intequus®

EXPANDING THROUGHPUT AT THE EDGE

As more applications and use cases draw on server resources at the edge, it's increasingly important to ensure those servers are operating at maximum throughput and performance efficiency. Doing so requires a host of considerations including the server hardware and its software. This white paper explores what you need to address when optimizing edge servers to ensure they meet the needs of your application users.

Improving Edge Server Throughput and Performance

The use of the edge to deliver latency-sensitive use cases is growing. IoT. Streaming Video. Cloud gaming. All of them require servers as close to the end user to reduce round-trip time and mitigate latency. But how do you ensure that those servers are operating as efficiently as possible? What if, by optimizing the server itself to maximize throughput and performance, you could reduce your physical server needs by 50% and improve end-user satisfaction?

Most businesses would jump at the chance but don't know how to optimize their servers. The solution? Engaging with a cloud partner who can understand your use cases and has a willingness to build balanced, optimized servers for your unique needs.

Why We Need a Better Performing Edge

According to Nielsen's Law, a user's connection speed grows 50% each The cloud has become an intrinsic part of modern business. Public cloud, private cloud, hybrid cloud—businesses around the world are employing containerized applications in distributed, elastic networks which scale to meet user demands. While there are plenty of challenges associated with deploying software into cloud resources, the benefits far outweigh the technical concerns. But like any technology, the cloud has evolved. It's no longer just about distributed infrastructure in lieu of physical servers; it's about where the cloud extends and where the application exists within the cloud.

Applications that move closer to the edge and the end user have different requirements than those that might exist further upstream in the network. These "edge cloud applications" are highly sensitive to response time and latency.

SAVING MILLISECONDS MIGHT SAVE LIVES

Consider an Internet-of-Things (IoT) application handling data from driverless car software. Placing the application at the edge significantly reduces the data's round-trip-time, ensuring that processing the data (and responding back to the car) happens in near real-time. In this example, a few extra milliseconds of latency could mean the difference between life and death.

There are other use cases for delivering from the edge besides IoT. These include streaming video (especially with dynamic ad insertion), video games, and online gambling. In the case of streaming video, every 100 miles of distance between the requesting user and the server can add a millisecond of latency to the video stream.[1]

But putting applications into the edge offers more than just reduced latency. Distributing applications across a large network surface area reduces the potential for DDoS and other cyberattacks by spreading the attack across far more servers. This prevents a single instance from tipping over and ensures user requests can still be met.

Streaming Video

Over the past five years, demand for streaming video has grown significantly. According to Nielsen, "people watched 165 billion minutes of streaming content during the week of March 16, up 36% from 115 billion the week of Feb. 24 and more than double the same week the previous year."[2]

Why does that make the edge important? Because the architecture for delivering streaming video depends on getting content as close to the user as possible to limit the latency in round-trip times. That means serving video from the cloud's edge. Those edge servers, called caches, store popular content so that a user's request doesn't have to travel through the operator network, and even through the internet, to get it.

Because of that architecture, most of the traffic load falls onto the edge. This situation is exacerbated by the nature of streaming traffic. The mass adoption of streaming into segmented HTTP can result in requests for millions of small objects, which puts a lot of pressure on those edge servers. Finally, streaming video is usually delivered from the edge over unicast, meaning that each viewer requires their own stream, further straining throughput and capacity.

Gaming

The advent of multi-player online gaming radically changed the gaming landscape. These games range from Massively Multi-player Online Role-Playing Games (MMORPG) such as World of Warcraft, team-based real-time strategy titles such as League of Legends, or first-person melees such as Call of Duty. In every case latency between the gamer and the server, even just a few milliseconds, can mean the difference between winning and losing.

As a result, the servers that host these multi-player games are located at the edge of the network to mitigate round-trip time, just like streaming video caches. But gaming has different requirements as well. Not only do these servers have to facilitate the connection between tens of thousands of players, but in some cases, their computational power may be needed to render parts of the game such as with Google Stadia and other cloud-based game streaming platforms. In this model, the bulk of the game's heavy-lifting is done in the cloud rather than on a user's gaming console or computer, creating even more demand for the highest levels of edge throughput.

Capacity is Growing, And So is Demand

CONNECTED DEVICES WILL OUT-NUMBER PEOPLE 4 TO 1

According to the Telecommunications Industry Association, by 2022 there will be 29 billion connected devices all vying for attention on the global network.[3] More than half of those, 18 billion, will be IoT devices.

The popularity of streaming video and cloud gaming are a direct result of the increase in network connection speed which, as per Nielsen's Law, grows 50% each year.[4] More available bandwidth means richer application and content experiences for consumers. As users have been able to consume higher-bandwidth digital experiences, businesses have rushed to meet them.

Consider that between December 2010 and March 2020, the median webpage size grew from 481KB to 2080KB, an almost 400% increase.[5] And the amount of data transferred between server and client is only half the story. The other half is the number of clients. Remember that there will be 29 billion connected devices by 2022. The combination of richer internet experiences, such as streaming video, and more devices requiring very low-latency connection to cloud-based software is increasing the pressure on edge capacity.[6]

Network operators and service providers (this includes commercial Content Delivery Networks) have been working to provide the capacity for both those richer experiences and proliferating devices. According to Cisco's Visual Networking Index report, Content Delivery Networks (CDNs) are expected to carry 72% of global internet traffic by 2022.[7]

In fact, global CDN capacity has increased tremendously. Consider the following anecdotal analysis: in May of 2018, Hotstar reported 10 million concurrent viewers for the online stream of the final match of the India Premier League; fast forward just a year, and Akamai reported that they served the 2019 Cricket World Cup to 25.3 million concurrent viewers.[8][9] That's almost double the number of viewers in one year.

The Many Causes of Latency

In many edge-based use cases, such as streaming video and gaming, just a few seconds of latency can destroy the end-user experience. But latency can come from a variety of places within the delivery chain. The table below covers a few of the more common causes and how they might be mitigated.

Type of Latency	Description	How often does it occur?	How can it be mitigated?
Bandwidth is exceeded	There is too much traffic for the provisioned lines	Not often	Good capacity planning (with burst thresholds) can often prevent this
ISP network congestion	The ISP's network is congested with various kinds of traffic	Not often	Better traffic management and prioritization
Problems with end-user equipment	The user's equipment (i.e. gaming console, laptop, connected device) is causing issues with rendering content received from the edge	Fairly often	For computers, users need to be vigilant about ensuring their system is operating at peak performance and has appropriate CPU, GPU, and memory for the content they want to render
Inefficient server configurations	Edge-servers are not configured in an optimal way to ensure the highest available throughput at consistent levels	Not often	Network operators are always looking at ways to improve how their edge servers perform. This can include modification of software to control NICs or even optimization of the TCP stack
Poorly-performing edge servers	Edge server throughput is hampered by the way it is balanced	More often than you would think	Server build-of-materials (BOMs) and configurations are constantly being adjusted to find better, more efficient hardware

Network Operators Are Trying to Keep Up

Increasing capacity at the edge isn't just a matter of throwing more servers at the problem. Many operators understand that the more servers in their network, the more power and physical rackspace they need.

Increasing throughput and capacity at the edge can't be done by brute force. Service providers, like commercial CDN Limelight Networks, are employing other means such as server and software optimization. The global CDN recently announced that its efforts to improve server efficiency had increased the egress amount of its edge by almost 80%.[10]

The big network operators, from ISPs to CDNs, are trying whatever they can to squeeze more throughput and performance out of edge machines without adding more servers. That includes optimizing the TCP stack, improving NIC performance, and utilizing drives that support high I/O. But, there might be a limit to the improvement gleaned from such optimizations. Despite all the tweaks made to server software, the server's fundamental throughput and performance capabilities could be limited by a core flaw: imbalance.

Understanding Server Imbalance

A server is composed of several components, the combination of which affect how well the server performs. When a server is built without considering the optimization of these components towards a specific use case, the result can be a server operating well under peak throughput and performance.

Memory

Configuring a server with balanced memory is important for maximizing its memory bandwidth and overall performance. Depending upon the motherboard configuration, a server will usually have multiple memory channels per processor and a certain number of DIMMs per channel. This makes it important to understand what constitutes a balanced configuration and what doesn't.

According to Lenovo, "When properly configured, the memory subsystem can deliver extremely high memory bandwidth and low memory access latency. When the memory subsystem is incorrectly configured, however, the memory bandwidth available to the server can become limited and overall server performance can be reduced."[11]

Disk

Optimizing hard disk performance is largely about the selection of the disk itself.

For example, a 2.5-inch enterprise-class disk can service a much larger number of random requests per second compared to equivalent 3.5-inch drives. Meanwhile SSD and high-speed flash disks are more appropriate for read-mostly applications with high I/O rates or latency-sensitive I/O (like boot disks).

And if the use case requires heavy, simultaneous I/O, NVMe SSDs are a great choice due to their greater command queue depths, more efficient interrupt processing, and greater efficiency for 4KB commands.[12]

Bus

The internal bus is probably the most important component of the operating system because it provides communication between components such as CPU, memory, disk, and NIC. But it's important to note that a faster bus does not guarantee faster performance.

For use cases that require a lot of throughput (equating to heavy I/O), it's better to have multiple buses for improved performance and throughput. Regardless, the bus is often the main bottleneck in a system and when selecting a bus type (PCI-X versus PCIeXpress), consider the memory requirements for the bandwidth of the channel. Keep in mind too, that improperly compatible hardware, such as connecting fast storage to a slower HBA/NIC, can reduce overall server performance.

CPU

Selecting the right CPU is a critical part of server performance optimization and balancing, especially in multi-CPU systems. That's because specific CPUs, whether Intel Xeon Platinum or AMD, can impact the balance of the server's CPU operation.

For example, when the objective is to deploy containerized instances on a server (say to provide segregated CDN edge node functionality), CPU selection is critical. Since many network operators and service providers have moved to a containerized architecture as part of a microservices approach, some of the services within the containers may be CPU intensive. These services include data analytics or in the case of video streaming, transcoding. Other services, like a reverse proxy, may be network I/O intensive.

But, according to Yuchao Zang, deploying these services onto the same box, "may cause heavily imbalanced resource utilization of servers which could affect the system availability, response time, and throughput."[13] This imbalance is not necessarily a result of the software within the containers. Instead it arises from the way the containers, assigned to specific cores within the processor, interact with the CPU and other server resources like memory, disk, and bus.

THE IMPACT OF CPU SELECTION ON SERVER THROUGHPUT WITH CONTAINERIZED SERVICES

In a simulated test, a specific CPU optimized for container selection at the hardware level (In this case, Intel® Xeon Platinum processors), can have a significant impact on an edge cache node throughput. In this case, the optimized CPU eliminated resource constraints that would lead to imbalanced I/O across the server:

- Network TX Throughput: ~194-196Gbps
- CPU Usage: 77%, 37 cores (Xeon 8276)
- Latency (5MB xfers): 100ms (avg), 523ms (p99)
- Memory Usage ~50GB
- Storage >2GB/s per NVMe

The simulation employed the following components and architecture:

Edge cache node

- 2x Xeon 8276 24C 2.9GHz
- 192GB DDR4
- 2x M.2 SATA RAID1 (OS)
- 10x P4510 4 or 8TB
- 2x Mellanox CX5 100GbE (x16 PCIe)
- CentOS Linux release 7.6.1810 (Core)
- NGINX software (as the reverse proxy)

Origin server (for initial cache fill and cache miss)

- 2x Xeon Gold 6252 24C
- 192GB DDR4
- 1x S4610 960GB (OS)
- 2x Mellanox 100GbE (x16 PCIe).
- NGINX and web_ats_gen[14]

Test generator (x2) to simulate HTTP requests:

- Single Xeon Platinum 82XX 28C
- 192GB DDR4
- 1x S4610 960GB (OS)
- 1x Mellanox 100GbE
- Configured with wrk[15]

The wrk configurations on the test generators were employed for two tasks. The first was to make requests on the Edge Cache Node to fill it from origin, much as would happen in a CDN for new content. The second was to make HTTP requests for that content, which would be in the Edge Cache Node, similar to user traffic.

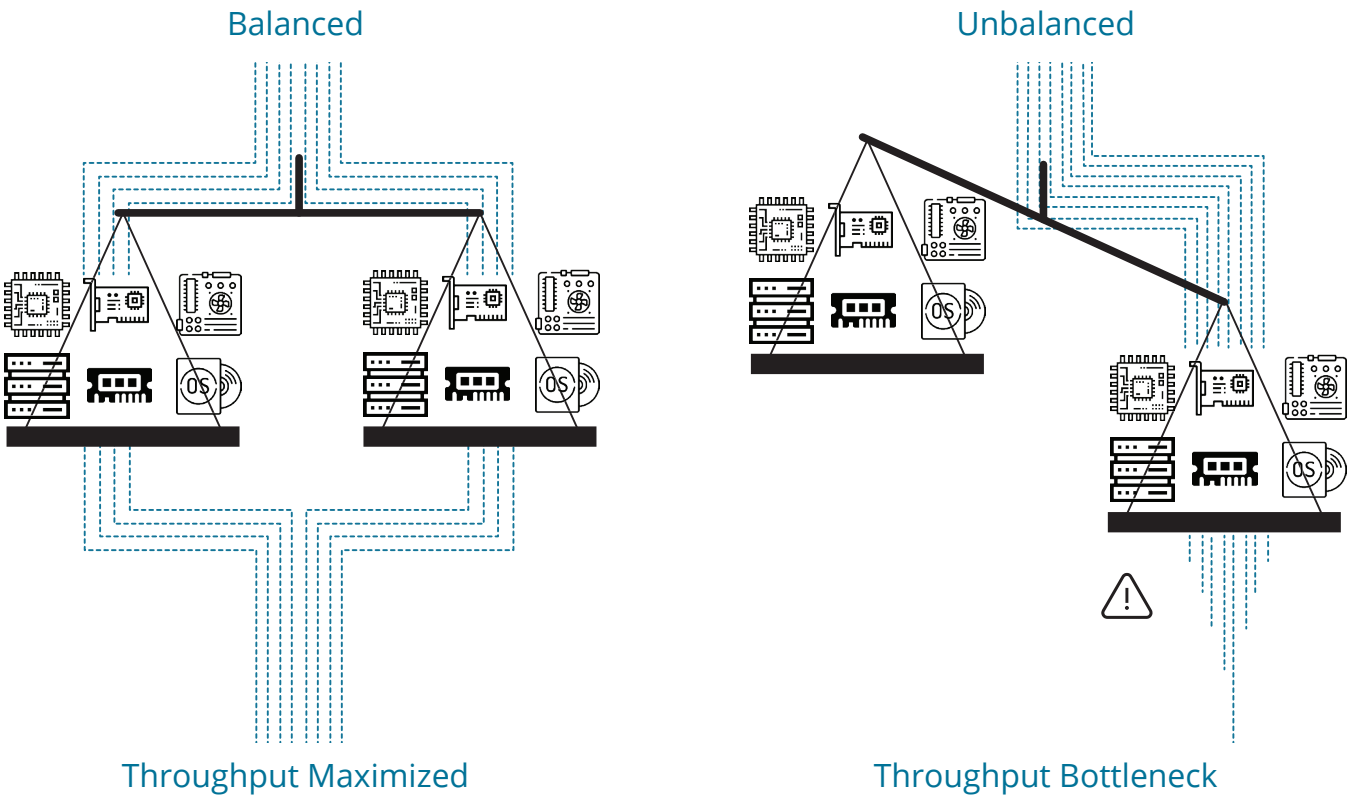
The actual test had the following parameters:

- Containerized VOD/Web CDN Instances (1 per NUMA node)
- 2x100Gbps
- ~5000 reqs/second
- 5MB Video objects

Software

But hardware selection and configuration isn't the only way to optimize server performance. In many cases, the software employed to address the specific requirements of a use case can be configured as well (in the test environment, NGINX acted as a reverse proxy for CDN services).

This configuration must be done on an application-by-application basis, which includes the operating system itself. For example, tuning the TCP stack can affect such things as how each user session is handled and shaped as well as how the server responds to requests. All of these configurations combined—from the OS to each application to even the container itself—work with the hardware optimizations mentioned previously. The result can be a server that is optimally balanced to provide the maximum throughput and performance.



Building a Server Based on the Use Case

Memory, disk, bus, CPU, and the OS/applications all work together to balance an edge server's I/O. The problem is there's no "one-size-fits-all" approach.

There is no toolkit to automagically balance a server's configuration. Imagine if there was an application that let the user select the server's purpose (such as a streaming video edge cache node) and have everything optimized as a result. And that's the fundamental issue: each server must be built based on the use case to ensure it is tailored to meet the specific needs of memory, storage access, throughput, and performance.

In Video Streaming, Lots of Little Objects Complicate the Problem

HTTP segmented delivery, whether it's for a streaming video or a streaming game, can impact the I/O balance in the server. Consider the following scenario:

- Millions of viewers streaming a live event
- Multiple different devices requiring a variety of encoding profiles and ABR ladders
- Transcoding and packaging are done "on-the-fly" in edge server containers

In this scenario, there might be several factors contributing to poor server performance and reduced throughput.

First, multiple containers on the machine requiring the use of different resources (CPU/GPU for transcoding and packaging, memory for temporary storage of rendered segments, disk access for longer-term storage of segments to enable DVR functionality) could result in an imbalance depending on how the CPU, memory, disk, and bus are configured. Second, the rate at which the imbalance is happening can be exacerbated by the type of traffic: very small objects.[16]

Because of the nature of the streaming video, the requests are not one-off. They are a persistent stream, which keeps the server in a constant state of imbalanced I/O. Third, although there are software configuration tools to address I/O imbalance when using containerized resources, the containers themselves may be a root cause of the imbalance by requesting, in parallel, access to unbalanced server components through an un-optimized CPU. Finally, the use of disks which may not be suited for high I/O read use cases (such as spindle rather than NVMe SSD), a high-performance bus, and improperly balanced memory for the number of channels and CPUs could all contribute to less than optimized throughput. All-in-all, it's a recipe for a poorly-performing edge cache node.

Maximizing Edge Server Throughput in the CDN

CDNs need to have high-performing edge servers in order to mitigate latency when replying to device requests. But an imbalanced server does more than just add milliseconds to the round-trip time.

Long-term operation of a server in an imbalanced state forces it to work harder to achieve the same results of an optimized server. This hard work means the server will wear out faster. For that reason, fixing imbalanced I/O will not only improve the overall performance of the edge caches but also prolong the life of the hardware.

Increasing Edge Performance	Description	Impact
Add more server resources (memory, CPUs, disk capacity) to edge cache nodes	Just adding CPUs and/or memory doesn't solve the underlying problem of imbalance. Yes, it can make a server perform better but the server is still only performing at a percentage of its maximum optimization. This also adds to the overall server's initial capitalization and maintenance.	Medium
Replace edge cache node server hardware (disks, NICs, and memory)	This might seem like an easy fix. If the server isn't performing optimally, replace the parts until it does. SSDs are better than spinning disks. Memory can be improved and utilized as transient cache. NICs can be replaced with cards that have higher link speed. Although this might improve the overall server performance, each of these replacements will still be subject to the use case.	High
Increase available bandwidth	Connecting a bigger pipe to a spigot that can only process a certain volume doesn't fix the problem. When the server's I/O is imbalanced, the spigot's size still limits output..	Low
Optimize server application software (including OS)	Configuring and optimizing all of the software on the server can benefit its throughput and performance. But, if the software still resides on a box with a hardware imbalance in one or more of the four component buckets, software optimizations might not be as effective.	Medium

There is no silver bullet to address I/O imbalance or optimization. Each use case, such as streaming video or cloud gaming, requires a specific understanding of how it will employ server resources and adapt to the expectations of end-users. If ultra low latency is part of the user experience, then the server CPU, memory, disk, bus architecture, and software need to be balanced against that requirement. To do that requires not only expertise and experience, but a willingness to do the work.

With Intequus, You Get the Edge Servers Your Use Case Demands

Many cloud service providers are built on generic servers with commodity hardware and software. Their strategy is about homogeneity: it's easier to manage a lot of physical machines if they are all the same; same hardware, same software, and same configurations. These same providers often provide management software for the containers and software on those boxes, ensuring customers can optimize their own configurations to match use case requirements.

But as discussed in this white paper, that's not enough. Intequus knows that software, whether the container or the OS on the server, is only one of many components which needs to be optimized to ensure I/O balance and peak performance. Rather than providing a homogenous cloud environment with commodity hardware, Intequus CS customizes specific servers for specific use cases.

That attention to detail, coupled with years of expertise in balancing server hardware and software components for optimum I/O throughput and performance, makes Intequus CS an ideal provider for discerning customers.

If the application you are deploying in the cloud isn't critical to the user experience, then perhaps it can be made available on generic cloud hardware. But if you are a CDN, then your edge cache nodes are the core of your offering. They must be balanced with specific configurations of hardware and software.

When you work with Intequus CS engineers, they begin by asking the critical questions:

- "What is your use case? Does it require ultra low-latency? Does your software read and write to disks frequently?"
- "How will server performance impact your end users?"
- "What kind of software are you planning on deploying? Is it containerized?"

The answers to these questions give Intequus CS engineers critical information to build your server profile. They not only want to understand the server resource demands of your application, but also the traffic your application will handle. An application that relies heavily on memcache and database read/writes is vastly different from an application that continually responds to TCP requests with chunks of data.

Conclusion

Solving edge server performance and throughput optimization by building bigger boxes (with more CPUs and memory) or replacing NICs, storage devices, and memory with faster versions doesn't address the underlying imbalance. Yes, the server may perform marginally better under certain use cases, but each of those components needs to be balanced in relation to the resource demands being placed on the server. And once the hardware is balanced, the software needs to be optimized as well, ensuring requests for server resources are handled in an efficient and effective manner through the CPU and bus architecture.

But addressing imbalanced I/O will have meaningful impacts to your application performance. First, it will likely reduce round-trip time, which can correlate directly to end-user satisfaction. Second, it may also reduce long-term capital costs associated with server maintenance (machines may be able to stay in rotation for much longer before needing replacement). Finally, by distributing within the edge you can help mitigate downtime and improve overall resiliency (especially against cyberattack).

Balancing I/O through hardware and software optimization is too important to ignore. As use cases such as streaming video and multi-player gaming grow and new use cases that need very low round-trip times (such as IoT or autonomous vehicles) become more prevalent, it will be important for service providers like CDNs and ISPs to have the most efficient and effective edge servers.

From the memory to the CPU, each hardware element must be balanced first through selection (which technology or product is the right one for the use case) and then through configuration. Then software, from the applications to the OS, must be optimized for the hardware it will interface with as well as any specific use case requirements (such as a specific traffic profile). That can only be done by partnering with a cloud provider like Intequus who has the expertise, the experience, and the willingness to ensure your servers are operating at peak performance for your specific needs.

About Intequus

Intequus is a leading provider of configurable compute and storage solutions. Headquartered in Minnesota for 30 years, Intequus has established itself as a key partner to the world's largest CDNs. Intequus's expert engineering, high-quality manufacturing, experienced program management, and personalized technical support give CDNs the tools they need to be the leaders in their field.