Technical Insight Report

Networking Issues in Scale Out Architectures

By Mitch Lewis

September 23, 2020

Evaluator Group

Enabling you to make the best technology decisions

Executive Summary

As modern datacenters increasingly embrace scale-out IT architectures, a number of problems can arise that impact network performance. Oftentimes, these issues extend past networking to negatively impact the performance of storage systems as well. As datacenters evolve, networks may also be required to evolve with changes in equipment, architecture, and protocols to overcome these problems. This paper seeks to shed light on those problems and the associated impact on connected storage systems.

Ethernet Networking in the Datacenter

Networking within the datacenter has a constant requirement of high bandwidth and low latency. There are a number of technologies that can be deployed to meet these needs such as Fibre Channel or Infiniband, but increasingly Ethernet is being used. Ethernet is a well-known technology that leverages TCP/IP and can be deployed with commodity switches, often at a lower cost than its competitors. Additionally, Ethernet has been able to provide increasing speeds, with 10GbE, 25GbE, 50GbE, and 100GbE available, without requiring intensive infrastructure changes that may be required by adopting new technologies.

While certainly not the only option, Ethernet has seen, and will continue to see, widespread use for datacenter networks. As the technologies, architectures, and strategies deployed within the datacenter evolve to address new challenges, the underlying Ethernet network must also find ways to adapt.

A Change in Traffic

Traditionally, a datacenter network's core focus has been to transfer data in and out of the datacenter quickly and efficiently. This traffic pattern of data moving to and from the datacenter is known as North-South traffic and for a long time has comprised the majority of a network's traffic. The other type of network traffic, known as East-West traffic, involves data movement within the datacenter. This East-West traffic flow comes from servers and storage systems communicating amongst themselves – a traffic pattern that traditionally takes a backseat to serving the client facing North-South traffic.

Modern datacenters, however, have adopted new technologies for both compute and storage resources that have dramatically changed the traditional network traffic patterns. Servers are now commonly virtualized with virtual machines migrating amongst physical hosts. Newer strategies are utilizing container technologies with applications deployed as microservices. Additionally, hyperconverged infrastructure, which combines storage, compute, and networking, is now commonly used and storage systems have scaled-out with distributed data, and automated replication and mirroring between systems. On top of this there are often intelligent monitoring and resource management solutions. All of

these developments have scaled-out the datacenter and contribute to an increase in East-West network communication.

The traditional thinking, that North-South traffic comprises the majority of network traffic and is therefore the most important, is no longer true. Datacenters have become so scaled out that the majority of network traffic now comes from within the datacenter. This increase in East-West traffic signifies a fundamental shift in a networks purpose within a datacenter, and likewise requires a fundamental shift in design.

The Three Tier Architecture

The traditional network architecture that has been commonly used in datacenters is known as the three tier architecture, aptly named for having three tiers of switches. The three tier architecture consists of a core layer, a distribution layer, and an access layer.

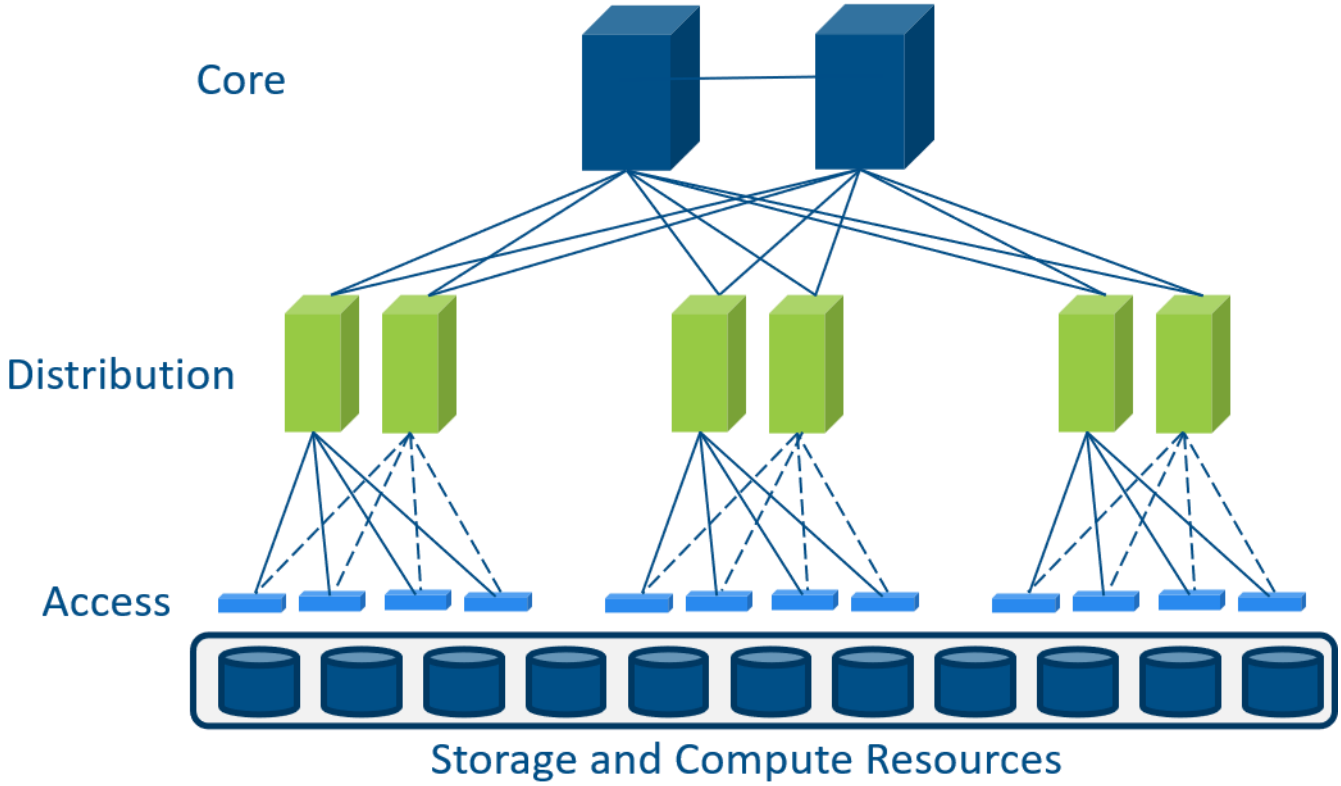

Figure 1: Three Tier Architecture

At the bottom of the architecture is the access layer, which consists of switches that are responsible for connecting to end devices such as servers and storage systems. The middle layer, known as the

distribution layer, connects the switches at the access layer. At the top of the architecture is the core layer, which is responsible for connecting the distribution layer, as well as routing data to networks outside of the datacenter. The core layer acts as the backbone of the network and typically uses the most high-performance, and therefore most expensive, routers in order to distribute data to the rest of the network as quickly as possible.

The three tier architecture uses the Spanning Tree Protocol to prevent loops from forming in the network. Spanning Tree Protocol prevents these loops by deterministically blocking redundant paths in the network.

While the three tier architecture has been successful for many years, the design was created with North-South traffic in mind, as this was traditionally the predominant traffic flow. When dealing with high volumes of East-West traffic, however, the three tier architecture becomes a less efficient solution. While the architecture itself may provide redundant paths, the Spanning Tree Protocol's blocking solution restricts the network to using a single path between any two switches at a given time. This can lead to paths becoming oversubscribed.

Additionally, the three tier architecture creates inefficiencies when data traverses the network in an East-West traffic pattern where data travels between switches in the access layer. The problem with this architecture is not only that the travel of East-West traffic may be inefficient, it is that the latency is unpredictable. Given the example of the three tier architecture shown below in Figure 2, data travelling between two access switches that are directly linked to the same switch in the distribution tier, the transaction may require 2 hops – one to the distribution tier, and one to its destination in the access tier.

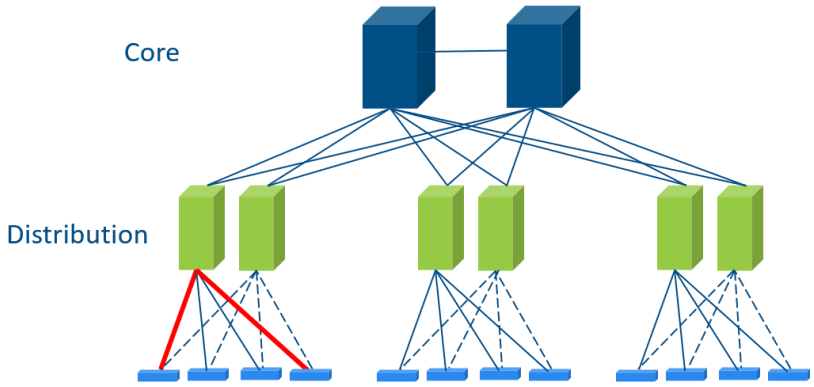

Figure 2: Two Hop Data Path in Three Tier Architecture

Within the same architecture, however, two switches on opposite ends of the access tier may take up to 4 hops to connect as the network must be traversed from the access tier, to the distribution tier, up to the core tier, back to a different area of the distribution tier, and then finally to its target on the access

tier. This variance in East-West traffic makes the latency unpredictable and inhibits the three tier architecture from effectively scaling outwards with the datacenter.

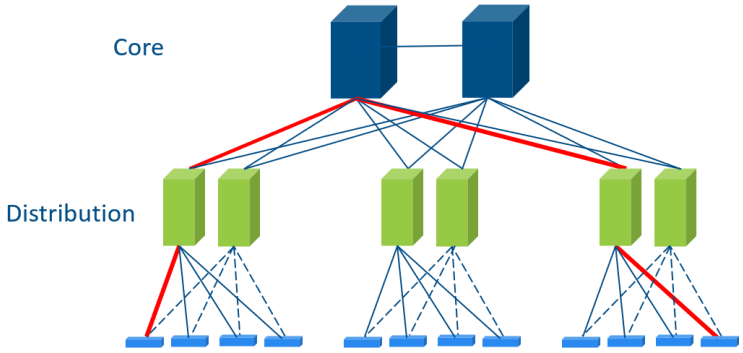

Figure 3: Four Hop Data Path in Three Tier Architecture

Spine and Leaf

In response to the three tier architecture's unsustainable handling of heavy East-West traffic patterns, a new network architecture has emerged to accommodate the traffic patterns found in most modern datacenters. Known as the spine and leaf architecture, this design collapses the core and distribution layers found in a three tier architecture into a single spine layer, that connects to the access, or "leaf" layer. In this architecture, every leaf is connected to every spine.

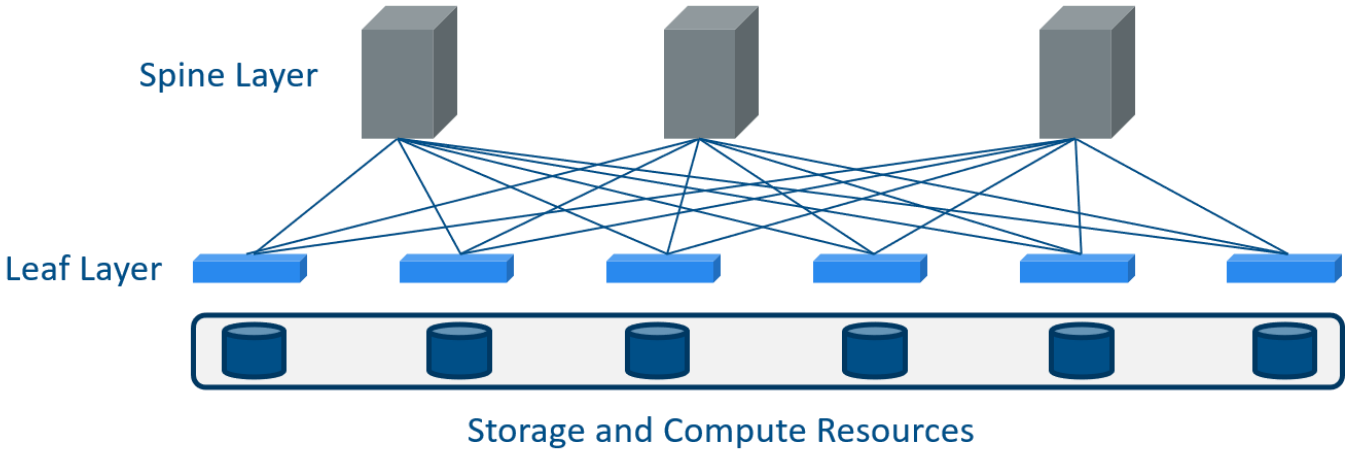

Figure 4: Spine and Leaf Network Architecture

Along with the change to a two-tiered approach, the spine and leaf network leaves behind the spanning tree protocol used in three tier architectures. Instead, spine and leaf architectures utilize Equal Cost

Mulitipath (ECMP) routing. This can be accomplished entirely at layer 2 via protocols such as Transparent Interconnection of Lots of Links (TRILL) or Shortest Path Bridging (SPB), or perhaps more effectively using layer 3 switches with protocols such as Border Gateway Protocol (BGP) or Open Shortest Path First (OSPF). ECMP allows for redundant connections to be utilized for transferring packets, rather than being blocked such as in Spanning Tree Protocol, while still preventing loops. This utilization of additional pathways increases network bandwidth and eliminates the cost associated with unused infrastructure.

Along with providing greater bandwidth through non-blocking protocols, spine and leaf topologies provide greater scalability and predictable latency. Unlike in the three tier model, in which the number of hops between access switches varies, the spine and leaf model can guarantee two hops between any two leaf switches due to the fact that every leaf is connected to every spine. This provides consistency for East-West traffic flows, even as the datacenter and network continue to scale outward.

As datacenter requirements increase, a spine and leaf network can scale to meet its needs. In a three tier architecture, scaling horizontally is inefficient. Adding core devices requires purchasing expensive equipment, while adding to the access or distribution tiers only provides a limited increase in bandwidth as added connections are blocked by Spanning Tree. With spine and leaf, scaling is much more efficient. Either spine or leaf switches can be added incrementally. As more end device connections are required, a leaf switch can be added. Likewise, as greater bandwidth or resiliency is required, additional spine switches can be added without wasting redundant connections.

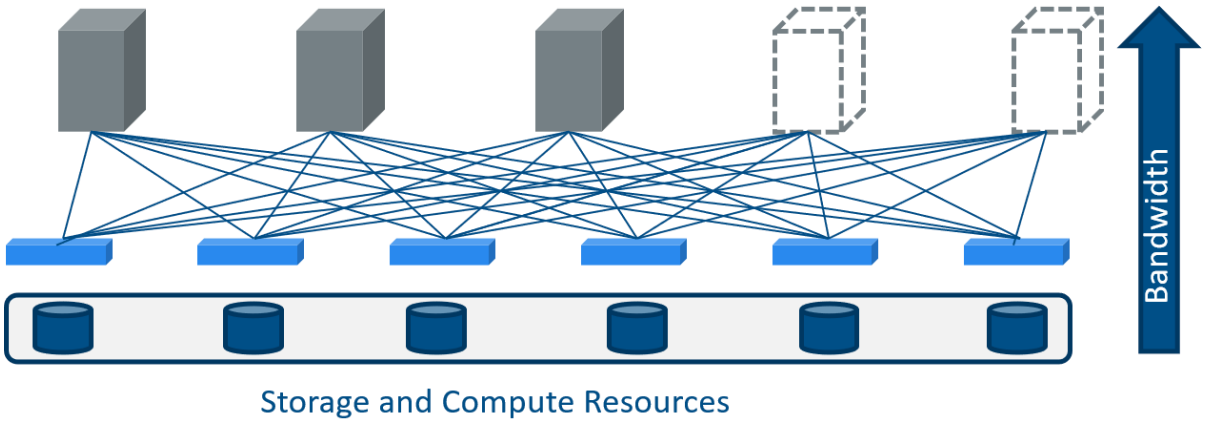

Figure 5: Scaling with Additional Spine Switches

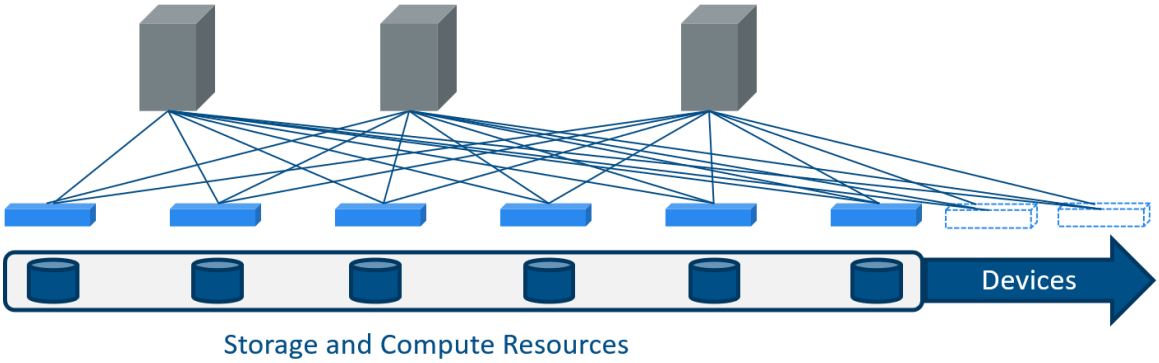

Figure 6: Scaling with Additional Leaf Switches

As datacenter's compute and storage needs continue to scale outward, the spine and leaf architecture provides a networking topology that can scale accordingly while providing high bandwidth and predictable latency.

The Incast Problem

While a change in network topology is crucial to addressing the growing East-West traffic found in modern datacenters, it alone cannot solve all of the problems associated with the rapid increase in intra-datacenter network traffic. One problem found in Ethernet networks that can be caused by East-West traffic patterns is known as TCP incast.

TCP incast, also sometimes referred to as microbursts, is a throughput collapse caused by many-to-one traffic patterns utilizing TCP. Although often an infrequent occurrence, it is possible for a burst of multiple packet streams to arrive at a switch simultaneously, resulting in an overflow of the switch's buffer and a collective pullback of all TCP sessions. The frequency of this phenomena increases as networks scale out and traffic between devices within the datacenter increases. In addition, the adoption of high bandwidth, low latency networks, and use of NVMe/TCP, which is capable of opening up thousands of parallel TCP connections, can lead to an increase of incast events.

Dropping packets is a normal occurrence in TCP sessions, and does not necessarily signify a disastrous situation on its own. When a packet is sent, a timer begins until the sender receives acknowledgement that it was received. If the timer expires before the acknowledgement, the packet is assumed to have been lost and is retransmitted. This waiting time before retransmission is known as a Retransmission Timeout (RTO). This timeout is used to avoid unnecessarily resending packets and congesting the network. If a packet needs multiple retransmissions, the RTO grows exponentially longer to give the network time to adjust.

This procedure works well in many instances, however, it proves troublesome with bursts of simultaneous traffic patterns, such as those found to cause TCP incast. In this scenario, multiple synchronous TCP sessions reach a switch simultaneously resulting in a buffer overflow and packet loss, potentially across multiple sessions. While seemingly unaffected sessions can continue to transmit packets, the client is left waiting for the duration of the timeout before it can receive the remainder of the missing packets. This leaves the client's link idle and results in a severe underutilization in bandwidth. The issue becomes increasingly severe as packets continue to be dropped and RTO grows exponential.

Incast scenarios are normally not frequent occurrences, but large amounts of network traffic flowing in East-West traffic patterns throughout datacenters increases the probability, and the effect can be devastating to a network's performance. A common cause is scale out storage systems in which a single request requires a response consisting of data striped over multiple storage nodes.

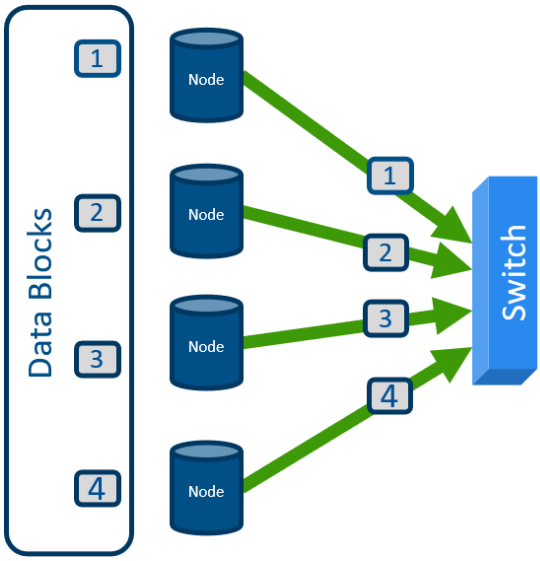

Figure 7: Synchronous Data Transfer from Scale Out Storage

This collapse in network throughput is not only detrimental to the performance of the network itself, but severely impacts the performance of all attached servers and storage systems as well. A number of strategies have been proposed to handle or avoid these incast situations, but due to the varying nature of network traffic across different datacenters, it becomes difficult to identify a singular solution that fits all needs.

Big Buffers

Perhaps the most immediate response to an issue regarding buffer overflow, and one that has been proposed by many switch vendors, is to increase the size of the buffer. A switch with a larger buffer can potentially accommodate the large burst in traffic found in incast situations. This stops packets from being dropped and ultimately prevents the resulting throughput collapse.

This solution may be adequate in some cases; however, it is important to note that increasing buffer sizes brings along its own issues. The first, and not insignificant, hurdle for organizations to overcome is the increased cost of switches with large buffers. A larger buffer means increased memory, and in turn, increased cost. For many organizations the cost of these large buffer switches may not be justifiable.

An additional concern with the use of big buffer switches, is increased latency. While a large buffer has more room to store data to protect against microbursts, the more data being stored at the switch increases the latency of the network. With large buffers, a few large TCP flows can clog the buffer, resulting in a delay for many short lived TCP flows that become stuck in the buffer. Large buffers may have enough room to accommodate packet congestion without incurring loss, but this can cause issues with TCP congestion control. Without dropping packets, there is no indication of congestion and the TCP congestion window will continue to grow, despite any congestion that is occurring at the switch. Ultimately this can lead to increased latency and unnecessary packet drops, even without an incast scenario.

DCTCP

Another approach to addressing the incast problem is to make adjustments to the TCP protocol. One such variation of TCP is known as Datacenter TCP, or DCTCP. Regular TCP can be modified to DCTCP with only a minimal code change. While TCP's normal congestion control uses an implicit detection of congestion — identifying when a packet is dropped — the DCTCP variation uses an explicit congestion detection.

DCTCP leverages Explicit Congestion Notification (ECN) to detect network congestion without dropping packets. ECN is not unique to the DCTCP variation of TCP, and in fact is a feature commonly supported by most commodity switches, although typically disabled by default. When using ECN, IP headers utilize two bits to signify congestion. The first bit, the ECN Capable Transport bit (ECT), signifies whether or not a switch has ECN configured and the second bit, the Congestion Experienced bit (CE) signifies that congestion has been detected. This information is echoed back to the sender, and if the CE bit has identified congestion in the network path, action is taken by implementing TCP's normal congestion control scheme — cutting the congestion window in half. This approach allows a network to respond to congestion before packet loss, reducing the need to re-transmit packets, but it does not assist in the loss of network bandwidth that is a result of decreasing the congestion window by a factor of two.

DCTCP expands on the use of ECN to react to the fraction of packets that have been marked for congestion. With a small number of packets marked, DCTCP will only reduce the congestion window by a small amount. Conversely, if all packets are being marked for congestion, DCTCP will react the same as TCP in reducing the congestion window by half. This method allows the reaction to congestion to be proportional to the congestion that is occurring, optimizing latency while using commodity switches with small or medium sized buffers.

The impact that DCTCP has on network congestion and TCP incast is quite different than that of applying additional buffer capacity. In these instances, DCTCP intervenes to reduce the queue length experienced by TCP flows. This helps prevent small flows from encountering extended latency as the buffer is occupied by larger flows, and from carelessly growing the congestion window during times of congestion, as can occur with large buffers. Additionally, the congestion management provided by DCTCP provides additional headroom in the buffer to absorb bursty traffic patterns, such as those found in incast situations.

The explicit congestion notification and proportionate congestion control used by DCTCP, however, is not enough to completely stop incast scenarios entirely. If a switch's buffer is immediately overwhelmed by the first set of packets transmitted by synchronous TCP sessions, DCTCP will not be able to assist, as the buffer will overflow and packets will be dropped before DCTCP can react. In a more likely scenario, however, DCTCP will provide assistance in a number of ways. First, the use of DCTCP will have provided congestion control to maintain a manageable queue in the buffer before the onset of an incast scenario allowing adequate headspace for handling bursty traffic, as previously discussed. Second, assuming the worst case of initial buffer overflow is avoided, DCTCP can take action to adjust the congestion window as the simultaneous sessions continue to transmit packets, avoiding packet loss and the subsequent waste of bandwidth.

Gentle Slow Start

Much of the focus in changing TCP to avoid TCP incast is concentrated on adjusting the congestion avoidance phase, such as is the case in DCTCP. The incast problem, however, is not bound specifically to packet transmission during congestion avoidance, and additional alterations to TCP may be needed.

One such approach, known as Gentle Slow Start, involves modifying the slow start period that precedes TCP's congestion avoidance phase. Gentle Slow Start can be used with traditional TCP, as well as variants such as DCTCP to further prevent throughput collapse. The Slow Start phase increases the congestion window over time, until the congestion avoidance phase begins, by doubling the congestion window with each Round Trip Time (RTT). In some cases, this approach may increase the congestion window too aggressively and lead to packet loss and incast scenarios before the session even reaches the congestion avoidance phase. This is especially true for short-lived TCP flows that may complete entirely within the Slow Start phase — never receiving assistance in the congestion avoidance phase.

Gentle Slow Start adjusts the rate at which the congestion window is increased by using the RTT to approximate congestion. This allows the congestion window to be increased less aggressively if needed to prevent further congestion of the network. The ability for the Slow Start phase to increase less aggressively helps avoid the buffer overflow, packet loss, and resulting throughput collapse of TCP incast, especially in short-lived TCP flows.

Reducing RTO

An additional response to the TCP incast problem is to reduce the minimum Retransmission Timeout (RTO). The default timeout before a packet can be re-sent is 200ms, but this value is orders of magnitude larger than the average RTT of a packet. This leads to excessive time waiting before retransmitting a packet, and as a consequence, an underutilization of bandwidth and loss of throughput.

To aid in the avoidance of throughput collapse due to TCP incast, the minimum RTO value can be reduced. The value may need to be reduced to 1 ms or even microseconds to provide a short enough RTO — one that is closer to the RTT — to avoid the excessive waiting and throughput loss. While an RTO that is close to the average RTT is capable of dramatically reducing the impact of incast scenarios, the operating system clocks used for RTO are often not granular enough to calculate microseconds, and require modifications to use high resolution timers.

Evaluator Group Comment:
TCP's minimum timeouts, on the order of milliseconds, can cause drastic performance issues which may be unnaccetable in many use cases, including storage. A modified RTO that is close to the RTT will idealy reduce the impact of timeouts, however, the variability of round trip times and difficulty in measuring fine grained timeouts make approximating the RTO to the RTT a challenging task. This may lead to possible spurious retransmissions and further TCP performance issues. The variability involved may make this an undesirable solution.

Beyond Networking

While the issues, and potential solutions, discussed in this paper describe network challenges, they are both caused by, and affect, the connected devices, such as storage systems. Datacenter storage strategies increasingly include solutions that increase East-West traffic throughout the datacenter.

Storage systems replicating and striping data across clustered and scale out implementations can put an increased strain on the datacenter's internal traffic, which as a result may impact the performance of the storage systems. In these instances, it is crucial to understand networking limitations that may affect storage solutions. The networks effect on storage performance can be decreased by proper planning and design, such as using the spine and leaf topology that can scale as needed while providing predictable latency.

Incast issues become more frequent in instances such as scale-out file systems, in which data blocks from multiple storage nodes are sent to fulfill a single request, leading to a many-to-one traffic pattern and risking the associated throughput collapse. While the incast issue occurs within the network, it can cause a severe performance impact for the connected storage systems. This relationship further shows that networking concerns cannot be ignored when implementing and scaling storage systems, even beyond the adoption of scalable topologies such as spine and leaf.

Conclusion

As datacenters continue to scale, the associated networks must be able to scale as well to accommodate new traffic patterns and potential congestion issues. The modern datacenter is much different than datacenters of the past, with significant East-West traffic, virtualization, and scale-out architectures. These datacenters require a network infrastructure that can scale horizontally while providing high bandwidth and low, predictable latency. With Ethernet and TCP/IP widely implemented, datacenters must also overcome the challenges that have been found to occur in Ethernet networks, such as incast.

Many datacenters have, or will soon, transition from older architectures, such as three tier, to newer, more flexible solutions such as spine and leaf to assist in handling the increase in East-West traffic flows. A variety of new approaches, including big buffer switches and a variety of modifications to TCP can be implemented to accommodate the many-to-one traffic patterns that may occur in these East-West flows.

The issues outlined in this paper demonstrate that networking is a crucial component to the effectiveness of a datacenter. Without proper scaling and handling of network issues, the performance and scalability of associated devices, such as storage, can be impacted severely. To avoid impact on the entirety of the datacenter, there are a number of technologies, architectures, and protocols that can be implemented to handle these challenges and improve the scalability of datacenter networks.

Enterprise IT organizations must be aware of the challenges that Ethernet networks may face when scaling out storage and compute resources. The network effects may be performance impacting, or require modifications in order to scale appropriately. As IT trends are adopted, organizations must keep in mind the potential impact to the performance and scalability of the network. Increased usage of

virtualization, software defined storage, hyperconverged systems, and scale out storage are a few examples of situations in which IT decision makers must be aware of the potential impact of an Ethernet network. The degradation of network performance, whether due to an inability to adequately scale or from technical challenges such as TCP incast, will be experienced throughout the entire datacenter and should not be ignored.

About Evaluator Group

Evaluator Group Inc., an Information management and data storage analyst firm, has been covering systems for over 20 years. Executives and IT Managers rely upon us to help make informed decisions to architect and purchase systems supporting their data management objectives. We surpass the current technology landscape by defining requirements and providing an in-depth knowledge of the products as well as the intricacies that dictate long-term successful strategies.