IEEE 802 Nendica Report:
Intelligent Lossless Data Center Networks

Editor

Name	Affiliation
Guo, Liang	CIACT/ODCC
Congdon, Paul	Huawei

Nendica Chair

Name	Affiliation
Marks, Roger	Huawei

Contributors/Supporters

Name	Affiliation
Li, Jie	CIACT/ODCC
Gao, Feng	Baidu
Gu, Rong	China Mobile
Zhao, Jizhuang	China Telecom
Chen, Chuansheng	Tencent
Yin, Yue	Huawei
Song, Qingchun	Mellanox
Lui, Jun	Cisco
He, Zongying	Broadcom
Sun, Liyang	Huawei

Trademarks and Disclaimers

IEEE believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. IEEE is not responsible for any inadvertent errors.

Comments on this report are welcomed by Nendica: the IEEE 802 "Network Enhancements for the Next Decade" Industry Connections Activity: <https://1.ieee802.org/802-nendica>

Comment submission instructions are available at: <https://1.ieee802.org/802-nendica/nendica-dcn>

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN xxx-x-xxxx-xxxx-x XXXXXXXXXX

IEEE prohibits discrimination, harassment, and bullying. For more information, visit http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

To order IEEE Press Publications, call 1-800-678-IEEE.
Find IEEE standards and standards-related product listings at: http://standards.ieee.org

NOTICE AND DISCLAIMER OF LIABILITY CONCERNING THE USE OF IEEE SA INDUSTRY CONNECTIONS DOCUMENTS

This IEEE Standards Association ("IEEE SA") Industry Connections publication ("Work") is not a consensus standard document. Specifically, this document is NOT AN IEEE STANDARD. Information contained in this Work has been created by, or obtained from, sources believed to be reliable, and reviewed by members of the IEEE SA Industry Connections activity that produced this Work. IEEE and the IEEE SA Industry Connections activity members expressly disclaim all warranties (express, implied, and statutory) related to this Work, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; quality, accuracy, effectiveness, currency, or completeness of the Work or content within the Work. In addition, IEEE and the IEEE SA Industry Connections activity members disclaim any and all conditions relating to: results; and workmanlike effort. This IEEE SA Industry Connections document is supplied "AS IS" and "WITH ALL FAULTS."

Although the IEEE SA Industry Connections activity members who have created this Work believe that the information and guidance given in this Work serve as an enhancement to users, all persons must rely upon their own skill and judgment when making use of it. IN NO EVENT SHALL IEEE OR IEEE SA INDUSTRY CONNECTIONS ACTIVITY MEMBERS BE LIABLE FOR ANY ERRORS OR OMISSIONS OR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS WORK, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Further, information contained in this Work may be protected by intellectual property rights held by third parties or organizations, and the use of this information may require the user to negotiate with any such rights holders in order to legally acquire the rights to do so, and such rights holders may refuse to grant such rights. Attention is also called to the possibility that implementation of any or all of this Work may require use of subject matter covered by patent rights. By publication of this Work, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying patent rights for which a license may be required, or for conducting inquiries into the legal validity or scope of patents claims. Users are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. No commitment to grant licenses under patent rights on a reasonable or non-discriminatory basis has been sought or received from any rights holder. The policies and procedures under which this document was created can be viewed at http://standards.ieee.org/about/sasb/iccom/.

This Work is published with the understanding that IEEE and the IEEE SA Industry Connections activity members are supplying information through this Work, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. IEEE is not responsible for the statements and opinions advanced in this Work.

TABLE OF CONTENTS

1. INTRODUCTION 2
   - Scope 2
   - Purpose 2
2. BRINGING THE DATA CENTER TO LIFE 2
   - A new world with data everywhere 2
   - Today's data center enables the digital real-time world 4
3. EVOLVING DATA CENTER REQUIREMENTS AND TECHNOLOGY 6
   - Technology evolution 6
   - Network requirements 10
4. CHALLENGES WITH TODAY'S DATA CENTER NETWORK 26
   - High bandwidth and low latency tradeoff 26
   - Deadlock free lossless network 27
   - Congestion control issues in large-scale data center networks 29
   - Configuration complexity of congestion control algorithms 37
5. NEW TECHNOLOGIES TO ADDRESS NEW DATA CENTER PROBLEMS 37
   - Approaches to PFC storm elimination 38
   - Improving Congestion Notification 40
   - Intelligent congestion parameter optimization 44
6. STANDARDIZATION CONSIDERATIONS 46
7. CONCLUSION 46
8. CITATIONS 47

1 Introduction

<<Editor’s notes will be noted inside these marking and removed in future drafts>>

<<short intro and the more detailed background intro is section 2. This will be written near the end>>

This paper is the result of the Data Center Networks work item [1] within the IEEE 802 “Network Enhancements for the Next Decade” Industry Connections Activity known as Nendica. The paper is an update to a previous report, IEEE 802 Nendica Report: The Lossless Network for Data Centers published on August 17, 2018 [2]. This update provides additional background on evolving use cases in modern data centers and proposes solutions to ~~new~~additional problems identified by this paper.

Scope

The scope of this report includes...

Purpose

The purpose of this report is to ...

2 Bringing the data center to life

A new world with data everywhere

~~<<~~

- ~~✓ Enterprise digital transformation needs more data for using AI~~
- ~~✓ Machine translation and search engines need to be able to process huge data simultaneously~~
- ~~✓ The era of internet celebrity webcast, all-people online games, data explosion~~
- ~~✓ Consumption upgrade in the new era of take-out, online takeout platform schedule and deliver massive orders~~
- ~~✓ The XX service of the carrier has higher requirements on data center network~~
- ~~✓ Data-based New World Requires Ubiquitous Data Center Technologies.~~

~~>>~~

Digital transformation is driving change in both our personal and professional lives. Work flows and personal interactions are turning to digital processes and automated tools that are enabled by the

Cloud, Mobility, and the Internet of Things. The Intelligence behind the digital transformation is Artificial Intelligence (AI). Data centers running AI applications with massive amounts of data are recasting that data into pertinent timely information, automated human interactions, and refined decision making. The need to interact with the data center in real-time is more important than ever in today's world where augmented reality, voice recognition, and contextual searching demand immediate results. Data centers networks must deliver unprecedented levels of performance, scale, and reliability to meet these real-time demands.

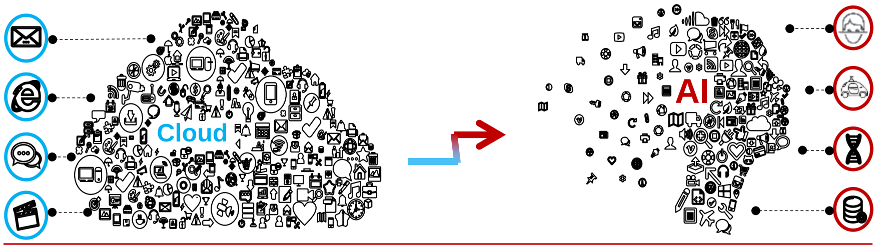
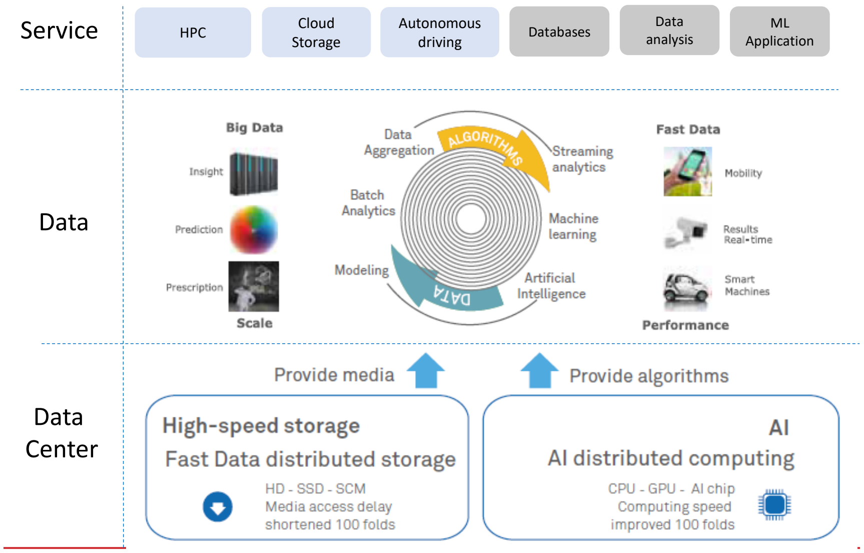


Figure 1 – Digital Transformation in the Era of AI

Data centers in the cloud era focused on application transformation and the rapid deployment of services. In the AI era, data centers are the source of information and algorithms for the real-time digital transformation of our digital lives. The combination of high-speed storage and AI distributed computing render big data into fast data, access by humans, machines, and things. A high-performance, large scale data center network without packet loss is critical to the smooth operation of the digital transformation.

For high-performance applications, such as AI, key measures for network performance include throughput, latency, and congestion. Throughput is dependent on the total capacity of the network for quickly transmitting a large amount of data. Latency refers to the total delay ~~on the network when performing~~in a transaction across the data center network. When the traffic load exceeds the network capacity, congestion occurs. Packet loss is a factor that seriously affects both throughput and latency. Data loss in a network may cause a series events that deteriorate performance. For example, an upper-layer application may need to retransmit lost data in order to continue. Retransmissions can increase load on the network, causing further packet loss. In some applications, delayed results are not useful, and the ultimate results can be discarded, thus wasting resources. In other cases, the delayed result is just a small piece of the puzzle being assembled by the upper-layer application that has now been slowed down to the speed of the slowest worker. More seriously, when an application program does not support packet loss and cannot be restored to continue, a complete failure or damage can be caused.

~~Data centers ultimately deliver the services in this era of digital transformation to our real-time digital lives. The combination of high-speed storage and AI distributed computing render big data into fast data, access by humans, machines, and things. A high-performance, large scale data center network without packet loss is critical to the smooth operation of the modern data center.~~

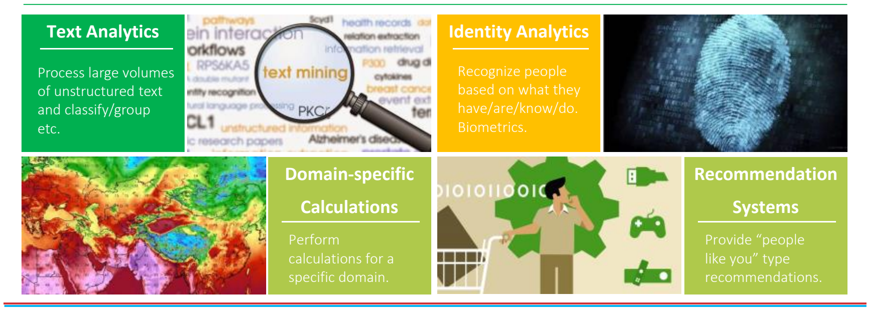


~~Figure 1 – How the Data Center enables our real-time digital lives~~

Today's data center enables the digital real-time world

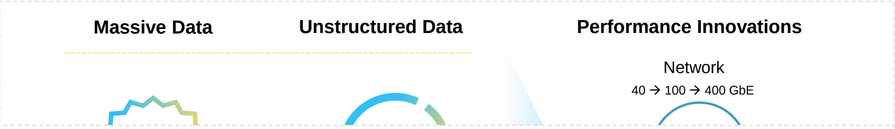
Currently, digital transformation of various industries is accelerating. According to analysis data, 64% of enterprises have become the explorers and practitioners of digital transformation <<IDC reference>>. Among 2000 multinational companies, 67% of CEOs have made digitalization the core of their corporate strategies [3].

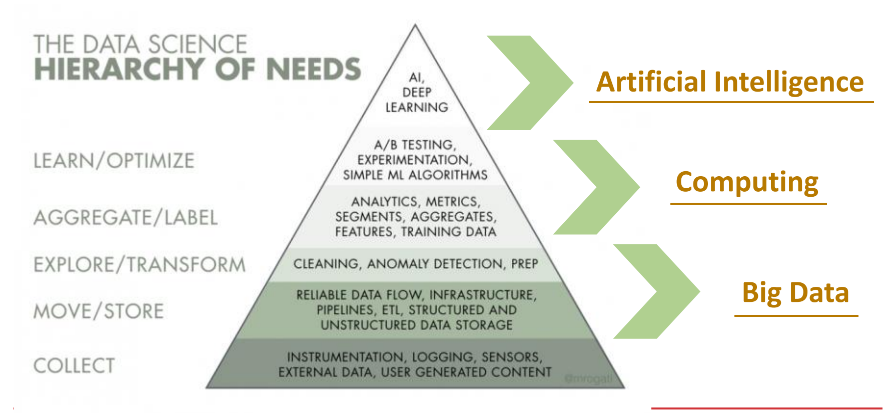
A large amount of data will be generated during the digitalization process, becoming a core asset, and enabling ~~a new~~the emergence of Artificial Intelligence ~~Applications as seen in Figure W~~applications. Huawei GIV predicts that the data volume will reach 180 ZB in 2025 [4]. However, data is not the “end-in-itself”. Knowledge and wisdom extracted from data are eternal values. However, the proportion of unstructured data (such as raw voice, video, and image data) increases continuously, and will ~~reach over~~account for 95% of all data in the future. ~~The~~Performance innovations are needed to extract the value from the raw data. At this scale, the current big data ~~analytics method is~~analytic methods are helpless. If manual processing is used, the data volume will be far greater than the processing capability of all human beings. The AI ~~algorithm~~approach based on machine computing for deep learning can filter out massive amounts of invalid data and automatically reorganize useful information, providing more efficient decision-making suggestions and smarter behavior guidance.


Text Analytics
Process large volumes of unstructured text and classify/group etc.
text mining
Identity Analytics
Recognize people based on what they have/are/know/do. Biometrics.
Domain-specific Calculations
Perform calculations for a specific domain.
Recommendation Systems
Provide “people like you” type recommendations.

~~Figure 2 – Emerging Artificial Intelligence Applications~~

~~AI applications are emerging everywhere, as shown in Figure 2.~~




THE DATA SCIENCE
HIERARCHY OF NEEDS
LEARN/OPTIMIZE
AGGREGATE/LABEL
EXPLORE/TRANSFORM
MOVE/STORE
COLLECT
AI, DEEP LEARNING
A/B TESTING, EXPERIMENTATION, SIMPLE ML ALGORITHMS
ANALYTICS, METRICS, SEGMENTS, AGGREGATES, FEATURES, TRAINING DATA
CLEANING, ANOMALY DETECTION, PREP
RELIABLE DATA FLOW, INFRASTRUCTURE, PIPELINES, ETL, STRUCTURED AND UNSTRUCTURED DATA STORAGE
INSTRUMENTATION, LOGGING, SENSORS, EXTERNAL DATA, USER GENERATED CONTENT
@mrogati
Artificial Intelligence
Computing
Big Data

~~Figure 3 – Data is the Foundation for Artificial Intelligence~~

~~Cloud~~The cloud data ~~centers improve~~center architecture improved the performance ~~of these applications. Cloud data centers are designed to~~ and scale ~~and act more like a service support center. They are application-centric and use the~~ of applications in general. The cloud platform ~~to quickly distribute~~ allows rapid distribution of IT resources~~. While~~ to create an application-centric service model. In the AI era, the applications are consuming unprecedented amounts of data and the cloud data ~~centers are application-centric, they are founded on big~~center architecture is augmented with necessary performance innovations to handle the load. Seamlessly introducing these innovations along with new AI applications can be tricky in an existing cloud data ~~as shown in Figure 3.~~

~~So, within data centers, understanding~~center. Understanding how to efficiently process data based on the needs of ~~different~~AI applications is a key focus area. ~~Data centers must know where to reserve~~Orchestrating the flow of data between the storage ~~to efficiently transmit the data to the~~and computing ~~engines~~resources of the applications~~.~~ is a critical success factor.

3 Evolving data center requirements and technology

Requirements evolution

~~<< First discuss the new and evolving requirements for data center networks hosting AI applications. These include: 1. huge amounts of data for AI learning. What is the data and why is it so large? Why is AI better with more data? 2. To hold the huge amounts of data, it must be fast and distributed. The latency needs to compete with local storage. 3. Huge amounts of computing cycles needed to work on that data. Describe the AI computing models; data parallelism vs model parallelism and how these differ in network communication requirements. >>~~

~~Take AI training of~~AI applications put pressure on the data center network. Consider AI training for self-driving cars as an example, the deep learning algorithm relies heavily on massive sample data and high-performance computing capabilities. ~~Training~~The training data collected is ~~close to~~approaching the P level (1PB = 1024 TB) per day. If traditional hard disk storage and common CPUs ~~are~~were used to process the data, it ~~takes~~could take at least one year to complete the training, which is ~~almost impossible.~~clearly impractical. To improve AI data processing efficiency, revolutionary changes are ~~occurring~~needed in the storage and computing fields. ~~The development of high-speed~~For example, storage ~~technology will help users to access the content more conveniently. Other data center technologies should be evolved together with distributed storage to ensure customers can obtain high input and output speed. Storage~~ performance needs to improve by an order of magnitude to achieve more than 1 million input/output operations per second (IOPS) [5].

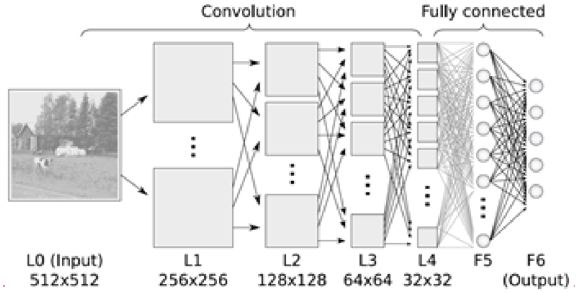
Storage media ~~evolve~~has evolved from HDDs to SSDs to meet real-time data access requirements, reducing the medium latency by more than 100 times. ~~With the significant improvement of storage media and computing capabilities, the current~~ Without similar improvements in network ~~communication~~ latency ~~becomes~~, these storage improvements are not realized and simply move the bottleneck ~~of further performance improvement in high-performance data center clusters. The~~from the media to communication latency. With networked SSD drives, the communication latency accounts for more than 60% of the total storage ~~E2E~~end-to-end latency~~, that is, more than half of~~. This creates a scenario where the ~~time of~~ precious storage media is idle~~.~~ more than half of the time.   When you consider recent improvements in both storage media and AI computing processors together, the communication latency accounts for more than 50% of the total latency, further hindering improvements and wasting resources [6].

Formatted: Font: Bold

~~In general, with the evolution of~~ The improvements in storage ~~media~~ and computing ~~processors, the communication duration accounts for more than 50% of the total communication duration, hindering the further improvement of computing and storage efficiency [6].~~

- ✓ ~~The development of fast storage provides necessary media for big data (distributed storage)~~
  - ~~Storage~~ performance ~~needs to improve by an order of magnitude to achieve more than 1 million input/output operations per second (IOPS).~~
  - ~~Communication latency has recently increased from 10% to 60% of storage E2E latency.~~
- ✓ ~~Computing speed improvement (distributed computing)~~

support the AI computing model~~-~~, which is growing in scale and complexity ~~is exploding~~



~~Figure A - Iterative machine learning network model~~

with the advent of AI ~~training is becoming increasingly complex with the development of~~cloud-based services. For example, there ~~are~~were 7 ExaFLOPS and 60 million parameters in ~~the Microsoft~~Microsoft's Resnet ~~in~~of 2015. ~~The number came to~~Baidu used 20 ExaFLOPS and 300 million parameters when ~~Baidu trained~~training their deep speech system in 2016. In 2017, the Google NMT used 105 ExaFLOPS and 8.7 billion parameters [7]. New characteristics of AI computing are requiring an evolution of data center network.

~~AI inference is the next great challenge so there must be an explosion of network design. The new characteristics of AI algorithm and huge computing workload require evolution of data center network.~~

Characteristics of AI computing

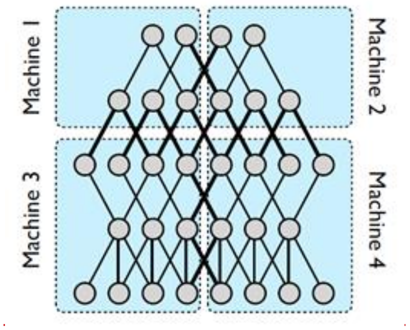
~~<< explain that AI computing is iterative, not a single pass, so communication is critical and the application runs for a long time. Describe the different modes for AI computing; data parallel vs model parallel and what that means to the network >>~~

Traditional data center services (web, ~~video~~database, and file storage) are transaction-based and the ~~calculation~~calculated results are often deterministic. For such tasks, there is ~~no~~little correlation or dependency between a single ~~calculation~~transaction and the associated network communication~~, and the~~. The occurrence ~~time~~ and duration of the ~~entire calculation and communication~~traditional transactions are random. AI computing, however, is ~~based on target~~different. It is an optimization ~~and~~problem with iterative convergence ~~is~~ required in the computing process~~, which~~. This causes high spatial correlation ~~in~~within the data sets and computing ~~process of AI services~~algorithms, and temporally creates similar correlations with communication ~~modes.~~flows.

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

~~To solve~~consequently must "divide-and-conquer" the ~~Big Data~~ problem~~, the~~. The computing model and input data ~~need to be~~sets are large (~~for~~e.g in a 100 MB node, the AI model ~~for~~with 10K rules requires more than 4 TB memory~~), for which a~~). A single server cannot provide enough storage capacity~~. In addition, because~~ and processing resources to handle the ~~computing time needs to be shortened and increasingly concurrent~~problem sequentially. Concurrent AI computing ~~of multiple~~and storage nodes ~~is~~are required~~, DCNs must be used to perform large-scale and concurrent~~ to shorten the processing time. The distributed AI computing~~.~~

~~Distributed AI computing~~ and storage requirement highlights the need for a fast, efficient, and lossless data center network that has the ~~following~~flexibility to support two distinct parallel modes of operation: model parallel computing and data 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 C~~.



~~Figure C - Model parallel training~~

~~For parallel data computing~~Model Parallel Computing

In model parallel computing, each node computes one part of the overall algorithm. Each node processes the same set of data, but with a different portion of the algorithm, resulting in an estimate for a differing set of parameters. The nodes exchange their estimates to converge upon the best estimate for all the data parameters. With model parallel computing, there is an initial distribution of the common data set to a distributed number of nodes, followed by a collection of individual

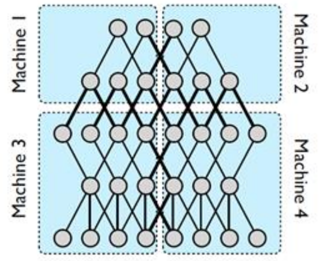


Figure 3 - Model parallel training

Commented [PC1]: A good summary of the differences is:

'Data parallelism' and 'model parallelism' are different ways of distributing an algorithm. These are often used in the context of machine learning algorithms that use stochastic gradient descent to learn some model parameters, which basically means that:

- The algorithm is trying to estimate some parameters from the given data.
- Parameters are estimated by minimizing the gradient against some loss function.
- Algorithm iterates over data in small batches.

In the data-parallel approach:

- The algorithm distributes the data between various cores.
- Each core independently tries to estimate the same parameter(s)
- Cores then exchange their estimate(s) with each other to come up with the right estimate for the step.

In the model-parallel approach:

- The algorithm sends the same data to all the cores.
- Each core is responsible for estimating different parameter(s)
- Cores then exchange their estimate(s) with each other to come up with the right estimate for all the parameters.

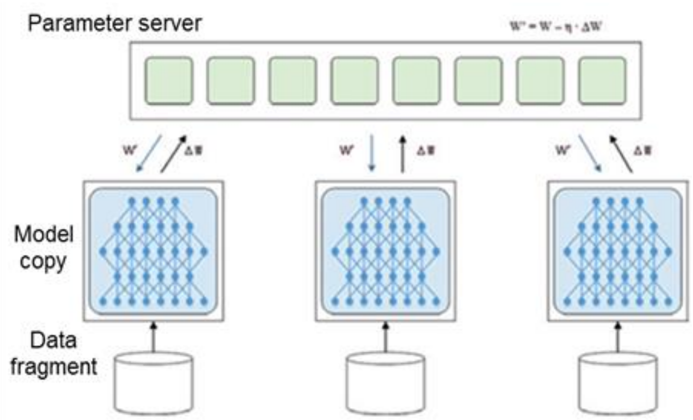
Data-parallel approach is useful when there are smaller number of nodes in the cluster and the number of parameters to be estimated is small whereas model-parallel approach is useful in the opposite condition.

parameters from each of the participating nodes. Figure 3 shows how parameters of the overall model may be distributed across computing nodes in a model parallel mode of operation.

Data Parallel Computing

In data parallel computing, each node loads the entire AI algorithm model~~. Multiple nodes can calculate the same model at the same time~~, but only processes part of the input data~~ is input to each node.~~. Each node is trying to estimate the same set of parameters using a different view of the data. 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~~calculations, the parameters are weighted and aggregated by a common parameter server as seen in Figure 4. The weighted parameter update requires that all nodes upload and obtain the information synchronously.

No matter the development of distributed storage or distributed AI training, data center network comes to the communication pressure. The waiting time for GPU communication exceeds 50% of the job completion time [8].



~~Figure D - Data parallel training~~

Evolving technologies

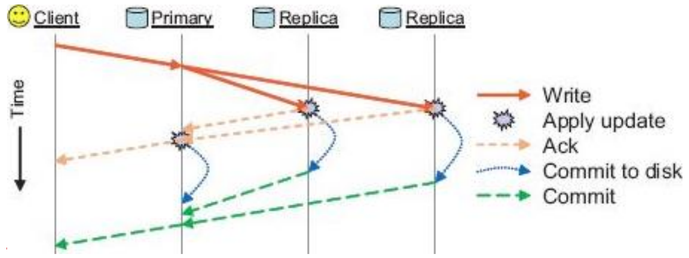
~~<< Here we describe some key new technologies that are evolving to meet the requirements. The main pieces should be fast storage (SSDs), GPUs, Smart Nics, Protocols like RDMA. Have a small sub-section on each of the above technology areas. End with what it means to the network to support these new technologies >>~~

Progress can be seen when evolving requirements and evolving technologies harmonize. New requirements often drive the development of new technologies and new technologies often enable new use cases that lead to, yet again, a new set of requirements. Breakthroughs in networked storage, distributed computing, system architecture and network protocols are enabling the utility of the next generation data center.

SSDs and NVMeoF: High throughput, low-latency network

In networked storage, a file is distributed to multiple storage servers for IO acceleration and redundancy. When a data center application reads a file, it will concurrently access different parts of data from different servers, and the data will be aggregated through a data center switch at nearly the same time. When a data center application writes a file, the data can trigger a series of storage transactions between distributed and redundant storage nodes. Figure ~~8~~5 shows an example of data center communication triggered by the networked storage service model.

When an application (i.e. Client in Figure ~~8~~5) requests to write a file, it will concurrently send data to the object storage device (OSD) servers. There are two types of OSD servers, one type is the primary, and the other type is the replica. When the primary servers receive data that need to be saved, it will transmit the data to the replica servers twice as backup (the orange arrowhead in Figure ~~8~~5). After receiving the data, the primary OSD server will send an ACK to client while the replica servers will send ACK to the primary server (pink dash line in Figure ~~8~~5). Each OSD server will then begin to commit the data to the storage medium. It takes a short period time to commit and store data. When the replica servers finish saving data, they will send commit notification to primary server to notify that the writing task is complete. Once the primary server has received all the commit information from all replica servers, the primary server will send a commit message to client. The storage write process is not complete until the primary server has sent the final commit message to the client.~~ << Consider making a comment about the impact of network latency here~~



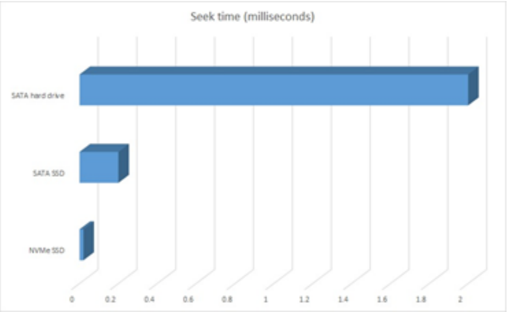
~~Figure 8 - Networked storage service model~~

~~>>.~~

The example highlights the importance of the network enabling both high throughput and low latency simultaneously. The bulk data being written to the primary storage server is transmitted multiple times to the replicas. The small sized acknowledgments and commit messages must be sequenced and ultimately delivered to the originating client before the transaction can complete, emphasizing the need for ultra-low latency.

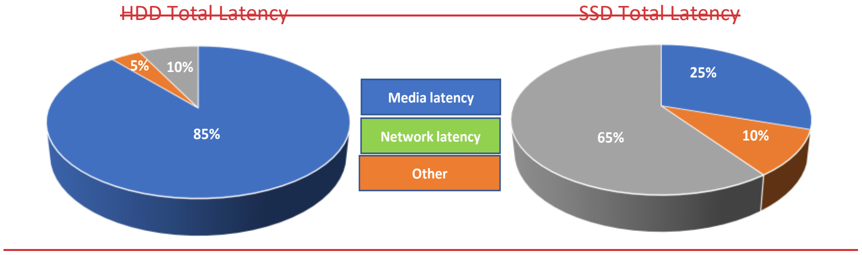
Massive improvements in storage performance have been achieved as the technology has evolved from HDD to SDD to NVMe (Non-Volatile Memory Express). The latest storage media technology, NVMe, has decreased access time by a factor of 1000 over previous HDD technology. ~~Figure 9 shows the difference in~~Sample seek ~~time~~times between the various technologies include; HDD = 2-5 ms, SATA SSD = 0.2 ms, and NVMe SSD = 0.02 ms. ~~Shorter~~While shorter overall average seek times are better, ~~but~~the performance of drives in each category can still vary [9].

~~When~~ NVMe~~ is used~~ -over-fabrics (NVMeoF) involves deploying NVMe for networked storage~~, the~~. The much faster access speed of the medium ~~can~~ result in greater network bottlenecks~~. Figure 10 shows a classical networked storage traffic model. In this traffic model, when data is aggregated each time, incast (many-to-one) easily to occurs. As storage performance continues to increase, pressure on the network increases, affecting distributed storage IO throughput.~~



~~Figure 9 - Performance improvement of storage media~~

~~Incast is a network traffic pathology that affects many-to-one communication patterns in datacenters. Incast increases application latency with the queuing delay of flows and decreases application throughput to something well below the link bandwidth [10]. The problem especially affects computing paradigms, such as AI training, where distributed processing cannot continue until all parallel threads in a stage complete.~~



~~Figure 11 – End-to-end latency breakdown for HDD and SDD scenarios~~

~~Since incast increases application latency, the concurrency of the networked storage system will be affected. Therefore, the performance of distributed IOPS is limited by network latency. With newer, faster storage technologies,~~ and the impact of network latency becomes more significant. Figure ~~11~~6 shows ~~that~~how network latency ~~is~~has become the primary bottleneck ~~in~~with networked SSD storage, whereas network latency was negligible with networked HDD storage. ~~Looking to~~To maximize the ~~future, with NVMe over fabrics (i.e. networked NVMe storage), to attain the maximum~~ IOPS performance of the new medium, the network latency problem must be resolved first.

Formatted: Indent: Left: 0"

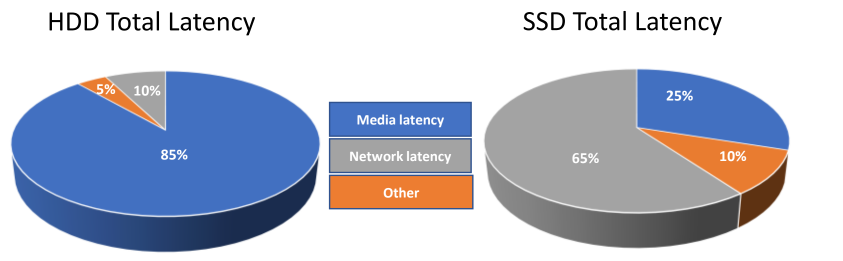
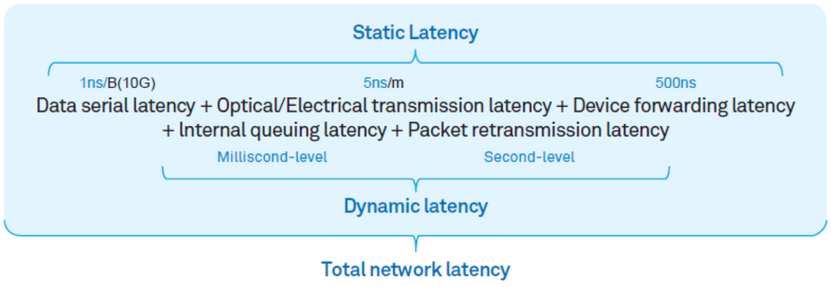


Figure 6 – End-to-end latency breakdown for HDD and SDD

~~To~~An analysis network latency ~~further,~~show that it ~~can be classified into~~is a combination of two distinct types of latency: static latency and dynamic latency. Static latency includes serial data latency, device forwarding latency, and optical/electrical transmission latency. This type of latency is determined by the capability of the ~~forwarding chip~~switching hardware and the transmission distance of the data. It usually ~~has a~~is fixed ~~specification.~~and very predictable. Figure ~~X says that~~ 7 shows the current industry measurements for static latency ~~is~~are generally at ~~ns~~nanosecond (10-9 second) or sub-~~us~~microsecond (10-6) level ~~in the industry~~, and ~~accounts~~account for less than 1% of the total end-to-end network delay.

~~The dynamic~~Dynamic latency ~~greatly affects the~~ plays a much greater role in total end-to-end network ~~performance. The dynamic latency ratio~~ delay and is ~~greater than 99%. The dynamic latency includes the~~ greatly affected by the conditions within the communication environment. Dynamic latency is created from delays introduced by internal queuing ~~latency~~ and packet retransmission ~~latency~~, which are caused by network congestion and packet loss. In the AI era, ~~traffic conflicts become~~ congestion from the unique traffic patterns of high-speed storage and specialized AI computing nodes becomes more and more severe on ~~networks~~the network. Packet queuing ~~or~~and packet loss ~~often occurs~~can occur frequently, causing the end-to-end network latency ~~within~~to skyrocket to the level of sub-seconds. ~~Therefore, the~~ The key ~~of the~~to low ~~latency network is the low dynamic latency.~~


Static Latency
1ns/B(10G)
5ns/m
500ns
Data serial latency + Optical/Electrical transmission latency + Device forwarding latency
+ Internal queuing latency + Packet retransmission latency
Millisecond-level
Second-level
Dynamic latency
Total network latency

~~Figure XX – Total Network Latency~~

~~Most existing network solutions focus on reducing the static latency caused by~~ end-to-end network ~~device forwarding, while the~~ latency is to improve dynamic latency ~~caused by~~.

The major component of dynamic latency is the delay from packet ~~loss during~~ retransmission when packets are dropped within the network ~~congestion~~. Packet loss latency is an order magnitude greater than queuing delay and has proven to have a ~~more~~ severe impact on applications. ~~In most existing systems the impact from latency comes primarily from dynamic latency which occurs across the network during packet loss of congestion management.~~ Figure ~~X below~~7 shows a typical network latency distribution.

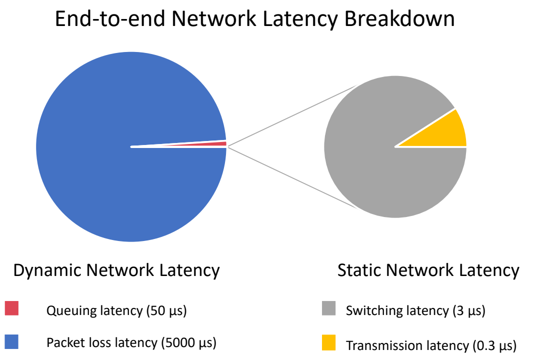
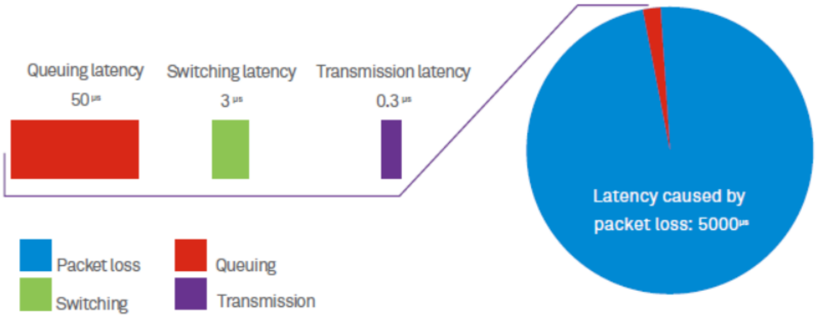


Figure 7 – Network Latency Breakdown

~~<< NOTE: We should define dynamic latency and its components. Highlight the issue of packet loss. That has not been discussed yet. >>~~

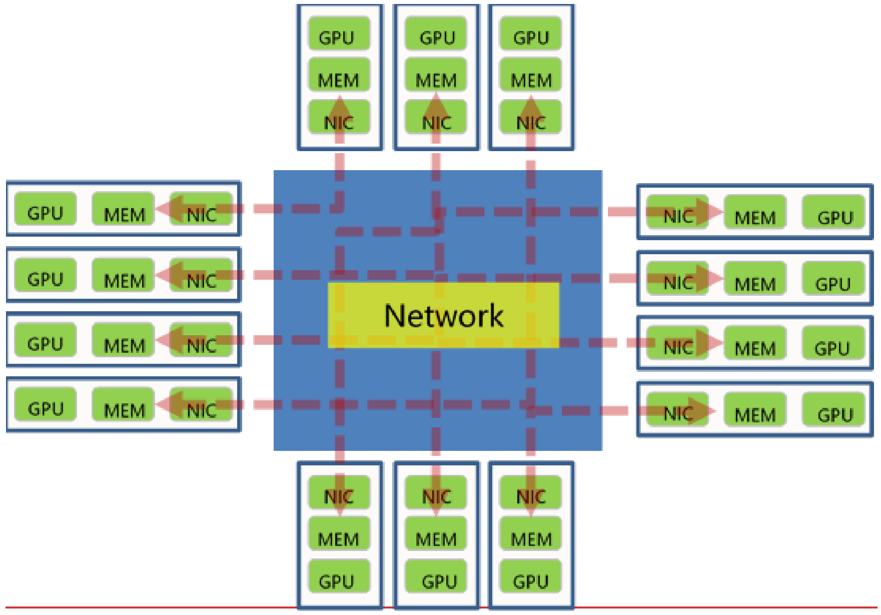


~~Figure YY – Latency Breakdown~~

Packet loss occurs when switch buffers are overrun because of congestion (NOTE: we ignore packet loss due low-probability bit errors during transmission). There are two key types of congestion within the network: in-network and incast. In-network congestion occurs on switch-to-switch links within the network fabric when the links become overloaded, perhaps due to ineffective load balancing. Incast congestion occurs at the edge of the network when many sources are sending to a common destination at the same time. AI computing models inherently have a phase when data is aggregated after a processing iteration from which incast congestion (many-to-one) easily occurs. Incast is a network traffic pathology caused by many-to-one communication patterns that can lead to large packet loss and increased queuing delay. Incast can increase application latency and decrease application throughput to a point well below the characteristics of link bandwidth [10]. The problem especially affects AI training, where distributed processing cannot continue until all parallel threads in a stage complete. Increased application latency degrades the concurrency of the networked storage system which lowers the number of IOPS for the entire solution.

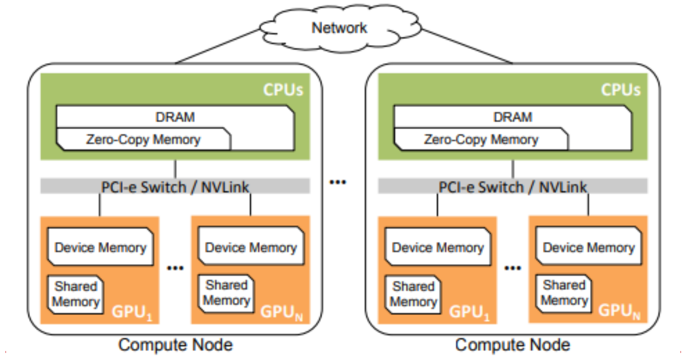
GPUs: Ultra-low latency network for parallel computing

~~As the number of AI algorithms and AI applications continue to increase, and the distributed AI computing architecture emerges, AI computing is implemented on a large scale. GPUs have ignited a worldwide AI boom. They have become a key part of modern supercomputing. They've been woven into a sprawling new hyperscale data centers. Still prized by gamers, they have become accelerators speeding up all sorts of tasks from encryption to networking to AI. GPUs provide much~~



~~Figure WW – Distributed AI Computing Architecture~~

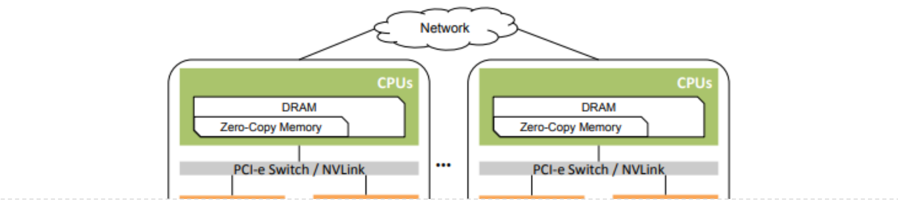
~~higher memory bandwidth than today’s CPU architectures. Nodes with multiple GPUs are now ubiquitous in high-performance computing because of their power efficiency and hardware parallelism. Figure X illustrates the architecture of typical multi-GPU nodes, each of which consists of a host (CPUs) and several GPU devices connected by a PCI-e switch or NVLink. Each GPU is able to directly access its local relatively large device memory, much smaller and faster shared memory, and a small pinned area of the host node’s DRAM, called zero-copy memory [11].~~

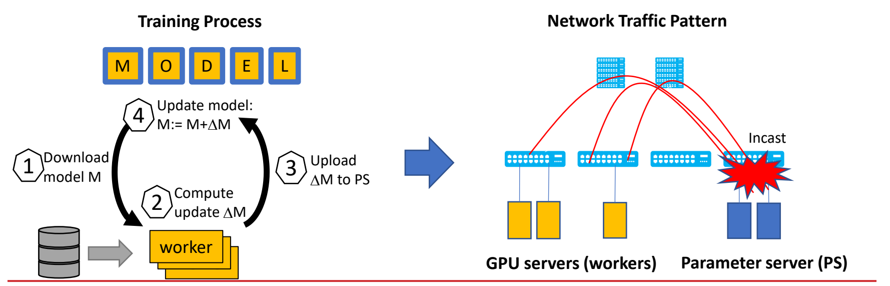


~~Figure WW2 – Distributed AI Computing Architecture~~

Today's AI computing architecture includes a hybrid mix of Central Processing Units (CPUs) and Graphics Processing Units (GPUs). GPUs, originally invented to help render video games at exceptional speeds, have found a new home in the data center. The GPU is a processor with thousands of cores capable of performing millions of mathematical operations in parallel. All AI learning algorithms perform complex statistical computations and deal with a huge number of matrix multiplication operations per second – perfectly suited for a GPU. However, to scale the AI computing architecture to meet the needs of today's AI algorithms and applications in a data center, the GPUs must be distributed and networked. This places stringent requirements on 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 underutilized resources on the 100 Gbit/s InfiniBand network. The presence of the network and network contention reduced the performance of the solution to less than linear scale [~~12~~11]. Consequently, network performance greatly restricts horizontal extension of the AI system.





~~Figure 12 – Periodic incast congestion during training~~

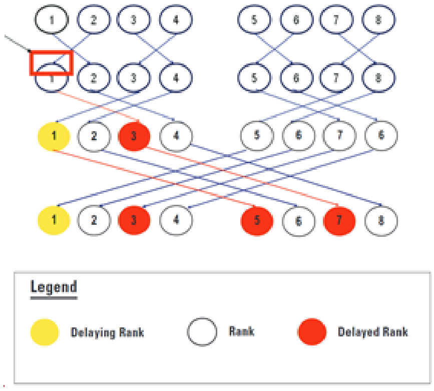
GPUs provide much higher memory bandwidth than today's CPU architectures. Nodes with multiple GPUs are now commonly used in high-performance computing because of their power efficiency and hardware parallelism. Figure 8 illustrates the architecture of typical multi-GPU nodes, each of which consists of a host (CPUs) and several GPU devices connected by a PCI-e switch or NVLink. Each GPU is able to directly access its local relatively large device memory, much smaller and faster shared memory, and a small pinned area of the host node's DRAM, called zero-copy memory [12].

GPUs are inherently designed to work on parallel problems. With AI applications, these problems are iterative and require a synchronization step that creates network incast congestion. Figure ~~12~~9 shows how incast congestion occurs with AI training. The training process is iterative and there are many parameters synchronized on each iteration. The workers download the model and upload newly calculated results (ΔM) to a parameter ~~servers at nearly~~server during a synchronization step. The uploading to the ~~same time.~~parameter server creates incast. When the computing time is improved by deploying faster GPUs, the pressure on the network and resulting incast increases.

The ~~high-bandwidth~~communication between the worker nodes and ~~low-latency DCN with only physical links cannot meet requirements~~the parameter server constitutes a collection of ~~large-scale and highly concurrent AI/HPC applications.~~interdependent network flows. In the iteration process of distributed AI computing, ~~a large amount of~~ many burst traffic ~~is~~flows are generated to distributed data to workers within milliseconds~~. In addition, because a parameter server (PS) architecture is used to update~~, followed by an incast event of smaller sized flows directed at the parameter ~~weights~~server when the intermediate parameters are delivered and updated. During the exchange of ~~the new model for data parallelization, the incast traffic model at a fixed time is easily formed. In this case,~~these flows packet loss, congestion, and load imbalance can occur on the network. As a result, the Flow Completion Time (FCT) of some ~~data~~of the flows is ~~too long. Distributed AI computing is synchronous~~prolonged. If a few flows are delayed, ~~more~~storage and computing ~~processes are affected~~resource can be underutilized. Consequently, the completion time of the entire application is delayed. ~~This~~

Distributed AI computing is synchronous, and it is desirable for the jobs to have a predictable completion time. When there is no congestion, dynamic latency across the network is small allowing the average FCT to be predictable and therefor the performance of the entire application is predictable. When congestion causes dynamic latency to increase to the point of causing packet loss, FCT can be very unpredictable. Flows that complete in a time that is much greater than the average completion contributes to what ~~we call the~~is known as tail latency. Tail latency is the small percentage of response times from a system, out of all of responses to the input/output (I/O) requests it serves, that take the longest in comparison to the bulk of its response times. ~~It~~Reducing tail latency as much as possible is ~~very~~extremely critical to the success of parallel algorithms and the whole distributed computing system. ~~Figure X shows how~~ To maximize the use of GPUs in the data center, tail latency ~~injures the whole system performance~~should be addressed.

~~Consequently, in order to minimize the FCT to complete the entire computing task, we need to reduce the tail delay as much as possible. Because the microbursts in data center network are within milliseconds, the tail delay needs to be controlled within milliseconds to ensure optimal system performance. Therefore, For HPC services, in order to have an ultra-low latency lossless network, the data center network should first solve the tail delay problem.~~



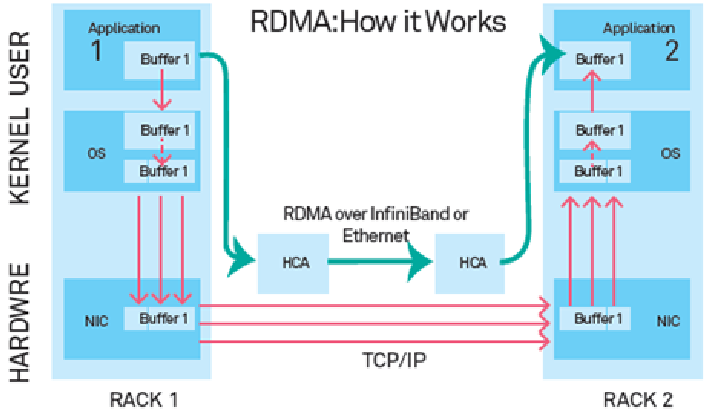
~~Figure X - Single-point delay causes the overall performance to deteriorate~~

SmartNICs

- ~~SmartNIC become the computer in front of computer~~
  - ~~SmartNIC is a NIC with all NIC functions regardless CPU/FPGA. Host CPU only request to install NIC driver.~~
  - ~~SmartNIC is a computer in front of computer. SmartNIC has independent OS and is able to run some applications independently.~~
    - ~~SmartNIC can be used to accelerate application~~
      - ~~Accelerate computing, storage…~~
    - ~~SmartNIC can be used to offload host CPU to run specific application more efficient~~
    - ~~SmartNIC is part of computing resource. Participate the application computing together with host CPU and GPU.~~
      - ~~Complement of CPU and GPU computing resource~~
      - ~~SmartNIC is not the replacement of CPU and GPU, major applications still run on CPU/GPU~~
    - ~~SmartNIC can be the independent domain than host domain and protect the host domain~~
      - ~~Offload OVS to SmartNIC to isolate the data classification from hypervisor~~
    - ~~SmartNIC can be emulated to other PCIe devices to support more advanced application~~
      - ~~NVMe emulation~~
  - ~~SmartNIC is programmable and easy use~~
    - ~~Open source software, major Linux~~
    - ~~Easy to program, no special request for programmer~~
  - ~~SmartNIC is not proprietary NIC, one NIC fits many applications, easy for user to program~~

~~RDMA~~

Over the years there have been periods of time when performance improvements in CPU speeds and Ethernet links have eclipsed one another. Figure 10 shows the historical performance gains with Ethernet link speeds [13] and benchmark improvements for CPU performance [14]. During some historical periods, the processing capability of a traditional CPU was more than enough to handle the load of an Ethernet link and the cost savings of a simplified network interface card (NIC) along with the flexibility of handling the entire networking stack in software was a clear benefit. During other periods, the jump in link speed from the next iteration of IEEE 802.3 standards was too much for the processor to handle and a more expensive and complex SmartNIC with specialized hardware offloads became necessary to utilize the Ethernet link. As time goes on and the SmartNIC offloads mature, some of them become standard and included in the base features of what is now considered a common NIC. This phenomenon was seen with the advent of the TCP Offload Engine (TOE) which supported TCP checksum offloading, large segment sending and receive side scaling.



~~Figure E - Working principle of RDMA~~

In today’s world, there are signs of Moore’s law fading while Ethernet link speeds continue to soar. The latest iteration of IEEE 802.3 standards is achieving 400 Gbps. Couple this divergence with the added complexity of software-defined networking, virtualization, storage, message passing and security protocols in the modern data center, and there is a strong argument that the SmartNIC architecture is here to stay. So, what exactly is a data center SmartNIC today?

Figure 11 shows a data center server architecture including a SmartNIC. The SmartNIC includes all the typical NIC functions, but also includes key offloads to help accelerate applications running on the server CPU and GPU. The SmartNIC does not replace the CPU or the GPU but rather complements them with networking offloads. Some of the key offloads include virtual machine interface support, flexible match-action processing of packets, overlay tunnel termination and origination, encryption, traffic metering, shaping and per-flow statistics. Additionally, SmartNICs often include entire protocol offloads and direct data placement to support RDMA and NVMe-oF storage interfaces.

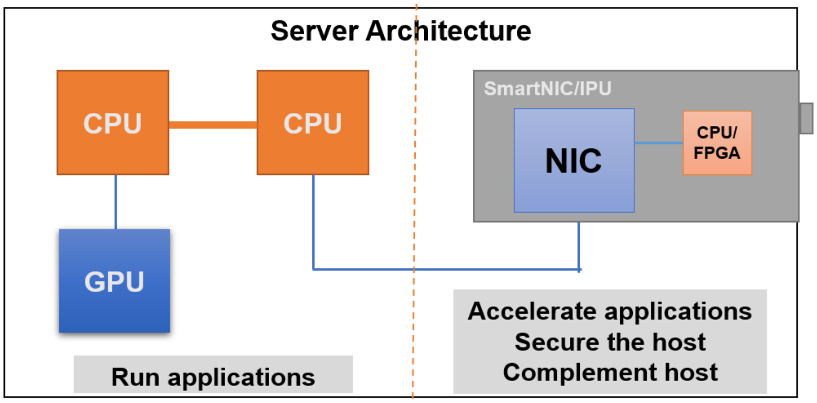


Figure 11 – Server Architecture with SmartNIC

One new critical component of today's SmartNIC is programmability. A criticism of SmartNICs in the past was their inability to keep pace with the rapidly changing networking environment. The early cloud data center environments favored using the CPU for most networking functions because the required feature set for the NIC was evolving faster than the development cycle of the hardware. Today's SmartNICs however have an open and flexible programming environment. They are essentially a computer in front of the computer with an open source development environment based on Linux and other software-defined networking tools such as Open vSwitch [15]. It is essential that SmartNICs integrate seamlessly into the open source ecosystem to enable rapid feature development and leverage.

SmartNICs in the data center increase the overall utilization and load on the network. They can exacerbate the effects of congestion by fully and rapidly saturating a network link. At the same time, they can respond quickly to congestion signals from the network to alleviate intermittent impact and avoid packet loss. The programmability of the SmartNIC allows it to adapt to new protocols that can coordinate with the network to avoid conditions such as incast.

RDMA

RDMA (Remote Direct Memory Access) is a new technology designed to solve the problem of server-side data processing latency in network applications, which transfers data directly from one computer's memory to another without the intervention of both operating systems. This allows for high bandwidth, low latency network communication and is particularly suitable for use in massively parallel computer environments. ~~By transferring telegrams~~ RDMA allows the transfer of data directly into the storage space of ~~the other~~another computer~~ through the network, data can be quickly transferred from one system to the storage space of another system~~, reducing or eliminating the need for multiple copies of the data ~~telegrams~~ during transmission~~, thus freeing~~. This frees up memory bandwidth and CPU cycles ~~and~~to greatly ~~improving~~improve system performance. Figure ~~E~~12 shows the ~~principle~~principles of the RDMA protocol. There are three different transports for the RDMA protocol: Infiniband, iWarp and RoCEv1/RoCEv2.

~~RDMA's development in the transport layer/network layer currently goes through 3 technologies, Infiniband, iWarp and RoCEv1/RoCEv2.~~

Infiniband

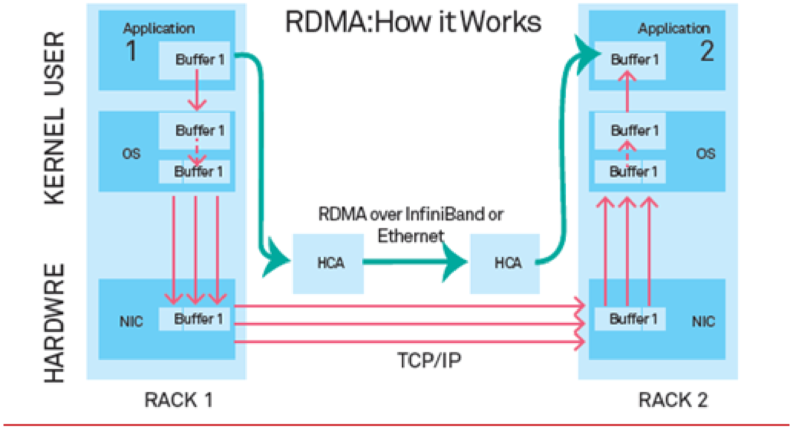
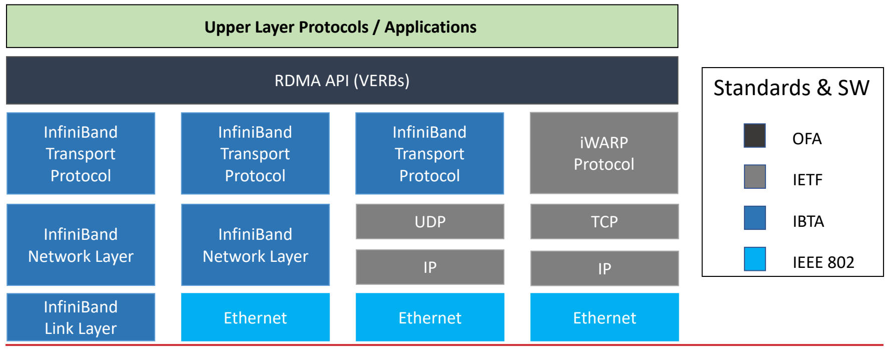


Figure 12 - Working principle of RDMA

In 2000, the ~~IBTA (~~InfiniBand Trade Association (IBTA) released the ~~first~~initial support for RDMA ~~technology~~, Infiniband, which is a ~~customized~~ network technology customized for RDMA ~~multi-layered, new design from the~~through a specific hardware ~~perspective~~design to ensure the reliability of data transmission. ~~The~~ InfiniBand ~~technology uses~~allows RDMA ~~technology~~ to ~~provide direct~~directly read and write ~~access to~~the memory of remote nodes. ~~RDMA used InfiniBand as the transport layer in its early days, so it must use InfiniBan~~ Infiniband is a unique network solution requiring specific Infiniband switches and ~~InfiniBan network~~ Infiniband interface cards~~ to implement~~.

iWarp



~~Figure xx – RDMA protocol stacks and standards – NOTE: SNIA Copyright~~

~~Internet wide area~~An RDMA protocol~~, also known as RDMA~~ that runs over TCP~~ protocol, is~~ , allowing it to traverse the ~~IEEE/~~Internet and wide area, has been defined by the IETF ~~proposed RDMA technology. It uses the TCP protocol to host~~ and is known as iWarp. In addition to the ~~RDMA protocol. This~~ wide area, iWarp also allows RDMA to ~~be used in~~run over a standard Ethernet ~~environment (switch) and the~~ network ~~card requirement is an iWARP enabled network card. In fact iWARP~~and within a data center. While iWarp can be implemented in software, ~~but this takes away~~to obtain the desired performance ~~advantage~~ of RDMA~~.~~ special iWarp enabled NIC card are used.

RoCE (RDMA over Converged Ethernet)

In April 2010, the IBTA released the RoCEv1 specification, which ~~was released as an add-on to~~augments the Infiniband Architecture Specification~~, so it is also known as IBoE (~~ with the capability of supporting InfiniBand over Ethernet~~).~~ (IBoE). The ~~RoCE~~RoCEv1 standard ~~replaces the TCP/IP~~specifies an Infiniband network layer ~~with an IB network layer~~directly on top of the Ethernet link layer~~ and~~. Consequently, the RoCEv1 specification does not support IP routing. ~~The Ethernet type is 0x8915. in RoCE, the link layer header of the infiniband is removed and the GUID used to represent the address is converted to an Ethernet MAC. infiniband~~Since Infiniband relies on a lossless physical transport, ~~and RoCE relies on~~the RoCEv1 specification depends on a lossless Ethernet ~~transport~~environment.

RoCEv2

~~Since the RoCEv1 data frame does not have an IP header, it can only communicate within a 2-tier network. To solve this problem, in 2014 IBTA proposed RoCE V2, which extends RoCEv1 by replacing GRH (Global Routing Header) with a UDP header + IP header. Because RoCE v2 packets are routable at Layer 3, they are sometimes referred to as "Routable RoCE" or "RRoCE" for short. As shown in the figure below.~~

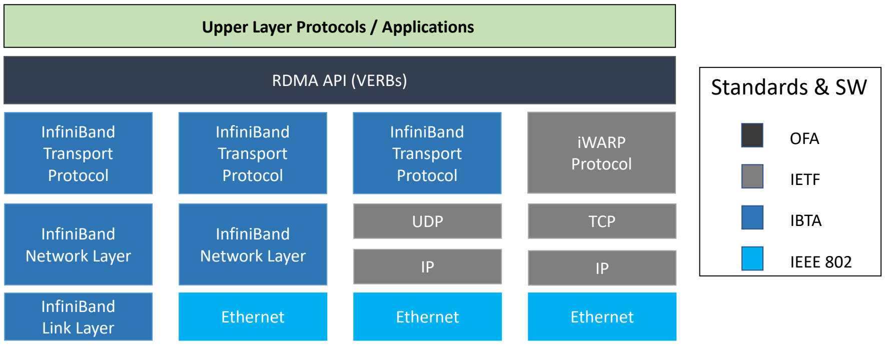


Figure 13 – RDMA protocol stacks and standards

~~RoCE technology can be implemented through a~~ Modern data centers tend to use layer-3 technologies to support large scale and greater traffic control. The RoCEv1 specification required an end-to-end layer-2 Ethernet transport and did not operate effectively in a layer-3 network. In 2014, the IBTA published RoCEv2, which extended RoCEv1 by replacing the Infiniband Global Routing Header (GRH) with an IP and UDP header. Now that RoCE is routable it is easily integrated into the preferred data center environment. However, to obtain the desired RDMA performance, the RoCE protocol is offloaded to special network interface cards. These network cards implement the entire RoCEv2 protocol, including the UDP stack, congestion control and any retransmission mechanisms. While UDP is lighter weight than TCP, the additional support required to make RoCEv2 reliable adds complication to the network card implementation. RoCEv2 still depends upon the Infiniband Transport Protocol, which was designed to operate in a lossless Infiniband environment, so RoCEv2 still benefits from a lossless Ethernet environment.

~~Technology~~	~~Data 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 interface~~	~~Compatibility with traditional Ethernet technologies, cost-effectiveness, and good performance~~	~~Specific NICs required~~ ~~Still have many challenges to~~
~~Omni-Path~~	~~100~~	~~100 ns~~	~~OPA network architecture~~ ~~Verbs programming interface~~	~~Good performance~~	~~Single manufacturer and specific NICs and switches required~~

~~Table X - Compares RDMA Network Technologies~~

RoCE/RoCEv2	40, 56, 100, or 200	300-500 ns	or transport layer and Ethernet link layer Verbs programming interface	traditional Ethernet technologies, cost-effectiveness, and good performance	Specific NICs required Still have many challenges to
Omni-Path	100	100 ns	OPA network architecture Verbs programming interface	Good performance	Single manufacturer and specific NICs and switches required

Table 1 – Comparison of RDMA Network Technologies

Figure 13 shows the most common ~~Ethernet switch, but the server needs to support RoCE network cards. Since RoCEv2 is a UDP protocol, although the UDP protocol is relatively high efficiency, but unlike the TCP~~ RDMA protocol~~, there is a retransmission mechanism to ensure reliable transmission, once there is a packet loss, must rely on the upper layer~~ stacks and their associated standards bodies. Table 1 compares the details of ~~the application found and then do retransmission, which will greatly reduce the transmission efficiency of RDMA. So in order to play out the true effect of RoCE, it is necessary to build a lossless network environment for RDMA without losing packets.~~

~~RoCE can run in both lossless and compromised network environments, called Resilient RoCE if running in a compromised network environment, and Lossless RoCE if running in a compromised network environment.~~

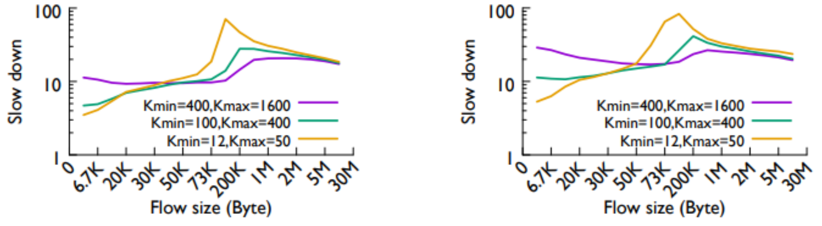
different implementations. RDMA is more and more widely used ~~in market, especially in OTT companies. There have been~~ to support high-speed storage, AI and Machine Learning applications in large scale cloud data centers. There are real world examples of tens of thousands of servers ~~supporting~~running RDMA~~, carrying our databases, cloud storage, data analysis systems, HPC and machine learning applications in~~ in production. Applications have reported impressive performance improvements by adopting RDMA [~~13~~16]. For instance, distributed machine learning training has been accelerated by 100+ times compared with the TCP/IP version, and the I/O speed of SSD-based cloud storage has been boosted by about 50 times compared to the TCP/IP version. These improvements majorly stem from the hardware offloading characteristic of RDMA.

4 Challenges with today's data center network

High ~~bandwidth~~throughput and low latency tradeoff

~~When we talk about data center network performance, low latency and high bandwidth are always the ultimate goals of data center network users. In the previous chapter, we introduced the features of the RDMA protocol. The zero copy characteristic help applications can perform data transfers without the involvement of the network software stack. Data is sent and received directly to the buffers without being copied between the network layers. In addition, the RDMA applications can perform data transfers directly from user-space without kernel involvement, which is called kernel bypass. Though "high bandwidth and low latency" has become a "catchphrase" of RDMA, we find it is practically hard to achieve them simultaneously in one DCQCN configuration. This is because for consistently low latency the network needs to maintain steadily small queues in buffers (which means low ECN marking thresholds), while senders will be too conservative to increase flow rates if ECN marking thresholds are low.~~

Simultaneously achieving both low latency and high throughput in a large-scale data center is difficult. To achieve low latency, it is necessary to allow flows to begin transferring at line rate while at the same time maintaining near empty switch queues. Aggressively starting flows at line rate will allow them to consume all available network bandwidth instantly and can lead to extreme congestion at convergence points in the network. Deep switch buffers absorb temporary congestion to avoid packet loss but delay the delivery of latency sensitive packets. Using a low ECN marking threshold can help slow aggressive flows and keep switch queue levels empty, but this reduces throughput. High throughput flows benefit from larger switch queues and higher ECN marking thresholds in order to not overreact to temporary congestion and slow down unnecessarily.

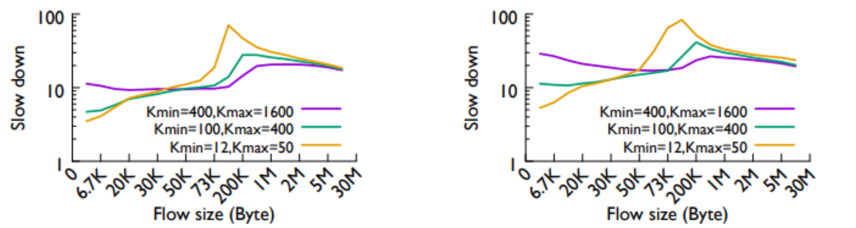


(a) 30% network load.

(b) 50% network load.

~~Figure X – 95-percentile FCT slowdown distribution with different ECN thresholds, using WebSearch~~

Experimentation shows the tradeoff still exists after varying algorithms, parameters, traffic patterns and link loads [~~13~~16]. Figure ~~X approximately~~14 from [16] shows ~~the issue. It shows the~~ how flow completion times (FCT ~~slowdown with~~) are extended beyond the theoretical minimum FCT when using different ECN marking thresholds (Kmin, Kmax) in switches and using an RDMA WebSearch application as the input traffic ~~loads. Figure Xa shows that when we use~~load. Lower values for Kmin and Kmax will cause ECN markings to occur more quickly and force a flow to slow down. As seen in the figure, when using low ECN thresholds, small flows which are latency-sensitive have lower slowdown in FCT, while big flows which are typically bandwidth-~~sensitive~~hungry suffer from larger FCT slowdown. The trend is more obvious when the network load is higher (Figure ~~Xb~~14-b when the average link load is 50%).



(a) 30% network load.

(b) 50% network load.

Figure 14 – FCT slowdown distribution with different ECN thresholds, using WebSearch

~~For instance, the 95th-percentile RTT is about 150μs — 30 (slowdown) × 5μs (baseline RTT) — when Kmin = 400KB, Kmax = 1600KB, which is a lot worse than the ML application's requirement (<50μs). We have tried out different DCQCN parameters, different average link loads and different traffic traces, and the trade-off between bandwidth and latency remains.~~

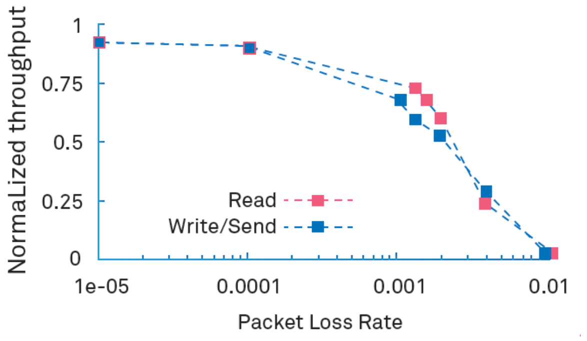
- ✓ ~~It's difficult to achieve high bandwidth and low latency simultaneously~~
- ✓ ~~Experimentation shows the tradeoff still exists after varying algorithms, parameters, traffic patterns and link loads~~
- ✓ ~~Reason explanation about why tradeoff exists~~

Deadlock free lossless network

RDMA advantages over TCP include low latency, high throughput, and low CPU usage. However, unlike TCP, RDMA needs a lossless network; i.e. there ~~must~~should be no packet loss due to buffer overflow at the switches [~~14~~17]. The RoCE protocol ~~is based~~runs on top of UDP ~~and~~with a go-back N retransmission strategy that severely impacts performance if invoked. As such, RoCE requires Priority-based Flow Control (IEEE Std 802.1Q-2018, Clause 36 [~~15~~18])~~-~~ to ensure that no packet loss occurs in the ~~entire~~ data center network. ~~Packet loss in the network severely affects the RoCE service performance. As shown in~~ Figure ~~X, the~~15 show how RoCE service throughput decreases

Commented [PC2]: Where is this figure from?

rapidly with an increase in the packet loss rate. ~~As~~ Loosing as little as one ~~thousandth of packet lost in the network,~~in one thousand packets decreases RoCE service performance by roughly 30%.



~~Figure X – Impact of packet loss on RDMA throughput~~

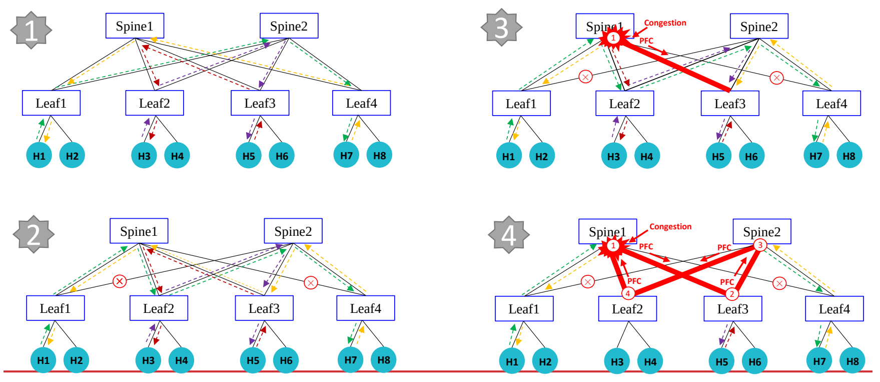
Figure 13 – Impact of packet loss on RDMA throughput

Priority-based Flow Control (PFC) prevents packet loss due to buffer overflow by pausing the upstream sending device when the receiving device input buffer occupancy exceeds a specified threshold. While this provides the necessary lossless environment for RoCE, there are problems with the large-scale use of PFC. One such problem is the possibility of a PFC deadlock.

Deadlocks in lossless networks using backpressure flow control such as PFC have been studied for many years [19, 20, 21]. A PFC deadlock occurs when there is a cyclic buffer dependency (CBD) among switches in the data center network. The CBD is created when buffers in a sequence of switches are waiting on buffers in other switches of the sequence to have capacity before a dependent switch can transmit a packet. If the switches involved in the CBD are using PFC and are physically connected in a loop, a PFC deadlock can occur. RDMA flows in the data center network are distributed across multiple equal cost paths to achieve the highest possible throughput and

Commented [PC3]: We should include a number of references that describe the issue at this point.

Commented [PC4R3]: ❖Tagger: Practical PFC Deadlock Prevention in Data Center Networks
❖David Lee Paper and other related references to deadlocks



~~Figure KK – PFC Deadlock~~

lowest latency. ~~These~~While there are no loops in the logical topology, these paths naturally contain loops in the physical topology. A PFC deadlock in the network can completely halt network traffic.

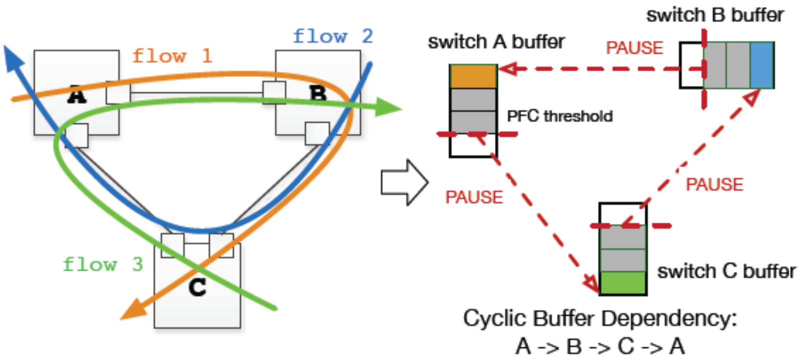
Consider the example in Figure ~~KK~~16. The figure shows four phases of PFC deadlock creation. In phase 1, four flows are equally load balanced across the Clos fabric and the network is running smoothly. In phase 2, the red cross indicates a transient or permanent fault in the topology, such as link failure, port failure, or route failure. Due to the failure, in the example, traffic between H1 and H7 (green line) and between H3 and H5 (purple line) is re-routed. The re-routing pushes more traffic through leaves 2 and 3 causing a potential overflow in spine 1 as shown in phase 3. To avoid loss, the spine 1 switch issues PFC towards leaf 3, shown in phase 3. Traffic in leaf 3 now backs up, causing further backups around the topology and a cascade of PFC messages along the loop backward towards the original point of congestion. Phase 4 shows the resulting PFC deadlock.

When the network size is small, the probability of PFC deadlock is low. However, at larger scale and with the high-~~-~~performance requirements of the RoCE protocol, the probability of PFC deadlock increases ~~exponentially~~significantly. Achieving larger scale and optimal performance is a key objective of the intelligent lossless data center network of the future. Section 5 discusses a possible new technology for PFC deadlock prevention.

~~A number of techniques have been explored to address the PFC deadlock problem [16]. <<Finish solution discussion>>~~

~~<< Consider including the following additional text that explains the CDB issue if necessary. The text above Figure KK tries to summarize CDB as concisely as possible. If we want more detailed description of the CDB issue, consider inserting this text somewhere above Figure KK and its associated paragraph >>~~

~~Before introducing detail information about PFC deadlock problem, a new term should be described first. It is called Cyclic Buffer Dependency (CBD) [17]. CBD means the occupied buffers are waiting for each other in a loop. A simple example is shown in Figure x.~~



~~Figure x – A simple but contrived example to illustrate CBD~~

Congestion control issues in large-scale data center networks

~~With the popularization of RDMA applications, networks face a new challenges: How to implement large-scale RDMA applications? When the~~ RDMA technology ~~is~~was initially used by customers~~,~~ ~~customers are~~ in constrained, conservative, ~~so they use RDMA only in the POD. In this way, the~~

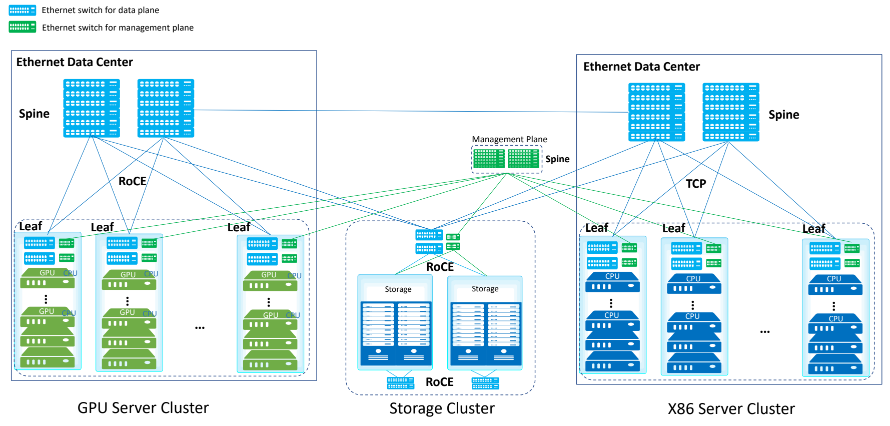
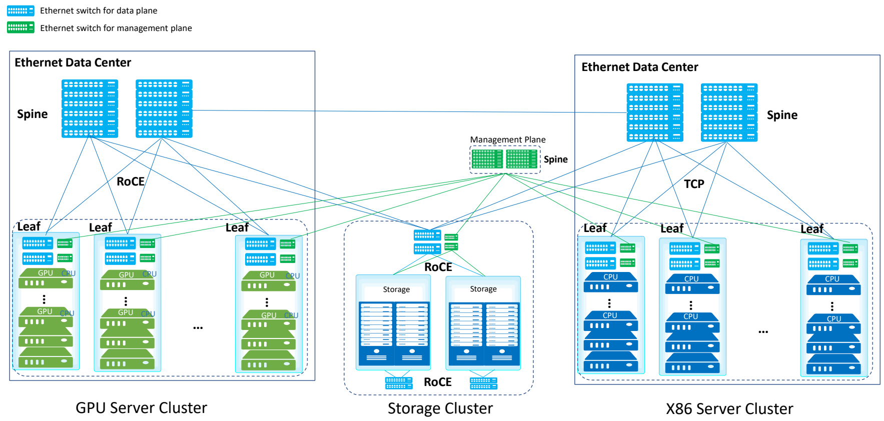


Figure 17 – RoCE application in large-scale data center networks

~~application scenarios of RDMA are limited to dedicated scenarios,~~small scale environments such as cluster computing or targeted storage networks.

Tuning the resources required for the dedicated environment was manageable by the network operator, at least to some degree. However, ~~with the application of RDMA, its~~ the performance advantages ~~are obvious. The customer began to think about a new question: How to use the RDMA network across PODs?~~of RDMA have proven useful in many application environments and there is a strong desire to use RDMA in a large-scale. Figure ~~X is a schematic diagram~~17 shows an example of a large-scale RoCE network~~ application. As shown in~~. In the ~~picture~~example, the ~~customer's~~ entire data center network is ~~deployed~~ based on ~~the~~ Ethernet. The computing cluster and storage cluster use the RDMA protocol while the X86 server cluster ~~use~~uses traditional TCP/IP~~ protocol~~.

In the large-scale data center network scenario ~~shown in figure x above, a new scenario is inevitable. That is~~ TCP and RoCE traffic ~~is transmitted mixed together in~~ can traverse common parts of the network~~. Currently, there are~~ in several ~~traffic hybrid scenarios.~~different ways:



~~Figure X – RoCE application in large-scale data center networks~~

Scenario 1: ~~For example, when~~A traditional web-based application with a high-speed storage backend expects an end user ~~needs to invoke some stored data from cloud, user needs to~~ to submit a request from the Internet to the web service ~~through~~using TCP. ~~Then the~~ The web service cluster ~~will invoke~~may fetch the shared storage ~~node through~~using additional TCP ~~protocol~~connections. When the storage ~~cluster~~front-end receives the request~~ of data reading~~, it uses the RoCE protocol to handle ~~reading work in order to make sure the network has very high IOPS. After~~ the actual reading of the shared data from the medium with the expectation of obtaining extremely high IOPS using RDMA. The shared data ~~user needed, storage cluster~~ will ~~send data back~~be returned to ~~web server and to~~the end user~~ finally, using~~, again with TCP.

Scenario 2: ~~The service network of the~~ More highly integrated computing~~/~~ and storage ~~cluster uses~~clusters use the RoCE ~~network~~protocol for the bulk of their communication, while the management ~~traffic (from the~~ and any SDN control of the overall infrastructure is based on TCP/IP.

All nodes need TCP connections for management ~~switch in green in figure x or from the SDN controller) in the DC is basically based on the TCP protocol. Therefore,~~ and control, so the two types of traffic ~~may run together~~will traverse common links in the ~~cluster~~network.

Scenario 3: Although RoCE has been gradually used in large scale computing and storage network. However there are still TCP-based storage and computing data centers. Therefore, in large-scale data center applications, there may be multiple combinations of TCP or RoCE between computing and computing, between storage and storage, and between computing and storage.

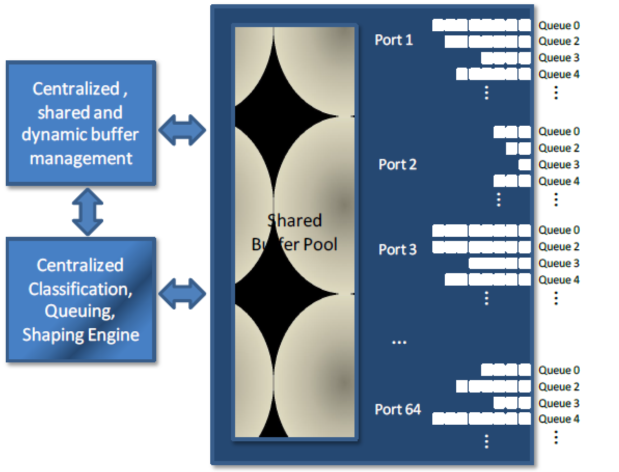
Commented [PC5]: We need a more specific scenario or delete the case and it isn't offering anything specific. Perhaps there is something in an AI application architecture that requires both TCP and RDMA together.

~~A large amount of~~ In theory, separating TCP and RoCE traffic ~~exists on~~within the network~~. Therefore, the~~ should be easy. IEEE Std 802.1Q defines 8 classes of service that can map to 8 queues with differing queue scheduling algorithms. Different switch ~~needs~~queues can be used to ~~use~~isolate the different ~~queues to schedule different types of~~ traffic~~.~~ types. The ~~IEEE 802.1P~~queues and ~~IEEE802.1Qaz (ETS) mechanisms~~the buffer management are ~~used for scheduling. These mechanisms need to be~~ implemented in hardware on the switch chip~~. The switch chip of the current data center encounters the Switch packet buffer~~ , but there is a performance and cost tradeoff problem. ~~Microburst is a common problem in current data center network. To prevent packet loss caused by microbursts,~~ Allocating sufficient dedicated memory to each queue on each port ~~of the switch should be configured with enough buffers~~ to absorb ~~the burst. On the other hand, for the chip implementability, the cost is~~ microbursts of traffic without incurring packet loss can be too ~~high for a switch to implement purely static~~ expensive and technically challenging as the number of ports per~~-port buffer allocation schemes.~~

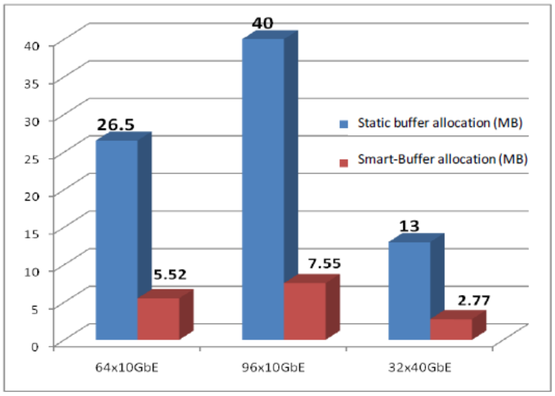
~~Based on the multi-queue technology, mainstream switching~~ switch chip goes up. To address this tradeoff, switch chip vendors ~~use~~implement a smart ~~buffer~~buffering mechanism ~~to solve the tradeoff problem. One of the~~ that allows for a hybrid of fixed and shared buffers.

A core ~~ideas~~idea of smart ~~buffer~~buffering is ~~to put forward~~the creation of a ~~concept of~~ dynamic shared buffer. ~~Smart-buffer mechanisms attempt~~The goal is to optimize buffer utilization and burst absorption~~, Dynamic sharing~~ by reducing the amount of statically dedicated buffers while providing a dynamic and self-tuning ~~is transparently enabled~~shared pool across all ports to handle temporary bursts [~~18~~22].

~~The Smart-Buffer~~ An example smart buffer architecture, as shown in Figure ~~1, takes into consideration~~18. Each port has some dedicated buffers for each of its queues and a dynamic pool of surplus buffers shown in gray. The approach considers that congestion in a typical data center environment is localized to a subset of egress ports at any given point in time and ~~realistically never happens~~rarely occurs on all ports simultaneously. This ~~enables its~~assumption allows the centralized on-chip buffer to be right-sized for overall cost and power~~; at the same time, the buffer is dynamically shareable and weighted towards~~ consumption while still providing resources for congested ports exactly when needed using self-tuning thresholds~~. Dynamic sharing and self-tuning is transparently enabled across all ports, regardless of the processing pipeline or switching core to which the ports belong~~.


Centralized , shared and dynamic buffer management
Centralized Classification, Queuing, Shaping Engine
Shared Buffer Pool
Port 1
Port 2
Port 3
...
Port 64
Queue 0
Queue 2
Queue 3
Queue 4

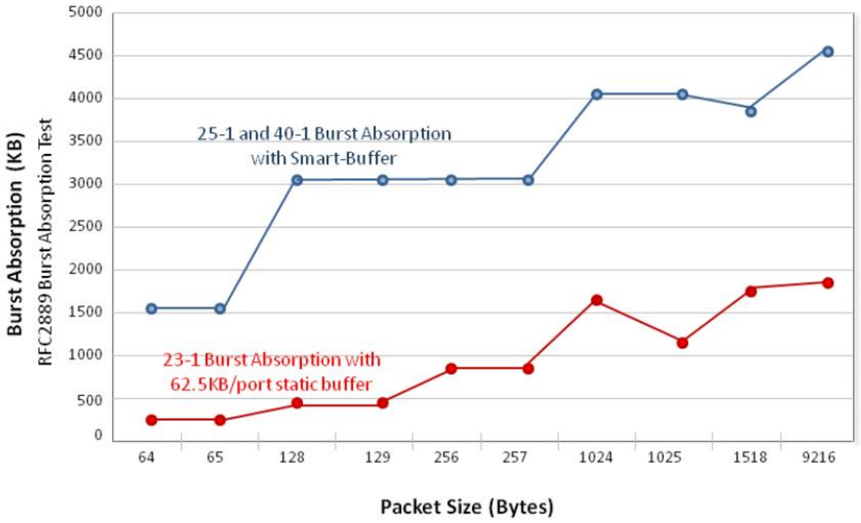
~~Figure x – Smart-Buffer centralized and dynamic buffer management~~



~~Figure X – Smart-Buffer delivers up to 5x better packet buffer utilization~~

~~Switch Chip Company did some experiments to compare the effects of static and dynamic buffers. Figure X shows the buffer utilization comparison.~~

Contrasted with static per-port buffer allocation schemes found in other switch ~~device~~ architectures, ~~Smart-Buffer~~ the smart buffer approach significantly improves buffer utilization and enables ~~maximum~~better performance ~~per unit of packet buffer~~ for data center applications. ~~In data~~



~~Figure X – Burst Absorption Test: 3 to 6x worse burst absorption capacity in 10GbE switch with per-port static buffer architecture as compared to a Smart-Buffer based switch~~

~~center traffic tests and simulations, results showed, on average, five times better packet buffer utilization with Smart-Buffer.~~

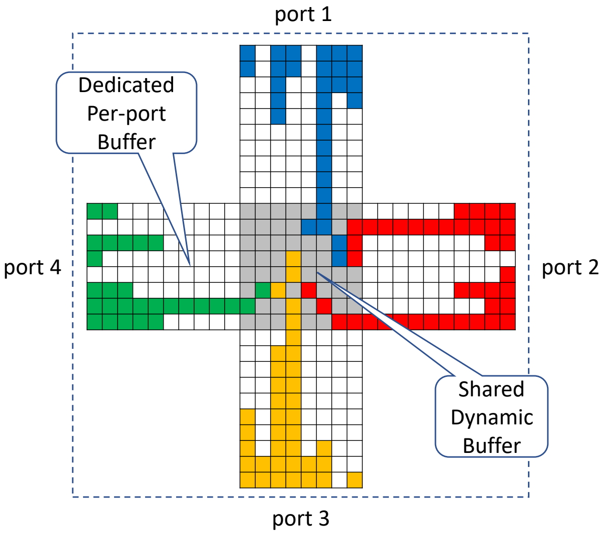
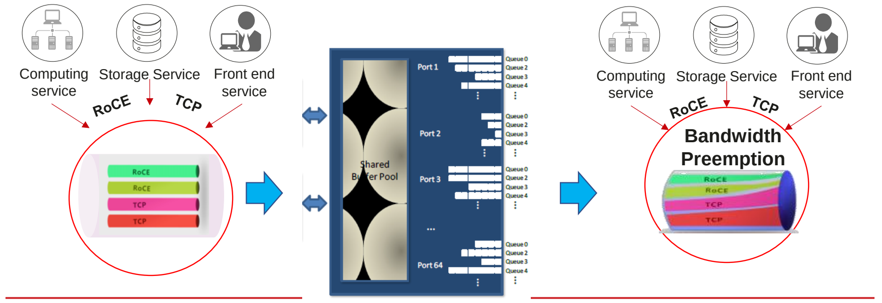


Figure 18 – Smart-Buffer centralized and dynamic buffer management

~~For burst~~ However, the shared dynamic pool has consequences on traffic class isolation in congested situations. TCP and RoCE flows may impact one another when they traverse common links, even if they are using separate traffic classes on those links. TCP and RoCE use different congestion control mechanisms, different re-transmission strategies and different traffic class configuration (lossless verse lossy). The algorithms and configurations do not allow a fair share of the common resource. Figure 19 shows the problem~~, the smart buffer mechanism improve 3 to 6 times better than the static buffer. In Figure x,~~. Network operators allocate the ~~red line denotes~~ network bandwidth to different traffic classes based on the ~~industry standard RFC 2889 burst absorption performance for a 10GbE switch architecture with per-port static buffering. The data is produced under a 23-to-1 burst condition (23 ingress ports bursting to a single egress destination).~~

~~In summary, smart-buffer delivers optimal buffer utilization and burst absorption for data center workloads by taking a holistic approach to buffer management – using real-life data center traffic scenarios to maximize overall throughput and lossless behavior.~~

~~However, for~~ service requirements of the ~~TCP and RoCE flow mix scenario, the smart buffer mechanism~~network. But over time and during periods of ~~switch chip may cause an unexpected problem. That is SLA (Service Level Agreement)~~congestion the bandwidth allocations cannot be ~~guarantee when TCP and RoCE traffic coexists. Because of different congestion control mechanism, the TCP flows and preempt the bandwidth of RoCE flows, even when using separate traffic classes. Figure X shows the schematic diagram of the problem. At the beginning, network bandwidth is allocated to different flows based on service requirements in initialization phase. But due to shared buffer architecture and different~~ met. The different congestion control methods~~,~~ create different traffic behavior that impacts the smart buffering mechanism's ability to fairly allocated the dynamic shared buffer pool. In this case, TCP preempts RoCE bandwidth~~.~~, even when it is allocated to separate traffic classes. The RoCE flow completion delay ~~increased~~ has been seen to increase by 100 times ~~(ms level)~~.



~~Figure X – Schematic diagram of traffic coexistence~~

ODCC conducted several tests to verify the problem of traffic coexistence. ~~Table X is the test specification on flow mix scenario.~~

~~<< Test tables omitted – too much detail for this paper, need to reference test results, papers, published ODCC documents. Summarize the test findings instead of referencing specific details. >>~~

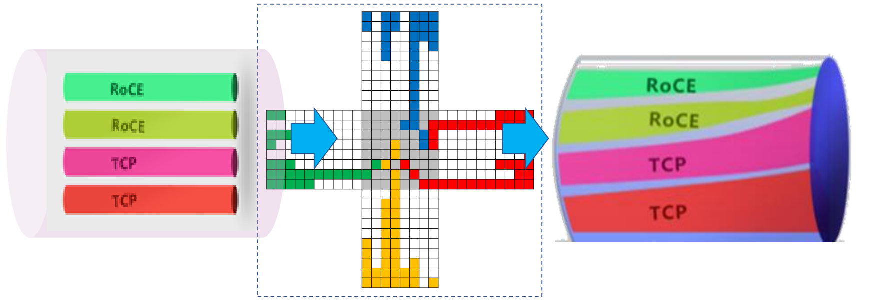


Figure 19 – TCP and RoCE coexistence with smart buffering

~~The data in red in the preceding test results is abnormal performance data. From test results that we can see:~~

~~In a scenario where both RoCE and TCP flows are transmitted, TCP flows preempt the bandwidth of RoCE flows. In other words, a traffic bandwidth proportion of the two at the receiver is different from that is set at the sender. Especially when the TCP traffic proportion increases (ROCE:TCP=9:1, message size = 64 KB), the effect of RoCE preemption is especially obvious, and the traffic proportion deviation reaches more than 50%. As described above, the smart buffer mechanism of the chip enables the TCP and RoCE to share the shared buffer. In addition, the Congestion Control mechanisms for TCP and RoCE traffic are different. The TCP packet loss threshold is higher than the CNP threshold of RoCE. As a result, TCP preempts more buffers. These two reasons cause it easier for TCP to preempt the bandwidth. Therefore, TCP congestion control is more aggressive than RoCE. To ensure the QoS of RoCE traffic, a proper ECN threshold for RoCE need to be configure.~~

~~As the number of QPs on the data center RDMA network increases, when the proportion of RoCE is small (e.g. TCP:ROCE = 9:1), the latency increases sharply (from microsecond level to millisecond level).In this case, all packets are marked with ECN, the queue is still overstocked, and the ECN becomes invalid. When the number of QP flows is large (for example, 4096 in the test), the bandwidth allocated to each flow is small. As a result, the packet interval of each flow (that is, the minimum interval of CNP packets that can be obtained by the flow) is greater than the rate increase interval. In this case, the rate of the flow is still increased even network is already congested. As a result, the rate control fails and the delay increases sharply. In the test result, when TCP:RoCE is 9:1 and the number of QPs is 4096, the maximum network latency is over 9 ms. This problem is easy to occur in a large-scale data center. For example, if there are 64 servers in a data center, at least 63 QPs are required for connecting one server to other 63 servers. At least 63 x 64 = 4032 QPs are required for 64 servers.~~

- ~~✓ How large scale today's data center is?~~
- ~~✓ Use cases for TCP and RoCE flows mixture~~
- ~~✓ Smart buffer mechanisms in mainstream switch chips~~
- ~~✓ SLAs cannot be guarantee when TCP and RoCE traffic coexists~~

Configuration complexity of congestion control algorithms

- ✓ Tuning RDMA networks is an important factor to achieving high-performance
- ✓ Current method of parameters configuration can be a complex operation
- ✓ Congestion control algorithms usually requires collaboration between the NIC and switch
- ✓ Traditional PFC manual configuration needs complex calculation with lots of parameters
- ✓ Excessive headroom leads to reduce the number of lossless queues while too little headroom leads to packet loss

5 New technologies to address new data center problems

Approaches to PFC storm elimination

Although traffic on the Clos network is up-down and loop-free, rerouting occurs when a transient or permanent link fault occurs, down-up traffic may be generated. (In large-scale data centers, down-up traffic is common. According to Microsoft's paper, the proportion of down-up traffic is about 10-5 [~~17~~21].) The reroute probability is around 10−5. Though 10−5 is not a big number, given the large traffic volume and the large scale data center networks, the deadlocks due to packet reroute do not just exist in paper designs. They are real! Especially with the scale of the data center RDMA network increases, RDMA starts to be deployed across PODs in large data centers. The larger the scale, the higher the probability of PFC deadlock, and the lower the service availability.

ODCC proposes a mechanism to prevent the deadlock problem. The deadlock-free technology and algorithm is a PFC-based deadlock prevention technology, which prevents deadlocks and eliminates PFC-based deadlocks in data center network. According to the preceding description, an important prerequisite for PFC deadlock is the occurrence of CBD loops. Therefore, to prevent PFC deadlock, it is a very important to discover and avoid CBD loops.

The core idea of the deadlock-free algorithm is to break the CBD deadlock. An innovative distributed topology role auto-discovery protocol is used to identify network locations and roles and identify abnormal traffic. The deadlock free algorithm mechanism consists of two algorithms: control plane algorithm and forwarding plane algorithm.

The following figure X shows the working principle. For example, in the figure, after the link from 24 to 11 goes down, green traffic is rerouted from server D to TOR24, to switch 1, to switch 2, to switch 3, to TOR switch 21, and finally to server A. This is the situation of down-up flow as we describe above. So the switch 2 can recognize itself as a CBD point. We propose a method to implement distributed self-learning of switches and obtain the switch level and port uplink/downlink attribute by using the extended LLDP protocol and algorithm. All switches learn the level and uplink/downlink attributes of their locations, and maintain and update the attributes automatically. This technology now is standardizing in the IEEE 802.1Qcz.More detail information will be discussed in chapter 6.

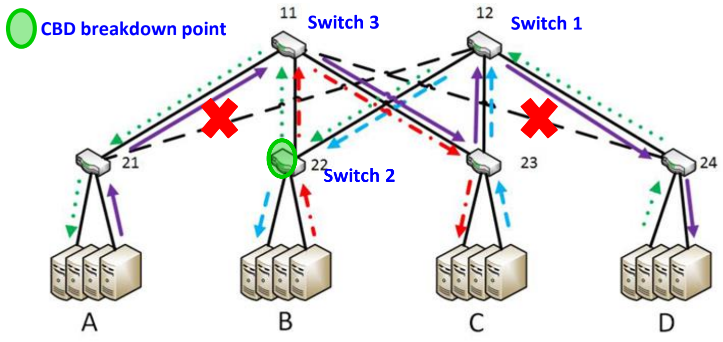


Figure X – Deadlock free algorithm identifies CBD points based on down-up flows

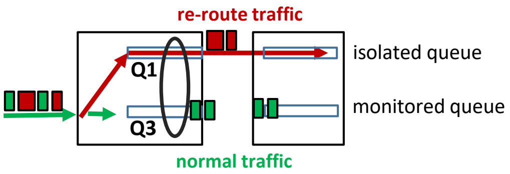
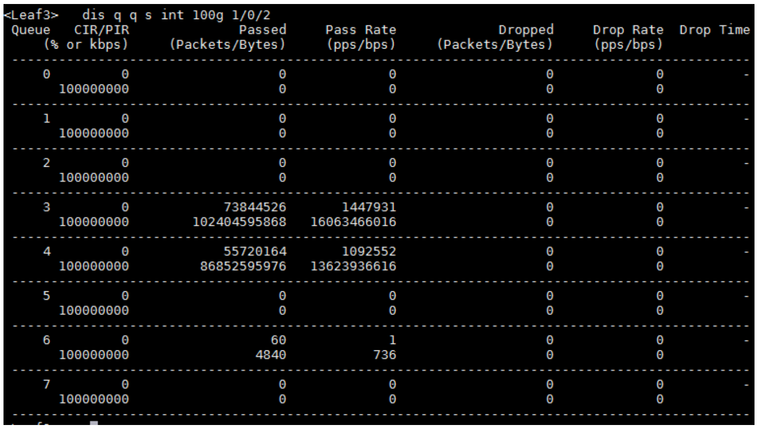


Figure X – Queue Switch according to CBD reroute flow recognition

After recognizing the CBD point, the forwarding plane is responsible for breaking the CBD. According to our description in the preceding chapter, when a traffic loop occurs and a CBD is formed, flows from each switches that in this loop are all in the same queue. Therefore, to break the CBD point, packets need to be switched to another queue. Figure X illustrates the process of queue switch. We know that switch 2 is the CBD point so the flow queue need to be switched. Assume there are two flows in the queue, the red one is re-route traffic that is determined by down-up reroute path. The algorithm will switch the red traffic into another queue. The flow that change to another queue will lead the elimination of PFC deadlock. Different flows can pass by through different queues.

To verify the effect of this algorithm, ODCC conduct a test to demonstrate the performance of the deadlock free algorithm. CAICT, Baidu, Meituan, China Telecom, China Mobile, Huawei, Cisco, Mellanox, H3C, Centec, Keysight companies participate in this test. We still use the same test topology in the Table X and enable the switch's deadlock-free feature.

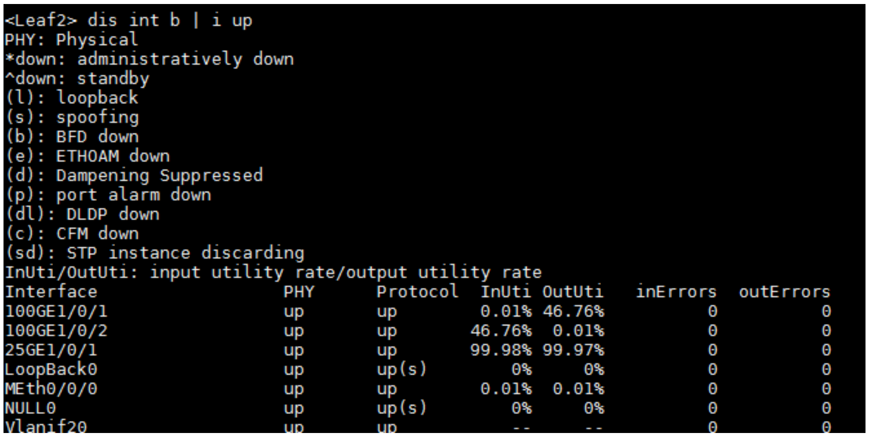
Figure X shows the test result of deadlock free. Without deadlock-free mechanism, the flows that cause CBD loop are all in the same queue (queue 3, shown in figure ddd). This is the main reason why PFC deadlock occurs. After PFC deadlock free feature is configured, the traffic causing deadlock switch from queue 3 to queue 4. This is due to the algorithm we described above, which implements automatic switching of the deadlock traffic queue.



```
<Leaf3>   dis q q s int 100g 1/0/2
 Queue   CIR/PIR              Passed     Pass Rate             Dropped     Drop Rate  Drop Time
     (% or kbps)     (Packets/Bytes)     (pps/bps)     (Packets/Bytes)     (pps/bps)
-----------------------------------------------------------------------------------------------
     0         0                   0             0                   0             0          -
       100000000                   0             0                   0             0
-----------------------------------------------------------------------------------------------
     1         0                   0             0                   0             0          -
       100000000                   0             0                   0             0
-----------------------------------------------------------------------------------------------
     2         0                   0             0                   0             0          -
       100000000                   0             0                   0             0
-----------------------------------------------------------------------------------------------
     3         0            73844526       1447931                   0             0          -
       100000000        102404595868   16063466016                   0             0
-----------------------------------------------------------------------------------------------
     4         0            55720164       1092552                   0             0          -
       100000000         86852595976   13623936616                   0             0
-----------------------------------------------------------------------------------------------
     5         0                   0             0                   0             0          -
       100000000                   0             0                   0             0
-----------------------------------------------------------------------------------------------
     6         0                  60             1                   0             0          -
       100000000                4840           736                   0             0
-----------------------------------------------------------------------------------------------
     7         0                   0             0                   0             0          -
       100000000                   0             0                   0             0
-----------------------------------------------------------------------------------------------
```

Figure X – traffic causing deadlock switch from queue 3 to queue 4 to avoid deadlock

In addition to the queue automatic switch, the test result shows that no deadlock occurs on the network. We obey the test step again according to the test specification in Table X. The test output shows that no PFC deadlock occurs on the network after the PFC deadlock free function is enabled.



```
<Leaf2> dis int b | i up
PHY: Physical
*down: administratively down
^down: standby
(l): loopback
(s): spoofing
(b): BFD down
(e): ETHOAM down
(d): Dampening Suppressed
(p): port alarm down
(dl): DLDP down
(c): CFM down
(sd): STP instance discarding
InUti/OutUti: input utility rate/output utility rate
Interface                   PHY      Protocol  InUti OutUti   inErrors  outErrors
100GE1/0/1                  up       up         0.01% 46.76%         0          0
100GE1/0/2                  up       up        46.76%  0.01%         0          0
25GE1/0/1                   up       up        99.98% 99.97%         0          0
LoopBack0                   up       up(s)         0%     0%         0          0
MEth0/0/0                   up       up         0.01%  0.01%         0          0
NULL0                       up       up(s)         0%     0%         0          0
Vlanif20                    up       up            --     --         0          0
```

Figure X – No Deadlock occurs, and the network traffic transmission is normal

- ✓ Tuning RDMA networks is an important factor to achieving high-performance
- ✓ Current method of parameters configuration can be a complex operation
- ✓ Congestion control algorithms usually requires collaboration between the NIC and switch
- ✓ Traditional PFC manual configuration needs complex calculation with lots of parameters
- ✓ Excessive headroom leads to reduce the number of lossless queues while too little headroom leads to packet loss

Improving Congestion Notification

According to the preceding test in table X, in large-scale data center when the RoCE traffic proportion is small in flow coexistence transmission scenario, the latency increases sharply (from dozens of microseconds to milliseconds) and the bandwidth QoS cannot be guaranteed. In addition, more QP connections of RoCE flows will aggravate the problem.

If these problems occur when traffic coexist, the switch sends a large number of PFC pause packets to the upstream device. As a result, packets are stacked on the outbound interface of the switch, causing a long delay.

After analyzing the causes of the high latency problem, we find that when the network is congested, even if all RoCE packets are marked with ECN, the queue is still overstocked, and the ECN becomes invalid. The root cause is that when network is congested, the sender cannot receive sufficient CNP notification packets. As a result, the sender still sends data, causing queue stacking and latency increasing.

To ensure the performance of the RoCE network, PFC is widely used in data centers to ensure no packet loss and DCQCN is used as the congestion control algorithm. As in Figure X, the DCQCN algorithm consists of three parts: a reaction point (RP), a switch (CP, congestion point), and a notification point (NP).

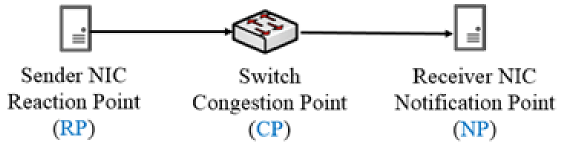


Figure X – Three parts of RoCE congestion control

CP, Congestion Point

On the switch side, if the number of packets in an egress queue exceeds the threshold, the packets are marked with ECN. The ECN marking probability is determined by the queue length. When the queue length is less than Kmin, traffic is not marked. When the queue depth is greater than Kmax, all packets passing through the queue are marked. When the queue depth is between Kmin and Kmax, the marking probability increases according to queue depth.

NP, Notification Point

When a data packet with an ECN flag arrives at a receiver, it indicates that congestion occurs in a network. Receiver needs to send a CNP packet back to the transmit end to transfer congestion information. If a data flow packet with an ECN flag arrives and no CNP has been sent in the past N microseconds, in this case, the receiver immediately sends a CNP packet. With enhancement of a current data center network adapter chip capability, N may be set to 0, that is, a receive end may respond to each packet with an ECN flag, and send a CNP packet.

RP, Reaction Point

When the transmit end receives a CNP, it decreases the current rate and updates the rate decrease factor. If the transmit end does not receive any CNP within a specified period, it increases the transmit rate based on a certain algorithm.

With the increase of the data center network scale, when the flow scale is large, the average bandwidth allocated to each flow is small. For the flow that is congested, the packet interval of each flow (that is, the minimum interval for the flow to obtain CNP packets) may be greater than the rate increase interval. So that the rate of the flow that is still in the congested state is increased, as a result, the speed control convergence fails.

For example, if the port rate of the switch is 25 Gbit/s and the number of RoCE flows is 300, the average rate of each RoCE flow is as follows: 25 Gbit/s (port bandwidth) /300 (number of flows) = 80 Mbit/s. In this case, a 4 KB packet interval (MTU for NIC) is generated every 400 us, if the speed-up interval time of NIC is 300us (default value in commercial NIC), so the time of packet interval is greater than the rate increase/decrease interval, the speed control fails when the switch port is congested.

To solve this problem, ODCC provides a solution to intelligently supplement the rate of the CNP packet based on the congestion level of the port, interval of the received CNP packet, and interval of the DCQCN rate increase on the network side.

This algorithm proposes a method for network congestion awareness and network congestion control.

After receiving CE packets, the CP determines whether a queue is congested based on the frequency of CE packets and determines whether a flow is congested based on the sequence number of CE packets. If a queue is congested and a flow is congested, the CP proactively supplements CNP packets based on the intervals for CE packets and congestion control rate increase. The problem that the sending rate increases when congestion occurs on the switch port is solved. The CNP packet is supplemented to cause the conditions are not met for triggering rate increase. It can prevent unexpected rate increase when the port is congested, thereby solving the problem of the rate control failure. The CNP operation is performed only when the port is severely congested and no CNP is performed for a long time. Therefore, the rate increase and throughput are not affected when the DCQCN is in normal state. The solution is shown in Figure x.

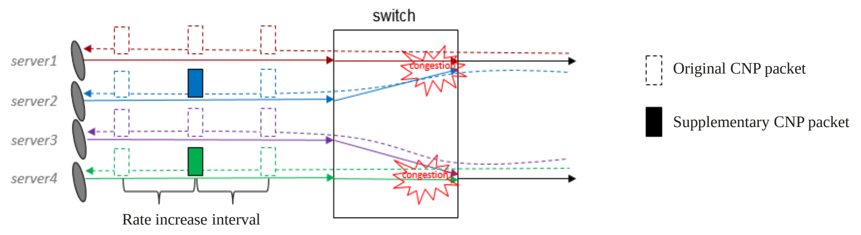


Figure x – Intelligent Complementary CNP

The ODCC organizes the test of the congestion control mechanism algorithm, and the effect is obvious. According to the test result, the bandwidth QoS performance is improved by more than 30% (TCP:RoCE = 9:1 scenario). [ODCC test result]

- ✓ Improved Explicit Congestion Notification
- ✓ Enhanced version of Quantized Congestion Notification (originally IEEE 802.1Qau)
- ✓ Intelligent Methods of improving QoS support in mixed traffic environments
- ✓ Test verification (ODCC lossless DCN test specification and result)

Configuration complexity of congestion control algorithms

Due to the high concurrency feature of distributed application architecture operations, a large number of concurrent data flows exist in data center network, which easily causes network congestion. Network congestion may cause extra delay, resulting in high packet transmission delay, low throughput, and a large amount of resource consumption. How to efficiently control network congestion, obtain higher bandwidth and lower latency, and improve network transmission efficiency is the key to improving data center performance.

As mentioned above, the ECN threshold is set to a low value to achieve low latency. However, a low ECN threshold often leads to low network throughput. Figure X shows the high ECN threshold has better performance for throughput-sensitive large traffic.

In the CC issue section, we also discuss how to minimize latency while maintaining throughput when traffic is mixed. It can be seen that the tune of the RDMA network is a big challenge to achieve the optimal throughput and latency and maximize the performance of the entire network.

The traditional congestion control algorithm commonly used in the industry usually requires network adapter and network collaboration. Each node needs to be configured with dozens of parameters, and the parameter combination of the entire network reaches hundreds of thousands. To simplify the configuration, you can only use the recommended static configuration based on the experience of engineers.

Common static configurations face the following two challenges: Real-time change of network traffic and effects on service performance.

Real-time change of network traffic

Take distributed block storage services as an example. During the running process, the read/write ratio, I/O block size, and number of concurrent read/write tasks always change, and the network traffic mode changes dynamically and continuously. Due to the dynamic traffic changes in the customer's environment, the manually configured static threshold may cause rough back pressure. Static threshold is difficult to adapt to the real-time network traffic changes. As a result, low throughput and high latency may occur, and network performance may deteriorate.

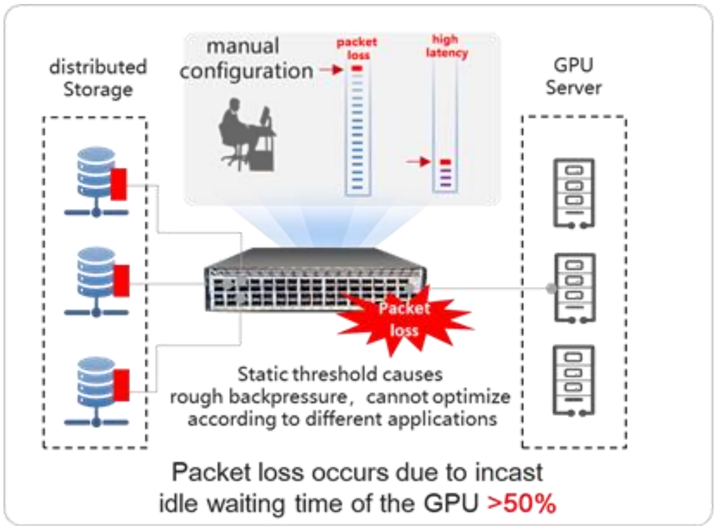

distributed Storage
manual configuration
packet loss
high latency
GPU Server
Packet loss
Static threshold causes rough backpressure， cannot optimize according to different applications
Packet loss occurs due to incast
idle waiting time of the GPU >50%

Figure Gx – Packet loss

Service performance is affected

The congestion control algorithm parameters configured on the entire network determine the effect of congestion control. The performance difference with different parameters can be as high as 50%. Static experience configuration cannot ensure the optimal performance of most service scenarios in the customer environment. For customers, parameter configuration may lead to a gap of more than 50% in service performance or device investment. Selecting appropriate CC algorithm parameters is significant for improving the service performance of customers. Lab tests show that different congestion control algorithms produce different effects in the same application scenario, shown in Figure X.

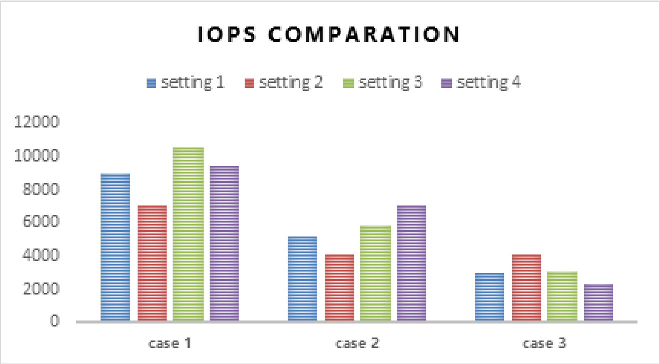


Figure X – IOPS Comparison

Intelligent congestion parameter optimization

Control network congestion to ensure efficient and stable running of DCN services. If incast traffic is sent, traffic bursts occur on the receive end. As a result, a large number of packets are accumulated in the queue, and the number of packets exceeds the capability of the interface on the receive end instantaneously. Consequently packet loss occurs due to network congestion. Based on traditional Ethernet, we use a heuristic algorithm to monitor network traffic bursts and proactively intervene in the network before congestion occurs, ensuring stable and efficient running of DCN services.

The dynamic threshold adapts to network traffic changes, and precise backpressure is used to decrease the rate.

Our algorithm proactively detects network traffic modes and interacts with network environments, greatly improving the adaptability of network congestion algorithms. Uses dynamic threshold, precise backpressure, and proper rate reduction without manual adjustment, reducing O&M costs.

Optimal entire network performance, improving network and application performance.

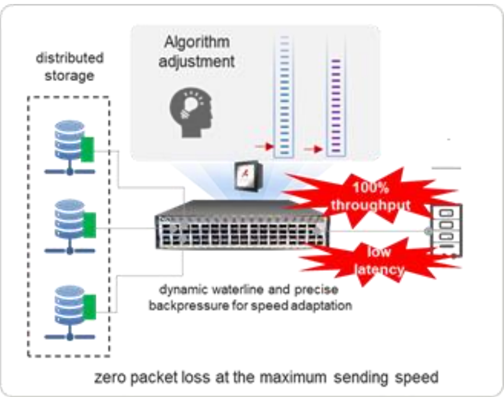

distributed storage
Algorithm adjustment
100% throughput
low latency
dynamic waterline and precise backpressure for speed adaptation
zero packet loss at the maximum sending speed

Figure X – zero packet loss at max speed

Compared with local optimization policies deployed on CPUs, the algorithm detects global network traffic changes, achieving the highest global network performance and ensuring optimal service performance.

ODCC tests the performance of the intelligent congestion parameter adjustment algorithm. The result shows that the new technical solution improves service throughput and latency simultaneously. For OLTP services, the delay decreases by up to 12%. For video services, the throughput increases by up to 25%.

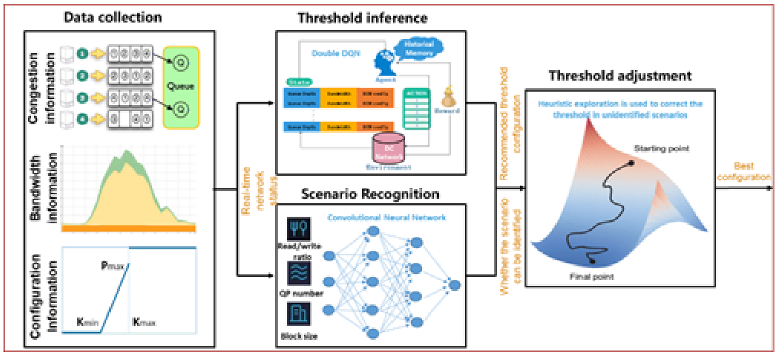


Figure Z – Cool Stuff

Buffer optimization to reduce the complexity of PFC headroom configuration

PFC is a hop-by-hop protocol between two Ethernet nodes. As show in above, the sender's egress port sends data packets to the receiver's ingress port. At the receiving ingress port, packets are buffered in corresponding ingress queues. Once the ingress queue length reaches a certain threshold (XOFF), the switch sends out a PFC pause frame to the corresponding upstream egress queue. After the egress queue receives the pause frame, it stops sending packets. Once the ingress queue length falls below another threshold (XON), the switch sends a pause with zero duration to resume transmission.

RoCE needs PFC mechanism to achieve lossless Ethernet. Network switch enables PFC to make sure that there's no packet loss in network. Each lossless queue needs to be configured with enough headroom buffer [~~19~~23].

Originally, configuring the PFC threshold was a very experiential task. The calculation of PFC threshold is complex with lots of parameters (Buffer structure and unit size, switching delay, cable delay and interface delay) (See Clause 36 of [~~15~~18]). PFC buffer requires both highly usage and implementation dependent.

Pay attention to the fact that different vendors may have different implementations and would imply a different configuration for the headroom. Thus, excessive headroom leads to reduce the number of lossless queues while too little headroom leads to packet loss [~~14~~17].

To solve this problem of PFC headroom configuration complexity, ODCC provides an adaptive headroom calculation algorithm to simplify the configuration complexity. The core idea of this algorithm is using intelligent and dynamic distance discovery method. A round trip timer determines the latency between two connected switches. According to the MTU size, bandwidth and so on, automatically allocates the desired amount of headroom needed to ensure no frame loss due to congestion. The switch will never allocate more headroom than the maximum needed. Thus, we can reserve enough headroom for more lossless queues. The test result shows that the adaptive headroom algorithm can release more buffer space for more lossless queues and improve the latency by 30% to 50% in long-distance transmission scenarios.

- ✓ Intelligent headroom calculation
- ✓ Intelligent heuristic algorithms for identifying congestion parameters
- ✓ Methods for dynamic optimization based on services
- ✓ Test verification (ODCC lossless DCN test specification and result)
- ✓ Self-adaptive headroom configuration

6 Standardization Considerations

Things for the IEEE 802 and IETF to consider. Possibly others as well – SNIA, IBTA, NVMe, etc..

7 Conclusion

Closing words…

8 Citations

<< format the table later – MS word screws up the format each time your rebuild, so just wait until the end and change the column widths to get it to look correct. >>

[1] IEEE, "Nendica Work Item: Data Center Networks," [Online]. Available: https://1.ieee802.org/nendica-DCN/. [Accessed 14 05 2020].

[2] IEEE, "IEEE 802 Nendica Report: The Lossless Network for Data Centers," 17 8 2018. [Online]. Available: https://xploreqa.ieee.org/servlet/opac?punumber=8462817. [Accessed 13 05 2020].

[3] J. Wiles, "Mobilize Every Function in the Organization for Digitalization," Gartner, 03 December 2018. [Online]. Available: https://www.gartner.com/smarterwithgartner/mobilize-every-function-in-the-organization-for-digitalization/. [Accessed 10 June 2020].

[4] Huawei, "Huawei Predicts 10 Megatrends for 2025," Huawei, 08 August 2019. [Online]. Available: https://www.huawei.com/en/press-events/news/2019/8/huawei-predicts-10-megatrends-2025. [Accessed 10 June 2020].

[5] J. Handy and T. Coughlin, "Survey: Users Share Their Storage," 12 2014. [Online]. Available: https://www.snia.org/sites/default/files/SNIA%20IOPS%20Survey%20White%20Paper.pdf. [Accessed 14 05 2020].

[6] Huawei, "AI, This Is the Intelligent and Lossless Data Center Network You Want!," 13 March 2019. [Online]. Available: https://www.cio.com/article/3347337/ai-this-is-the-intelligent-and-lossless-data-center-network-you-want.html. [Accessed 14 05 2020].

[7] E. K. Karuppiah, "Real World Problem Simplification Using Deep Learning / AI," 2 November 2017. [Online]. Available: https://www.fujitsu.com/sg/Images/8.3.2%20FAC2017Track3_EttikanKaruppiah_RealWorldProblemSimplificationUsingDeepLearningAI%20.pdf. [Accessed 14 05 2020].

[8] O. Cardona, "Towards Hyperscale High Performance Computing with RDMA," 12 June 2019. [Online]. Available: https://pc.nanog.org/static/published/meetings/NANOG76/1999/20190612_Cardona_Towards_Hyperscale_High_v1.pdf. [Accessed 14 05 2020].

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Deleted Cells

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted Table

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted ...

Formatted ...

[9] J. L. Jacobi, "NVMe SSDs: Everything you need to know about this insanely fast storage," 10 March 2019. [Online]. Available: https://www.pcworld.com/article/2899351/everything-you-need-to-know-about-nvme.html. [Accessed 14 05 2020].

[10] M. Alipio, N. M. Tiglao, F. Bokhari and S. Khalid, "TCP incast solutions in data center networks: A classification and survey," *Journal of Network and Computer Applications,* vol. 146, p. 102421, 2019.

~~[11] Z. Jai, Y. Kwon, G. Shipman, P. McCormick, M. Erez and A. Aiken, "A distributed multi-GPU system for f processing," in *VLDB Endownment*, 2017.~~

[~~12~~11] T. P. Morgan, "Machine Learning Gets An Infiniband Boost With Caffe2," 19 April 2017. [Online]. Available: https://www.nextplatform.com/2017/04/19/machine-learning-gets-infiniband-boost-caffe2/. [Accessed 14 05 2020].

[12] Z. Jai, Y. Kwon, G. Shipman, P. McCormick, M. Erez and A. Aiken, "A distributed multi-GPU system for fast graph processing," in *VLDB Endownment*, 2017.

[13] Wikipedia, "IEEE 802.3," 5 June 2020. [Online]. Available: https://en.wikipedia.org/wiki/IEEE_802.3. [Accessed 22 July 2020].

[14] K. Rupp, "42 Years of Microprocessor Trend Data," February 2018. [Online]. Available: https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/. [Accessed 22 July 2020].

[15] The Linux Foundation, "Open vSwitch," 2016. [Online]. Available: https://www.openvswitch.org/. [Accessed 23 July 2020].

[~~13~~16] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, F. Kelly, M. Alizadeh and M. Yu, "HPCC: high precision congestion control," in *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM '19)*, New York, NY, USA, 2019.

[~~14~~17] C. Guo, H. Wu, Z. Deng, G. Soni, J. Ye, J. Padhye and M. Lipshteyn, "RDMA over Commodity Ethernet at Scale," in *In Proceedings of the 2016 ACM SIGCOMM Conference (SIGCOMM '16)*, 2016.

[~~15~~18] IEEE, IEEE Std 802.1Q-2018, IEEE Standard for Local and Metropolitan Area Networks — Bridges and Bridged Networks, IEEE, 2018.

[19] M. Karok, J. Golestani and D. Lee, "Prevention of deadlocks and livelocks in lossless backpressured packet networks," *IEEE/ACM Transactions on Networking,* vol. 11, no. 6, p. 11, 2003.

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted Table

Deleted Cells

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted Table

Deleted Cells

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

[~~16~~20] S. Hu, Y. Zhu, P. Cheng, C. Guo, K. Yan, J. Padhye and K. Chen, "Deadlocks in datacenter networks: Why do they form, and how to avoid them," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, 2016.

[~~17~~21] S. Hu, Y. Zhu, P. Cheng, C. Guo, K. Tan, J. Padhye and K. Chen, "Tagger: Practical PFC Deadlock Prevention in Data Center Networks," in *In Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '17)*, 2017.

[~~18~~22] S. Das and R. Sankar, "Broadcom Smart-Buffer Technology in Data Center Switches for Cost-Effective Performance Scaling of Cloud Applications," April 2012. [Online]. Available: https://docs.broadcom.com/docs-and-downloads/collateral/etp/SBT-ETP100.pdf. [Accessed 24 June 2020].

[~~19~~23] Huawei, "Configuration Guide - Low Latency Network," [Online]. Available: https://support.huawei.com/enterprise/en/doc/EDOC1100040243/c28a82e4/buffer-optimization-of-lossless-queues. [Accessed 14 07 2020].

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Deleted Cells

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted Table

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around

Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: In line, Relative to: Margin, Horizontal: 0", Wrap Around