards large-scale distributed AI. This prevents FCT increase caused by packet loss, congestion, and Incast. In addition, a dynamic low latency and stable bandwidth for communication must be ensured.

JCT optimization

Based on guaranteed E2E data flow performance, the distributed AI DCN needs to effectively optimize the job completion time (JCT) for AI computing. In the distributed AI parallel computing process based on iterative convergence, over 80% of computing can converge after a few iterations, while the remaining computing requires a large number of iterations. The practice proves that communication scheduling required by distributed AI computing can greatly improve the calculation convergence duration.

Figure 2-6 Importance of JCT optimization



Spatial correlation of distributed AI computing causes transmission of network data traffic to be correlated. In terms of space, the optimal policy for synchronously computing multiple data flow groups (co-flow) is to minimize the transmission completion time of the entire flow group. In terms of time, due to the sequence of multiple iterations, data flows in earlier iterations need to be completed earlier, so that subsequent relevant computing processes can start as soon as possible. To achieve optimal parallelization between computing and

communication, an AI DCN needs to optimize the transmission of multiple flows to optimize the JCT.

2.3 Major Problems of Current DCNs

Compared with a traditional DCN, the AI DCN has better performance indicators such as high bandwidth, low latency, and low packet loss ratio. However, there are still some problems in the design of fine-grained single-point control, network-wide control, and software and hardware integration. The AI DCN cannot meet requirements for parallel computing and communication of AI applications in the future.

There is no effective flow control in the Incast scenario: On the distributed AI DCN that uses the parameter server mode, high burst traffic in Incast mode is common. However, common flow control mechanisms such as Priority-based Flow Control (PFC) on DCNs have problems such as header blocking, being unable to ensure zero packet loss in most cases, and possibly causing long delays in processing some traffic. As a result, the overall JCT is very long.

Passive congestion control does not offer quick, effective responses: Burst traffic on distributed AI DCNs is heavy and requires quick responses. However, the congestion control mechanism based on Explicit Congestion Notification (ECN) on DCNs has a long delay. This leads to issues such as low throughput and traffic bursts. In addition, this mechanism cannot prevent congestion when heavy burst traffic lasts for milliseconds. InfiniBand-based proactive traffic scheduling technology provides link-level traffic control and congestion control, but cannot meet future requirements for large-scale parallel computing of AI DCNs.

There is no fast or effective load balancing mechanism: On a large-scale AI network, the concurrent computing of multiple AI applications may cause a large amount of Incast traffic. For the AI computing model, traffic models are uncertain, leading to unbalanced traffic on network paths. Flow-based load balancing policies cannot load balance high-bandwidth and continuous heavy traffic.

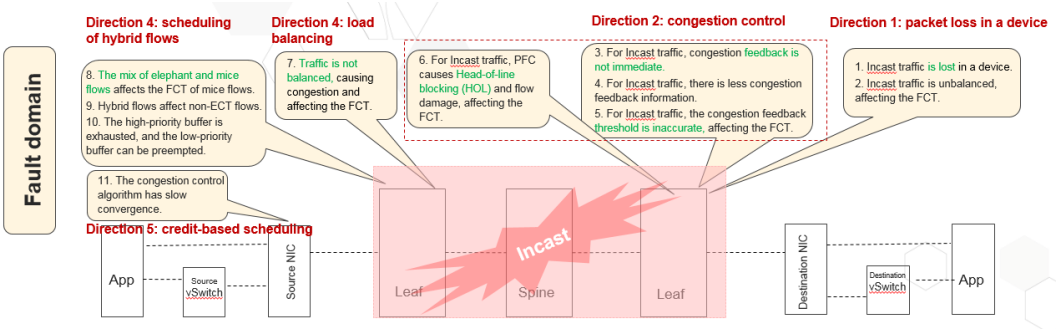
Elephant and mice flows centrally scheduled without being differentiated: Distributed AI features centralized computing and high throughput, so most data traffic on AI DCNs is comprised of elephant flows (10 MB to 100 MB) and control traffic is comprised of mice flows (several KB). Frequent and long-lasting elephant flows seriously affect control data transmission of AI applications, prolonging the application completion time. However, few network devices and NICs on DCNs can differentiate elephant flows on the data plane and mice flows on the control plane. This prevents mice flows from effectively utilizing bandwidth due to buffer, HoL blocking, and lack of high-priority scheduling.

3 AI Fabric

3.1 AI Fabric Solution

The AI Fabric DCN has the following features:

- Basic network features: high bandwidth, low delay, no packet loss, and no blocking
- System-level features: in-network computing for communication processing, transmission optimization by AI application detection, and parallel scheduling mechanism for high-performance computing and communication



3.2 AI Fabric DCN Technology Directions

To address the preceding issues, AI Fabric technologies are developing in the following directions.

3.2.1 Incast Traffic Scheduling Without Packet Loss or Congestion

Currently, the inbound and outbound buffer thresholds on each interface of the forwarding chip are configured to ensure basic scheduling of interface queues in the common traffic model. The threshold parameters are statically configured. This means they cannot adapt to different traffic models or guarantee zero packet loss in the Incast traffic model with high concurrency.

Large-scale deployment of high-performance computing and distributed storage applications on a DCN requires the DCN to guarantee zero packet loss. These distributed applications use the N:1 Incast traffic model. As these applications are deployed on a larger scale, the Incast

degree increases accordingly. As a result, the packet buffer of switches is increased and packet loss occurs due to buffer overflow.

The buffer settings for interface queues of the forwarding chip include the uplink and downlink buffer thresholds.

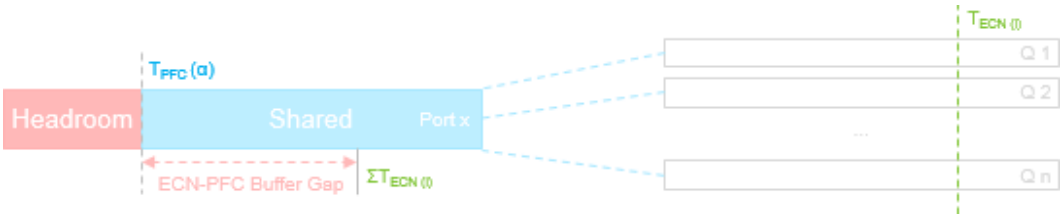
1. The uplink buffer threshold is used to set the guaranteed buffer, shared buffer, and headroom buffer for the inbound interface. The guaranteed buffer ensures that the inbound interface can receive a packet with the size of the MTU anytime. This prevents service traffic interruption caused by a failure to preempt buffer resources. For the lossless priority, PFC needs to be enabled. The shared buffer is the PFC-XOFF threshold. When PFC is triggered, the local device receives PFC frames from the remote end and stops sending packets. In this case, the headroom buffer is used to cache the packets that have been sent by the local end and traveling packets on links in the preceding process, preventing these packets from being discarded.
2. The downlink buffer threshold is used to set the guaranteed buffer and shared buffer for queues on the outbound interface. The shared buffer is the maximum buffer that can be preempted by the outbound interface.

The physical buffer of the forwarding chip does not require the uplink or downlink buffer thresholds, which are configured for interface queues of the forwarding chip for buffer management in the inbound and outbound directions. A packet is allowed to enter the physical buffer of the forwarding chip only when both the inbound and outbound buffer thresholds are not exceeded; otherwise, the packet is discarded.

That is, inbound and outbound buffer thresholds on an interface must be within the allowed range so that packets are not discarded. Inbound and outbound buffer thresholds differ between different traffic models and scenarios.

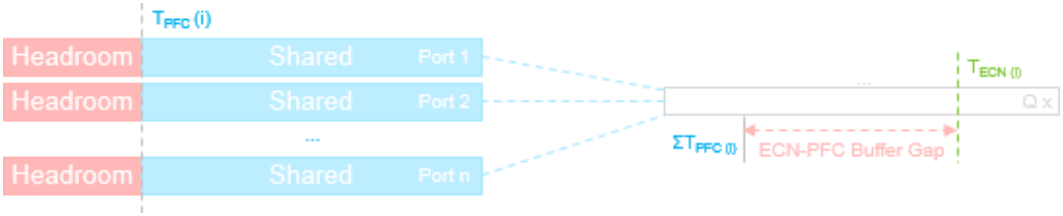
In the 1:N scenario, to ensure that no packet loss occurs and the throughput is not affected, the following buffer configuration requirements must be met:

1. $T_{PFC} > \Sigma T_{ECN}(i)$: Before ECN is triggered for all downlink queues, PFC is not triggered for the uplink queue.
2. $\Sigma T_{Q\text{-}BUF}(i) > (T_{PFC} + T_{HDRM})$: Before PFC is triggered for the uplink queue, no packet loss occurs in any downlink queue.

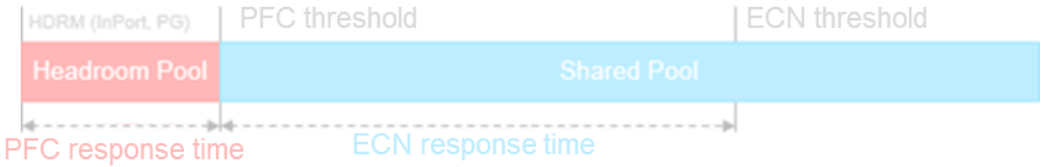


In the N:1 scenario, to ensure that no packet loss occurs and the throughput is not affected, the following buffer configuration requirements must be met:

1. $\Sigma T_{PFC}(i) > T_{ECN}$: Before ECN is triggered for all downlink queues, PFC is triggered for the uplink queue.
2. $T_{Q\text{-}BUF} > \Sigma(T_{PFC}(i) + T_{HDRM}(i))$: Before uplink PFC is triggered, no packet loss occurs in any downlink queue.



Based on the mapping from uplink and downlink buffer thresholds to the physical buffer, Q-BUF involves the following buffer allocation: {Headroom, PFC-Threshold, ECN-Threshold}.



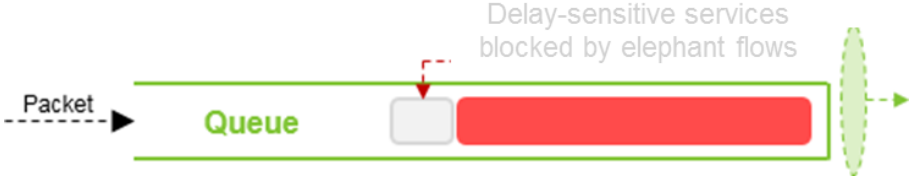
For the lossless priority of the inbound interface, the headroom buffer is required for zero packet loss when PFC is triggered. The headroom buffer affects the link rate and link distance (that is, link latency) and its value is fixed. The PFC-XOFF threshold for the lossless priority of the inbound interface and the shared buffer threshold for queues on the outbound interface affect the physical buffer of the forwarding chip and traffic model. When the traffic model changes, the buffer thresholds must be dynamically adjusted to ensure no packet loss.

The buffer between the ECN threshold and the PFC threshold is described as follows: Between the time an ECN flag is marked and the time the source end reduces the rate, PFC is not triggered by traffic. In this case, ECN technology can preferentially resolve congestion to avoid or reduce PFC triggering. The buffer between the ECN threshold and PFC threshold affects the frequency of PFC triggering.

For the N:1 traffic model of distributed and parallel data center applications, the buffer threshold configuration of the forwarding chip needs to be adjusted. The buffer threshold configuration can be adjusted dynamically and immediately when the traffic model (burst degree: N) changes. Ensure that no packet loss occurs on switches and the throughput is not affected.

3.2.2 Differentiated Scheduling of Elephant and Mice Flows

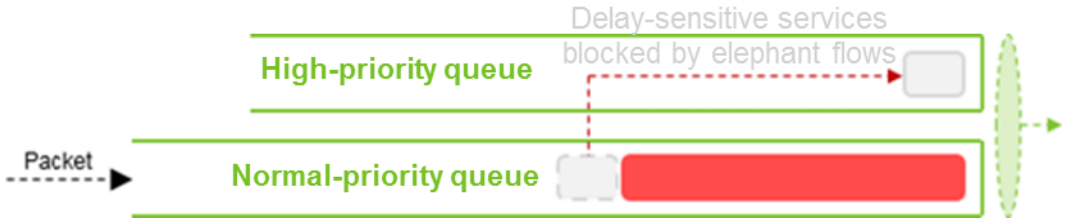
The packets that enter a queue must be dequeued in order. When elephant and mice flows are transmitted together, elephant flows occupy the queue when traffic congestion occurs. The subsequent mice flows may be discarded because they cannot enter the queue. Even if the mice flows can enter the queue, they may be delayed for a long time because the elephant flows occupy much of the queue. When a queue is blocked by elephant flows, the FCT of the delay-sensitive mice flows is increased greatly.



A technical solution is required to preferentially schedule packets in mice flows, so that the latency of mice flows is not affected by elephant flows, ensuring the FCT of mice flows.

The solution needs to identify elephant and mice flows, and then reduce the priority of elephant flows so that mice flows have a higher priority. The solution is ineffective for the mice flows that have been blocked before elephant flows are identified.

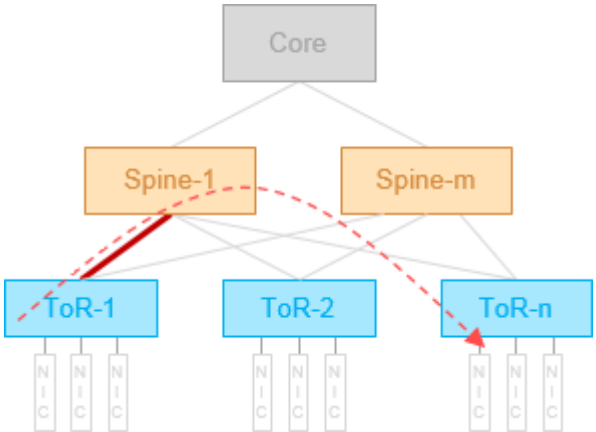
For this reason, several initial packets of service flows need to be placed in a high-priority queue so that they can be preferentially scheduled and sent. After the solution identifies elephant flows among the service flows, it reduces the priority of the elephant flows. In this way, mice flows are no longer delayed due to being blocked by elephant flows.



To identify elephant and mice flows, the forwarding chip needs to learn and store each flow in real time through the flow table (for example, based on the IP quintuple), and to collect traffic statistics on each flow based on the flow rate, byte, and lifetime. After identifying an elephant flow, the forwarding chip dynamically decreases the scheduling priority of the flow.

3.2.3 Load Balancing of Network Traffic

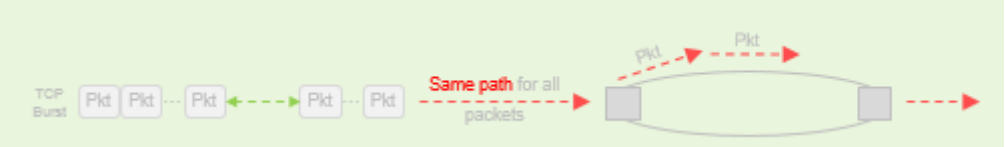
On DCNs, service traffic is usually transmitted through multiple equal-cost paths. To ensure that service traffic can be evenly distributed to physical links and no packet mis-sequencing occurs, the static hash algorithm based on characteristic fields of packets is used. However, the static hash algorithm may cause a hash imbalance. That is, a physical link may be overloaded or, in the worst case, packets may be discarded due to congestion even if other physical links are only lightly loaded. In this situation, the bandwidth usage is low and the application performance (FCT and throughput) is affected.



On DCNs, for example, when there are high-performance computing and distributed storage applications, there are delay-sensitive mice flows in a short time and elephant flows with high bandwidth, sensitive throughput, and long lifetime. Generally, fewer than 10% of flows are elephant flows, which occupy more than 80% of traffic. When a large number of flows are load balanced in static hash mode, the number of flows on each link is almost the same, but the total bandwidth occupied by the flows on each link is unbalanced. This causes the link

load to be unbalanced. In addition, when elephant and mice flows are statically hashed to the same path, elephant flows will preempt the bandwidth and mice flows are blocked, seriously degrading the application performance.

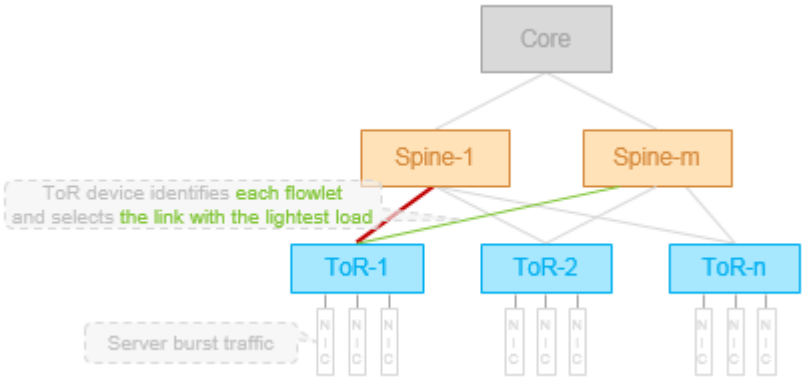
Flow-based load balancing is used in static hash mode. That is, to prevent packet mis-sequencing, this mode selects the same path for the same flow, regardless of the bandwidth of the flow.



However, a considerable amount of traffic on a DCN is burst traffic (for example, TCP bursts). If the time since the previous burst of traffic occurred exceeds the latency difference between paths, a route is re-selected for a new burst of traffic. This prevents packet mis-sequencing.

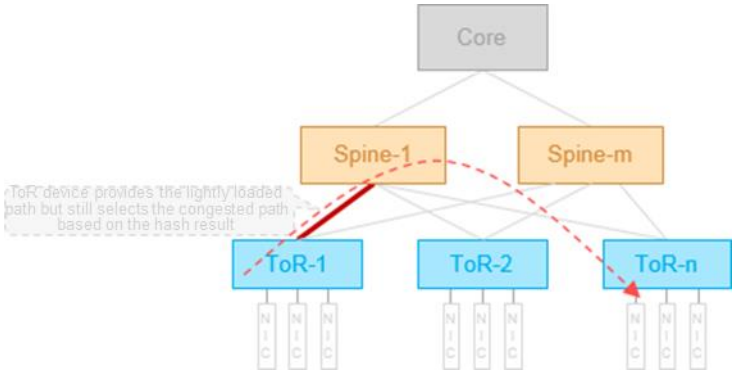
A burst of traffic is called a flowlet. The interval between packets in a flowlet is smaller than the latency difference between paths. Therefore, the same path is selected to ensure that the packets in the flowlet are transmitted in the correct order.

Elephant flows on a DCN can be identified by a flowlet. An elephant flow can be divided into several mice flows to release the bandwidth that would otherwise be occupied by the elephant flow. The route is selected by the flowlet to balance the load of each link.

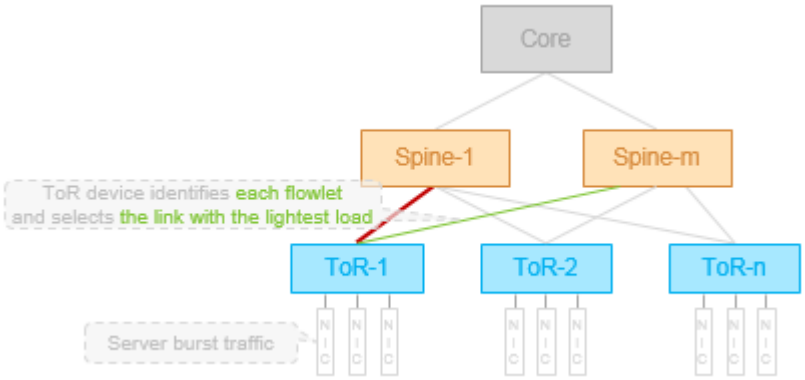


The forwarding chip needs to record the member links selected by each flowlet, so that subsequent packets entering the flowlet are still transmitted through the same link, preventing packet mis-sequencing. To achieve this, the forwarding chip is required to learn and age flowlet entries, and to record timestamps of different packets of each flow, so as to distinguish different flowlets.

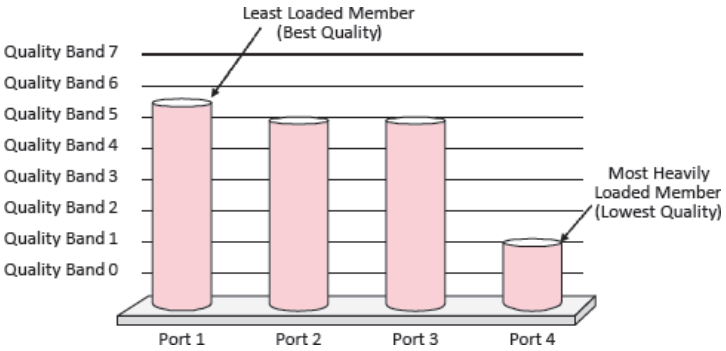
Load balancing in static hash mode performs static route selection based on characteristic fields of packets, regardless of the load on each link.



To implement load balancing based on the flowlet, consider the load of each link and select the lightly loaded link for flowlets.



To measure the congestion level of each link, the forwarding chip needs to measure the quality of each interface based on the buffer length and bandwidth usage. Packets of the flowlet are sent over the link with the best quality and least congestion.

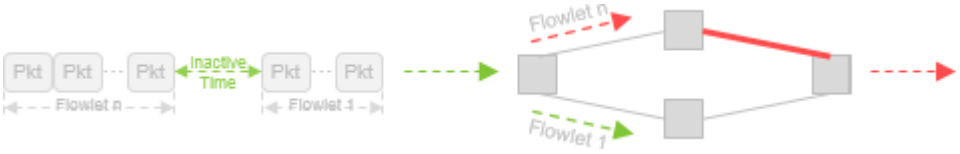


In load balancing mode, links are dynamically selected according to congestion. This ensures link usage is more balanced and the performance (FCT and throughput) of applications is improved.

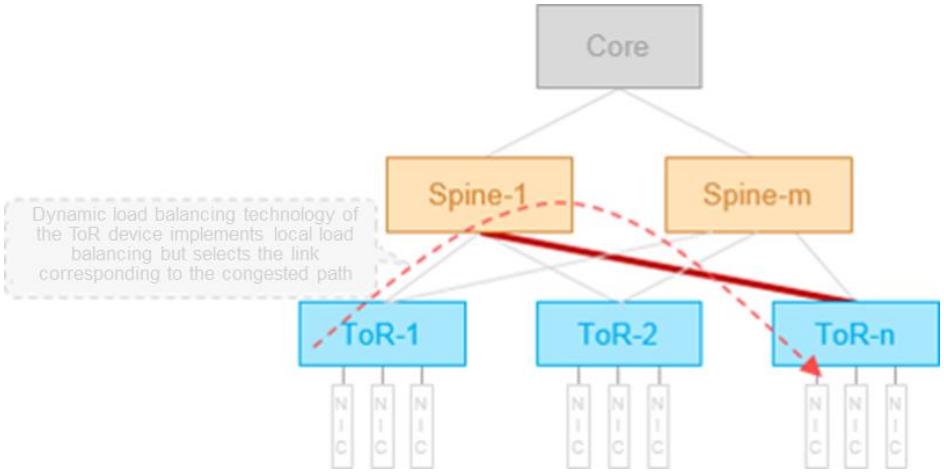
On a DCN, flows of high-performance computing applications are often mice flows but not flowlets. Elephant flows of distributed storage applications may not be divided into multiple flowlets (NICs can be used for dividing flows into flowlets). In this case, only flow-based dynamic load balancing can be implemented for traffic of these distributed applications. To

better load balance traffic, the InfiniBand Base Transport Header (IB BTH) field of RDMA traffic needs to be used in the hash algorithm to learn flows.

Flowlet-based dynamic load balancing considers only the link load of the local device, not the link load of the remote device.

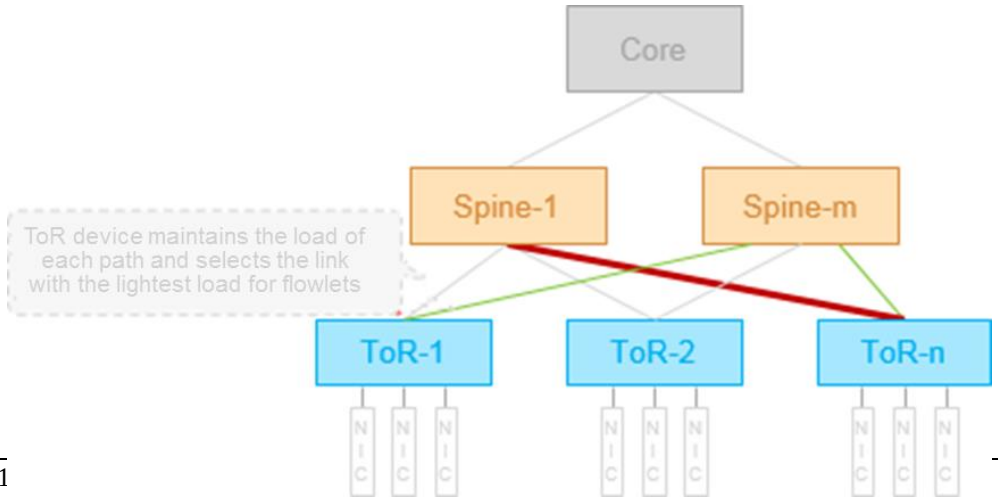


Even if a switch enabled with dynamic load balancing achieves load balancing among local links, congestion points may still exist at the remote end on the service forwarding paths. This is because the load of each path of service flows is not considered.



The switch maintains the load of paths on the entire network for service flows. During load balancing at the local end, the switch selects a local link corresponding to the path with the lightest load.

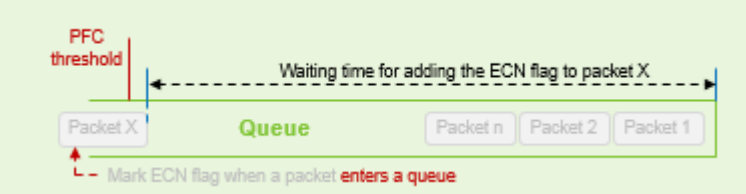
This global load balancing mechanism balances the load of paths for service flows. It achieves balanced path usage and improves the application performance (FCT and throughput).



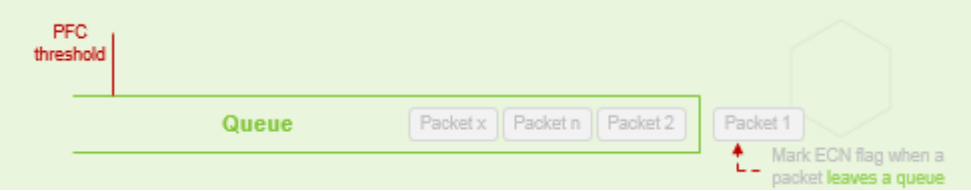
To measure the congestion status of each path, the forwarding chip needs to detect the congestion status of each device along the path in real time by sending packets, and then measures the quality of the path. The switch selects a link for a flowlet based on the quality of the path corresponding to the link rather than based on the quality of the link.

3.2.4 Congestion Control

Traditional ECN technology adds the ECN flag to a packet when it enters a queue. In this way, congestion notification is triggered when the packet leaves the queue. This mechanism creates a delay that is the period from the packet entering the queue to the packet leaving the queue. Because of this, the source cannot be immediately instructed to reduce the transmission rate to relieve congestion. If the queue is seriously congested, the device with a small buffer may take several milliseconds to send a packet and then notify the source end. As a result, the queue congestion deteriorates, and, in some cases, PFC is triggered on the entire network and traffic is stopped.

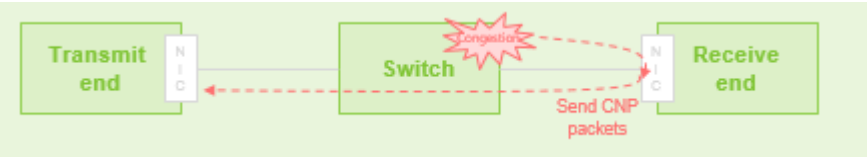


If packets that are sent out of a queue are marked with the ECN flag as soon as congestion occurs in the queue, the delay of the congested queue and the waiting time for congestion notification are shortened. That is, congestion notification is accelerated.

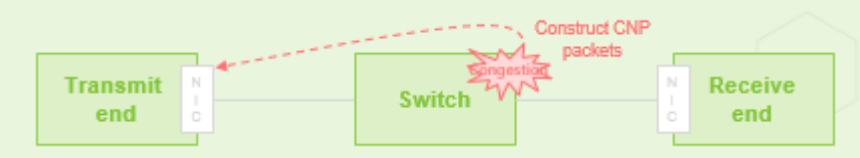


ECN allows the NIC of the source server to quickly respond to congestion by reducing the transmission rate. In this way, the cache congestion of network devices can be quickly mitigated, effectively reducing the delay and improving application performance.

Traditional CNP packets are sent by the destination server. Even if RoCEv2 packets encounter queue congestion on the intermediate device, the intermediate device only adds the ECN flag to the packets. The packets then still need to be sent to the destination server. The NIC of the destination server identifies that congestion occurs, constructs CNP packets, and sends the packets to the source server. Due to the long feedback path of congestion notification, the traffic rate cannot be reduced in time. As a result, the buffer of the intermediate device may encounter severer congestion, and in some cases, PFC is triggered and spread. This degrades the application performance.



To shorten the feedback path of congestion notification, the intermediate device can assume the role of the NIC on the destination server, immediately generate CNP packets, and return the packets to the source server when congestion occurs. By doing so, the source server can immediately adjust the traffic rate to relieve congestion of the queue buffer on the intermediate device.

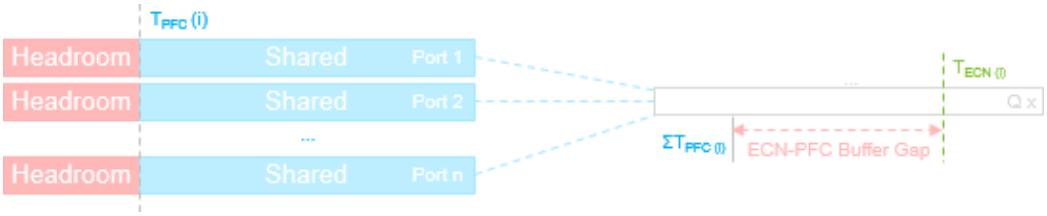


If a switch is required to construct CNP packets and send them to the source server when congestion occurs, the forwarding chip needs to establish the mapping relationship between source and destination queue priorities of flows. To implement this, the switch needs to learn and maintain a flow table.

For distributed parallel applications such as high-performance computing and distributed storage in a data center, the N:1 Incast model is used. A large value of N indicates a higher Incast degree and heavier burden for burst traffic in the buffer.

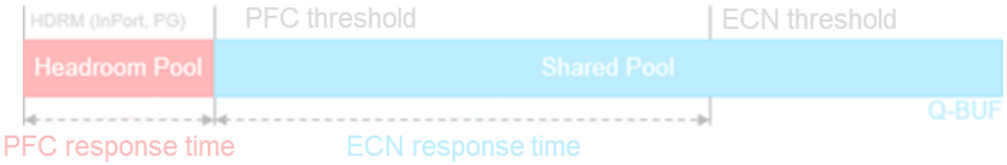
To relieve congestion in the interface queue buffer, the switch can add the ECN flag to packets. When queues are congested, the switch requests the source server to reduce the transmission rate. This prevents the queue buffer from being congested continuously or even deteriorating, thereby avoiding PFC triggering. PFC triggering and spreading will reduce the throughput of application flows, increase the FCT of applications, and degrade the application performance.

For the Incast traffic model, $\Sigma T_{PFC}(i)$ must be larger than T_{ECN}. This prevents PFC for the priority of the inbound interface from being triggered before packets in the outbound interface queue have the ECN flag added.



If ECN flag marking is preferentially triggered, CNP packets are sent to request the source server to reduce the transmission rate, rather than using PFC to stop sending flows to eliminate packet loss. This relieves the queue congestion on the intermediate device. However, between the time the ECN flag is marked and the time the rate of the source server is reduced, traffic still passes through the congested queue. Congestion may become aggravated before the rate is reduced. As a result, PFC is triggered to stop sending flows.

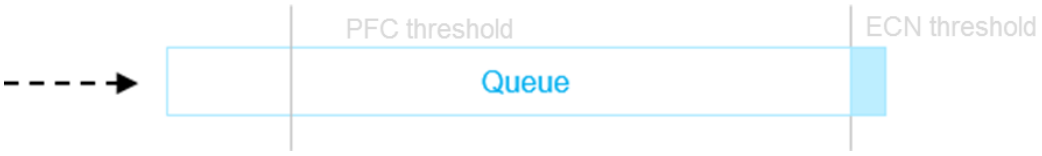
Ensure that the buffer between the ECN threshold and the PFC threshold is large so that traffic during the period can be properly processed.



The required buffer between the ECN threshold and the PFC threshold depends on the concurrency of the Incast traffic model. A higher Incast concurrency indicates a larger buffer between the ECN threshold and the PFC threshold. In theory, the buffer should reach hundreds of megabytes. However, the packet buffer of the switch's forwarding chip is limited, and the packet buffer of the existing BRCM XGS forwarding chip cannot meet this requirement. Therefore, PFC triggering cannot be avoided.

In addition, when configuring the ECN threshold, take into consideration the balance between the throughput and the delay.

If a low ECN threshold is used, the device can add the ECN flag and request the source server to reduce the transmission rate as soon as possible. In this way, the low buffer depth is maintained, that is, the queue delay is low. This is beneficial to delay-sensitive mice flows. However, the low ECN threshold affects throughput–sensitive elephant flows, failing to ensure high throughput.

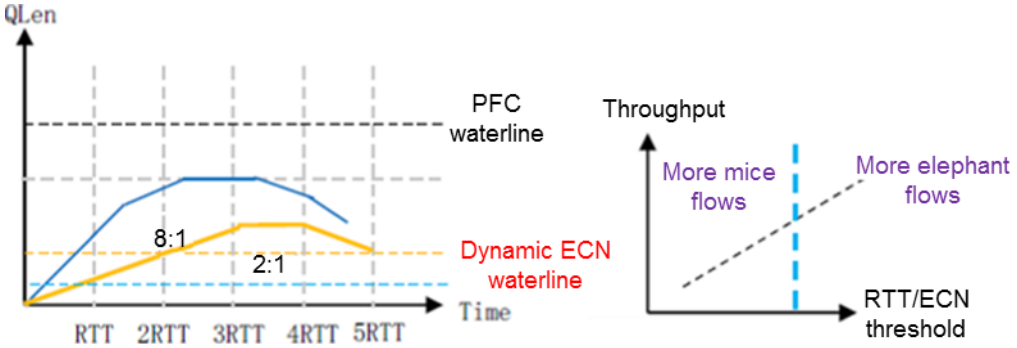


If a high ECN threshold is used, the time for triggering ECN flag marking is prolonged. The queue's capability to process burst traffic is enhanced and bandwidth of throughput-sensitive elephant flows is ensured. However, when a queue is congested, there is a long queue delay. This is detrimental to delay-sensitive mice flows.

The ECN threshold is relevant to the proportions of elephant and mice flows on network devices and the concurrency of the Incast traffic model.

When the concurrency of the Incast traffic model is high, a low ECN threshold is used to ensure the low latency of the queue. In addition, the buffer between the ECN threshold and the PFC threshold needs to be increased, avoiding PFC triggering. In contrast, when the concurrency of the Incast traffic model is low, a high ECN threshold is used to narrow the buffer space between the ECN threshold and the PFC threshold. This ensures high throughput of a queue.

When the proportion of mice flows is high, a low ECN threshold is used to ensure low latency of most mice flows. When the proportion of elephant flows is high, a high ECN threshold is used to ensure high throughput of most elephant flows.

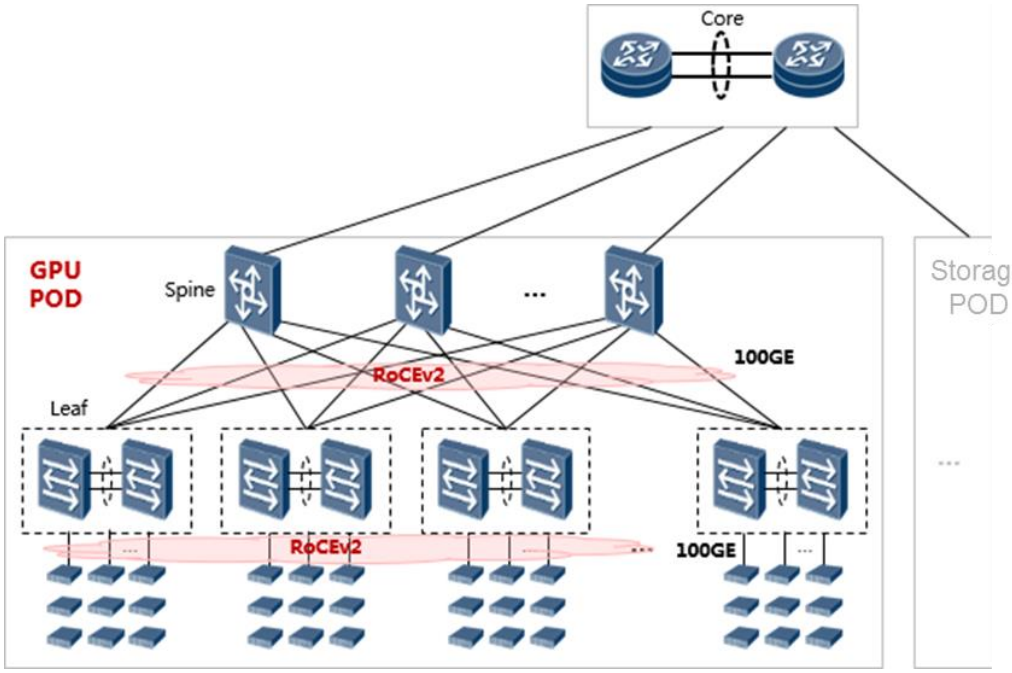


The ECN threshold needs to be dynamically adjusted based on the concurrency of the Incast traffic model and proportions of elephant and mice flows to achieve low latency of mice flows, high throughput of elephant flows, and PFC triggering.

The forwarding chip of the switch needs to obtain the concurrency of the Incast traffic model and data about the proportion of elephant or mice flows in real time. The data includes the queue depth, queue sending rate, and chip buffer usage. The forwarding chip then calculates the ECN threshold.

3.3 AI Fabric Network Topology

Computing traffic is within a PoD, and there is no computing traffic between PoDs.



In AI scenarios, GPUs are used as dense compute nodes. They support 100GE access and are dual-homed to leaf switches through M-LAG.

3.4 AI Fabric Network Device Selection

3.4.1 Leaf Switch

In AI scenarios, it is recommended that the CE8861-EI be used as the leaf switch. It provides a maximum of sixteen 100GE uplink interfaces. In distributed storage scenarios, it is recommended that the CE6865-48S8CQ-EI be used as the leaf switch. It provides a maximum of forty-eight 25Gbit/s interfaces and eight 100GE uplink interfaces.

3.4.2 Spine Switch

You are advised to use the CE8850-64CQ-EI as the spine switch. It supports a maximum of 64 100GE interconnection interfaces, 64 leaf switches, and 1,000 servers.

If there are more than 1,000 servers, the CE12800E equipped with 36*100GE FD-X card is recommended. It supports more than 5,000 servers.

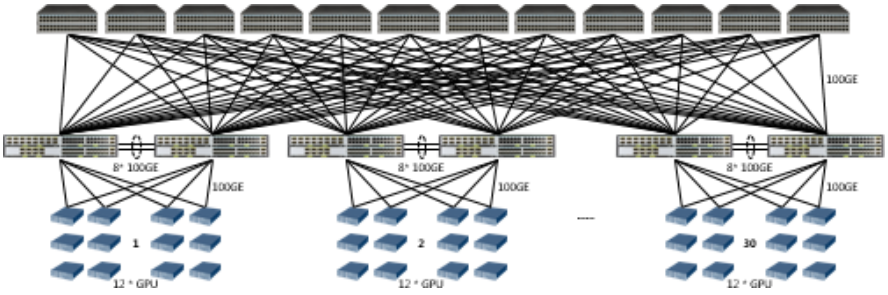
3.4.3 Server NIC

You are advised to use Mellanox ConnectX-5 or ConnectX-4 series 100GE NIC. You can also use the HI1822 NIC of Huawei servers.

3.4.4 Device Quantity

The network with 360 GPU compute nodes is used as an example. The CE8850-64CQ-EI is used as the spine switch, the CE8861-EI is used as the leaf switch uses CE8861-EI, and servers are dual-homed to leaf switches through M-LAG.

For the preceding switch models, each leaf switch supports a maximum of 16 GPU servers. Considering the convergence ratio of 1:1 of leaf switches and 12 GPU servers connected to each leaf switch, a total of 60 (360/12 x 2) leaf switches are required.



The M-LAG link between two leaf switches is composed of eight 100GE links, and there are twelve 100GE uplinks and twelve 100GE downlinks.

Each spine switch is connected to 60 leaf switches, so a total of 12 spine switches are required.

Table 3-1 Required network devices for building the network with 360 GPU compute nodes

Device	Model	Quantity
Spine switch	CE8850-64CQ-EI	12

Device	Model	Quantity
Leaf switch	CE8861-EI	60
Server NIC	Mellanox ConnectX-5	720

3.5 AI Fabric Feature Configuration

The following describes how to configure the AI Fabric in AI scenarios.

Table 3-2 Procedure for the simplest AI Fabric deployment

Procedure	Description
Enable the AI Fabric function.	The software provides the default functions and parameter settings of the AI Fabric function in AI scenarios.
Configure PFC.	Configure the priority of RoCEv2 traffic transmitted on the entire network and enable PFC.

For leaf or spine fixed switches, the AI Fabric function can be enabled using the following command:

```
[~HUAWEI] low-latency fabric
```

After the AI Fabric function is enabled, the device must restart. Then you can configure PFC priority mapping. Generally, no additional command line configuration is required.

3.5.2 Leaf Switch

3.5.2.1 Enabling the AI Fabric Function

The AI Fabric function is under license control and needs to be enabled using a command. After the license is loaded, run the following command to enable the AI Fabric function:

```
[~HUAWEI] low-latency fabric
```

When the AI Fabric function is enabled, Virtual Input Queue (VIQ) and dynamic ECN threshold are enabled by default.

After the AI Fabric function is enabled, restart the device.

3.5.2.2 Configuring PFC Priority Mapping

A separate priority needs to be planned for RoCEv2 traffic in a point of delivery (PoD), and PFC is enabled based on the traffic priority. Then the RoCEv2 packet header field (802.1p or DSCP priority) carries the traffic priority in the PoD.

Assume that the priority of RoCEv2 traffic of distributed storage is 3. On the switch, PFC needs to be enabled for priority 3 on each panel interface by running the following commands:


```
[~HUAWEI] dcb pfc mypfc
[*HUAWEI-dcb-pfc-mypfc] priority 3
[~HUAWEI-25GE1/0/1] dcb pfc enable mypfc mode manual
```

To enable a PFC-enabled switch to perform priority mapping based on the DSCP priority, run the following command on the CE switch:

```
[~HUAWEI] dcb pfc dscp-mapping enable slot 1
```

Then, run the following command to configure each panel interface on the CE switch to trust DSCP priorities of incoming packets:

```
[~HUAWEI-25GE1/0/1] trust dscp
```

If the DSCP priority is planned on a network, configure a DiffServ profile and apply it to each panel interface on the CE switch.

```
[~HUAWEI] diffserv domain ds1
[*HUAWEI-dsdomain-ds1] ip-dscp-inbound 8 phb af3 green
[~HUAWEI-25GE1/0/1] trust upstream ds1
```

In addition, a priority needs to be planned separately for CNP packets so that CNP packets can be transmitted without delay. The switch is configured with PQ scheduling for each outbound interface queue based on the priority. Assuming that the priority queue is 6, the command is as follows:

```
[~HUAWEI-25GE1/0/1] qos pq 6
```

3.5.2.3 Configuring VIQ

When the AI Fabric function is enabled, VIQ is enabled by default. VIQ eliminates packet loss caused by congestion due to traffic bursts in the forwarding chip.

VIQ provides buffer optimization for internal packets on the forwarding chip by default in typical lossless application scenarios of AI Fabric. After VIQ is enabled, the switch can prevent packet loss caused by congestion due to RoCEv2 traffic bursts. In most cases, you do not need to perform any configuration.

3.5.2.4 Configuring Dynamic ECN Threshold

The NIC implements Data Center Quantized Congestion Notification (DCQCN) for RoCEv2. DCQCN requires only the switches on the network to mark the ECN flag for the congested packets. Then NICs at both ends adjust the rate limit based on the ECN flag.

The existing switches can add the ECN flag to the packets that encounter queue congestion, notifying the source end to reduce the rate in order to relieve congestion. The switch marks the ECN flag based on the configured ECN threshold.

When configuring the ECN threshold, consider the balance between the throughput and delay of service flows. If a high ECN threshold is set, the queue has a strong capability to handle burst traffic, which makes it easy to process high-throughput services. However, the queue depth and delay are large, which makes it difficult to process delay-sensitive control or protocol flows. In contrast, if a low ECN threshold is used, the queue length and delay are low, which is beneficial to process delay-sensitive service flows. However, the queue has a weak capability to handle traffic bursts, adversely affecting the processing of high-throughput services.

The dynamic ECN threshold considers disadvantages of the static ECN threshold and balance between the throughput and delay of service flows from the perspective of distributed

high-performance applications. The dynamic ECN threshold uses an adjustment algorithm that detects the change of RoCEv2 service traffic in the internal queue of a switch and periodically adjusts the ECN threshold of the RoCEv2 priority queue to keep a low queue delay while ensuring the throughput.

Dynamic ECN threshold is automatically enabled when the AI Fabric function is enabled. In most cases, you do not need to perform any configuration. However, in the following situations, you can adjust the parameters of the dynamic ECN threshold algorithm based on the service performance counters.

1. PFC is triggered frequently, decreasing the throughput of RoCEv2 service flows.
2. PFC is not triggered, and the throughput of RoCEv2 service flows does not meet service requirements.

You can run the following command to adjust the target delay:

```
[~HUAWEI-low-latency-fabric] qos dynamic-ecn-threshold target-delay 300
```

The default queue target delay provided by the AI Fabric software version is 50 us.

If PFC is triggered frequently and the throughput of service flows is affected, run the preceding command to decrease the queue target delay to reduce or avoid PFC triggering.

If PFC is not triggered and the throughput of service flows does not meet service requirements, run the preceding command to increase the queue target delay. This improves the queue's capability to process burst traffic.

You can run the following command to check the PFC triggering count or frequency:

```
<HUAWEI> display dcb pfc interface 25ge 1/0/1
-------------------------------------------------------------------------------
--------------------
Interface        Queue        Received(Frames)     Transmitted(Frames)
                            DeadlockNum          RecoveryNum
-------------------------------------------------------------------------------
--------------------
25GE1/0/1          3                    0                    0
                                       0                    0
-------------------------------------------------------------------------------
--------------------
```

3.5.2.5 Configuring Fast CNP

The NIC supports DCQCN for RoCEv2. When the destination NIC receives a packet with the ECN flag, it constructs a CNP packet and sends it to the source NIC, requesting the source NIC to reduce the rate of congested flows.

Fast CNP shortens the congestion feedback path. The intermediate switch that encounters congestion constructs and sends a CNP packet to the source NIC, without waiting for the destination NIC to receive the packet with the ECN flag.

When the AI Fabric function is enabled, the fast CNP function is not enabled by default. In distributed storage scenarios, when network convergence occurs, burst traffic easily causes congestion. For this reason, fast CNP is recommended. Then the source NIC can be notified as soon as possible to reduce the rate of congested flows. This prevents PFC triggering that may lower the throughput of distributed storage. Fast CNP is enabled using the following command:

```
[~HUAWEI-low-latency-fabric] qos fast-cnp enable
```


3.5.2.6 Dynamic Load Balancing

Leaf switches connect to spine switches through multiple uplinks. These uplinks form ECMP equal-cost paths. By default, leaf switches use static load balancing to distribute RoCEv2 traffic to uplinks. In static load balancing mode, route selection is only based on the characteristics of packets. That is, the traffic bandwidth and link load usage are not considered. This may cause hash polarization. As a result, the link is seriously congested or, in the worst case, packets are discarded. The throughput and delay of RoCEv2 service flows are also affected.

To solve the preceding static hash problem, dynamic load balancing dynamically selects proper links based on the traffic bandwidth and link load. In this way, traffic can be evenly distributed to prevent a long queue delay or packet loss caused by heavy load on a link.

When the AI Fabric function is enabled, dynamic load balancing is not enabled by default. In distributed storage scenarios, the bandwidth of each RoCEv2 flow is high. You can run the following command to check whether loads of 100GE uplinks on the leaf switch are heavy or unbalanced.

```
[~HUAWEI] display interface brief
InUti/OutUti: input utility rate/output utility rate
Interface         PHY     Protocol    InUti    OutUti   inErrors  outErrors
100GE1/0/1        up      up      19.99%    12.01%       0        0
100GE1/0/2        up      up      50.82%    99.99%       0        0
100GE1/0/3        up      up      12.36%     8.29%       0        0
100GE1/0/4        up      up      22.87%    16.56%       0        0
```

If links are unbalanced, run the following command to enable ECMP dynamic load balancing globally:

```
[~HUAWEI] load-balance ecmp dynamic enable
```

3.5.2.7 Configuring PFC Deadlock Detection

Each device on the entire network must be enabled with PFC so that RoCEv2 packets are not discarded on the Ethernet network.

After PFC is enabled, you are advised to enable PFC deadlock detection and recovery on CE switches. This prevents PFC deadlocks due to traffic topology changes caused by network device faults or exceptions.

```
[~HUAWEI] dcb pfc mypfc
[*HUAWEI-dcb-pfc-mypfc] priority 3 deadlock-detect time 1
[*HUAWEI-dcb-pfc-mypfc] priority 3 deadlock-recovery time 1
```

After detecting a PFC deadlock, the CE switch generates an alarm. You can run the following command to check the PFC deadlock count.

```
<HUAWEI> display dcb pfc interface 25ge 1/0/1
------------------------------------------------------------------------------
--------------------
Interface        Queue        Received(Frames)     Transmitted(Frames)
                            DeadlockNum          RecoveryNum
------------------------------------------------------------------------------
--------------------
25GE1/0/1         3                    0                  0
                                      0                  0
```



```
--------------------------------------------------------------------------------
-------------------
```

3.5.3 Spine Switch

3.5.3.8 Configuring the Fixed Switch

When the CE8850-64CQ-EI fixed switch is used as the spine switch, configure the following functions and features for 100GE interfaces according to the configuration of the leaf switch:

- AI Fabric enabling
- PFC priority mapping
- Packet buffer
- Dynamic ECN threshold
- Fast CNP
- PFC deadlock detection

3.5.3.9 Configuring the Modular Switch

When the CE12800E modular switch equipped with the 36*100GE FD-X card is used as the spine switch, configure the following functions and features for 100GE interfaces according to the configuration of the leaf switch:

- AI Fabric enabling
- PFC priority mapping
- Packet buffer
- Dynamic ECN threshold
- Fast CNP
- PFC deadlock detection

3.5.4 Server NIC

The server NIC must support RoCEv2 and be configured with the following functions:

- Configure QoS priority mapping based on the planned priority of RoCEv2 packets on the entire network, and set the DSCP priority corresponding to RoCEv2 packets.
- Enable PFC based on the priority of RoCEv2 packets.
- Configure Enhanced Transmission Selection (ETS) if separate bandwidth allocation control is required for RoCEv2 and non-RoCEv2 packets.
- Configure DCQCN for RoCEv2.
- Configure the MTU for RoCEv2 packets.