The American Journal of Interdisciplinary Innovations and Research
ISSN 2642-7478 | Open Access

Type Original Research
PAGE NO. 123-136
DOI 10.37547/tajiir/Volume07Issue07-11

Designing a Reliable, Ultra-Low Latency Data Access Environment for Real-Time Applications in Modern Data Centers

Ajay Prasad
Independent researcher, USA

OPEN ACCESS

SUBMITED 19 June 2025
ACCEPTED 25 June 2025
PUBLISHED 28 July 2025
VOLUME Vol.07 Issue 07 2025

CITATION
Ajay Prasad. (2025). Designing a Reliable, Ultra-Low Latency Data Access Environment for Real-Time Applications in Modern Data Centers. The American Journal of Interdisciplinary Innovations and Research, 7(07), 123–136. https://doi.org/10.37547/tajiir/Volume07Issue07-11

Abstract- Achieving ultra-low latency (ULL) with end-to-end delays of 1–5 milliseconds is vital for real-time applications such as high-frequency trading, autonomous vehicles, and personalized e-commerce. This study defines latency as the time from initiating a data processing task to receiving its result, proposing a holistic approach to ULL through optimized hardware, software, and network components. Latency is broken down into network, I/O, processing, queuing, application, and security factors, reducing the standard ~19 ms latency to below 5 ms with targeted enhancements. Key strategies leverage high-performance hardware (NVMe SSDs, FPGAs, GPUs), low-latency interconnects (InfiniBand with RDMA), and efficient software. A real-time fraud detection scenario handling 10,000 concurrent queries per second is analyzed, detailing tiered technology stacks. The study contrasts networking protocols, emphasizing InfiniBand's sub-microsecond latency advantage for ULL, and demonstrates feasibility with edge infrastructure, dedicated instances, and RDMA-enabled NVMe-oF. This framework offers practical guidance and cost estimates for 2025 ULL implementations, acknowledging that actual latency, performance, and costs may vary by use case.

Keywords: Ultra Low Latency, Standard Latency, NVMe, NVMe-oF, SSD, FPGA, Low Latency Interconnect, InfiniBand, Edge Infrastructure GPU, RDMA, Optimized Hardware, Targeted optimizations, Fast API, Distributed Solution.

1. Introduction

Latency is a delay or a wait. It's the time it takes for something to happen after you request it. It is the period between when we press a button or send a message and when we see the result or receive a reply. For example, when we click a link on a website, latency is the time from our click until the page fully appears. In computer networks, latency is how long it takes for a piece of data to travel from one point to another. High latency means there is a long delay, which can make things feel slow or unresponsive. For instance, a long delay can cause websites to load slowly, interrupt videos, or make online applications unusable. Ideally, latency should be as close to zero as possible to provide a good experience, typically in the range of 1–100 milliseconds (ms), which can be achieved by optimizing processing cycles and network transmission.

ULL refers to the capability of systems and technologies to achieve the most minimal delay during transaction processing. It is an extremely short, end-to-end latency of 1–5 milliseconds or less [1] for time-critical applications requiring near-instantaneous responses. Financial trading systems require ultra-low latency to execute trades in real-time, often measured in microseconds. The acceptable latency for high-frequency trading (HFT) is typically below 100 microseconds (µs), with some systems targeting latencies as low as 10 µs. Minimal network delay allows for nearly instantaneous operations. Beyond just speed, ULL also encompasses the reliability and consistency of transaction processing, which is essential for maintaining confidence in these systems. [2] This is much faster than typical "low latency" systems.

The demand for ULL, with end-to-end delays of 1–5 ms, is critical for real-time applications like high-frequency trading and personalized e-commerce in modern data centers, where tiered architectures (GTM, Presentation, Application, Data, Storage) handle thousands of concurrent requests. However, a gap exists in tailoring ULL strategies to each tier, as current approaches often overlook interdependencies and unique requirements, leading to suboptimal performance. This document bridges this gap by offering a comprehensive framework to optimize hardware (NVMe SSDs, FPGAs), low-latency interconnects (InfiniBand with RDMA), and software stacks across all tiers, targeting a 3 ms latency for scenarios like 10,000 queries per second fraud detection as of 2025.

A very simple mathematical way to calculate standard latency and ULL is explained in table 1.

Table 1: Simple Latency Calculation

Aspect	Standard Latency	ULL
Network Latency	1 ms	20 µs
I/O latency	3 ms	50 µs
Processing Latency	5 ms	500 µs
Queuing Latency	5 ms	200 µs
Application Latency	3 ms	2000 µs
Security Latency	2 ms	230 µs
Total Latency	19 ms	~3000 µs = 3 ms

Use cases and some specialized hardware that makes ULL possible in explained in the table 2.

Table 2. Latency and ULL

Aspect	Latency	ULL
Use cases	Standard IoT such as smart home sensors, General-purpose databases, Moderate I/O and processing	Real-time personalized ecommerce analytics, High-frequency trading, Real time fraud detection, Autonomous Driving
Performance Requirements	Standard hardware, such as consumer SSDs. For example, a PostgreSQL query on a consumer NVMe SSD (Samsung 970 EVO) with ~50 µs I/O latency, total query time ~5-20 ms due to network and processing.	High-performance hardware (enterprise NVMe SSDs, GPUs,FPGA,Infiniband network). For example, a RocksDB query on an Intel Optane NVMe SSD with ~10 µs I/O latency, total query time ~50-100 µs with edge processing and optimized software.

Investing in ULL infrastructure dramatically enhances real-time data analytics by reducing processing delays, increasing data throughput, and enabling rapid decision-making crucial for applications like high volume high frequency financial trading, telecommunications, and industrial IoT. Key technologies such as edge computing minimize latency by processing data closer to its source, while FPGA-based solutions efficiently handle large volumes of streaming data for ultra-reliable, low-latency communication Network-centric approaches, including software-defined networking and programmable switches, InfiniBand network, RDMA , further accelerate analytics by offloading tasks to the network, improving data flow and load balancing. Additionally, scalable architectures using tools like Kafka, Flink, Cassandra, and Kubernetes ensure high throughput and sub-second latency, supporting continuous data flow and instant insights required for time-sensitive workloads.

While ULL is a critical component of modern AI infrastructure of real time analytics it is important to consider the challenges and limitations associated with its implementation. Factors such as network coverage, resource contention, and data synchronization can impact the effectiveness of low-latency solutions. Moreover, the deployment of advanced technologies like 5G and edge computing requires significant investment and infrastructure development. Despite these challenges, the pursuit of ultra-low latency remains a priority for enhancing the performance and reliability of AI-driven applications across various domains.

2. Causes of latency

Latency is caused by a combination of **hardware limitations**, such as outdated processors, insufficient RAM, or slow hard drives, which can bottleneck data processing and slow down task handling, **network bandwidth and congestion** [3], where limited bandwidth, high traffic, or physical distance between servers increases data transfer times and delays communication, and **software inefficiencies**, including poorly optimized code, inefficient database queries, and operating system overhead, all of which add extra processing steps and slow down data handling. **Large data volumes and complexity**, like processing big datasets or real-time streaming demand more resources and time, further increasing latency. **Network delays** arise from protocol overhead, packet loss, and the cumulative effect of intermediary devices such as routers and switches, each introducing small delays. **I/O**

bottlenecks [4] from slow disk operations or contention for shared resources in multi-user environments can also be significant sources of latency. **Concurrency and resource contention**, where multiple users or processes compete for CPU, memory, or disk, lead to queuing and waiting, while poorly managed parallel processing can create additional bottlenecks. **Data pipelining and workflow dependencies** [5] mean that a delay in one stage can hold up the entire process, and reliance on **external systems** like third-party APIs introduces unpredictable response times. Configuration and tuning issues, such as improper buffer sizes, caching strategies, or lack of load balancing, can lead to inefficiencies and uneven performance. Finally, external factors like environmental issues (power fluctuations, overheating) and security measures (encryption, authentication) add further processing overhead, all contributing to increased latency.

2. Understanding latency in a tiered architecture and finding the right topology

A typical multi-tier application architecture, as illustrated in Fig1, is foundational for organizations , public or private, that support users accessing data, performing online transactions, or engaging with social media. This comprehensive stack highlights the necessity of considering all architectural layers to achieve performance, reliability, and scalability goals. Additionally, cost remains a critical factor influencing architectural decisions in any data center environment.

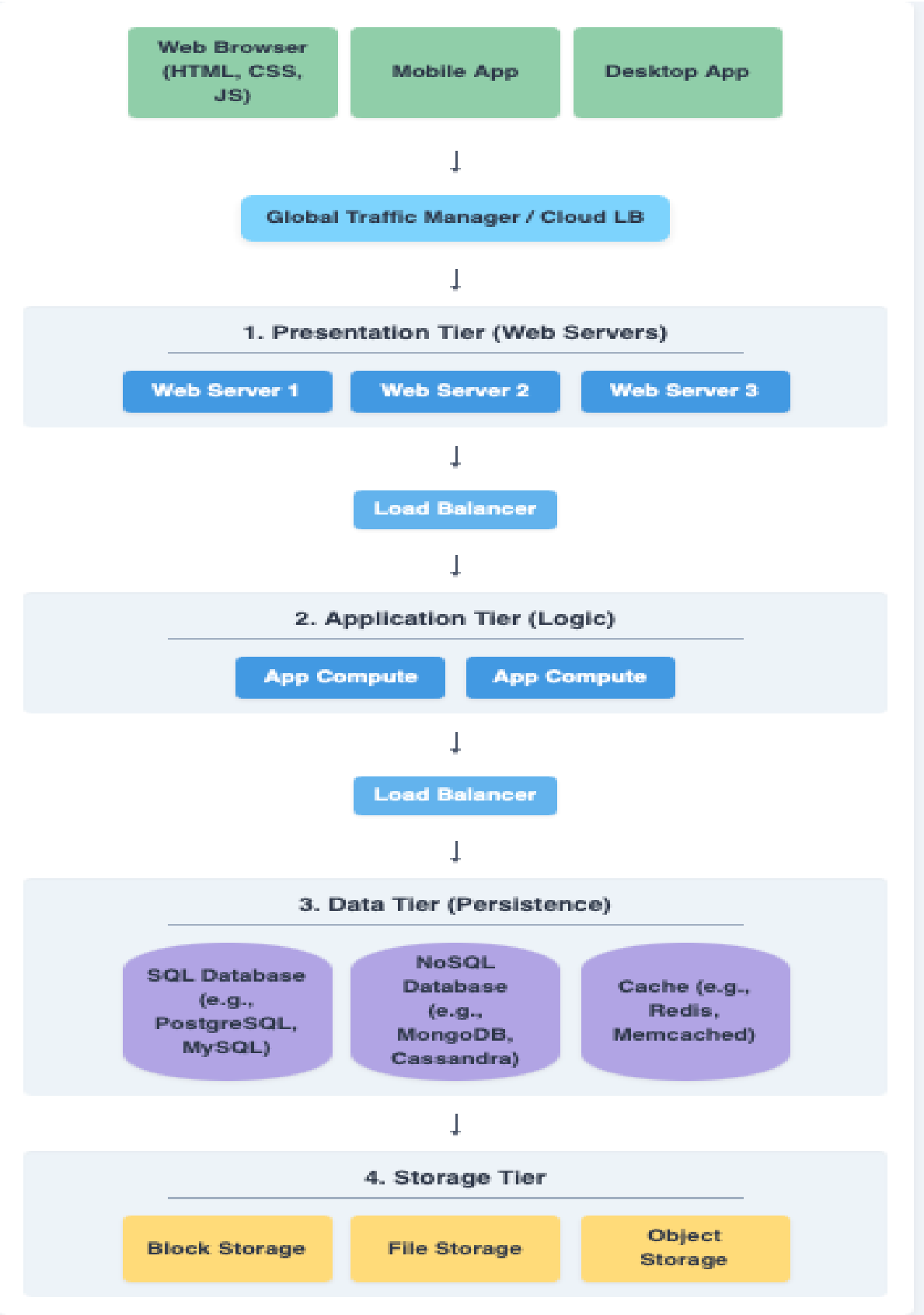

Figure 1. Today's tiered architecture

One of the most important aspects of achieving ULL is addressing data volume and complexity related to I/O bottlenecks. This is directly related to capacity planning, which is outside the scope of this discussion at the moment, but it cannot be avoided when talking about latency.

Any hardware or software an organization purchases

typically has a lifespan of roughly five years. In today's fast-moving world, technology changes quickly, new and improved hardware becomes available, and data volume continues to grow often beyond our initial estimates. Therefore, when aiming for ultra-low latency, we must evaluate existing compute capacity, forecast demand, and project the throughput requirements of egress and ingress devices ,such as, network switches, routers, NICs, HBAs, and HCAs,for the next five years as part of the planning process.

Network provides the foundation for all devices to connect with each other. Ultra-low-latency and reliable communications is perhaps the most challenging task in future [6]. In most of data centers, the network is generally built on mainly three protocols, TCP/IP, InfiniBand (IB), and Fiber Channel Protocol (FCP). Table 2 does a comparative study between them.

Table 2. Comparison between TCP/IP, FCP and IB protocols

Aspect	**TCP/IP**	**InfiniBand (IB)**	**Fibre Channel Protocol (FCP)**
Understanding	A suite of protocols for general-purpose networking, widely used for internet and LAN communication.[7]	InfiniBand Architecture is a high-performance networking standard developed in 1999, providing higher reliability, availability, performance, and scalability compared to traditional interconnect technologies like Ethernet.[8]	A protocol for high-speed data transfer, primarily for storage area networks (SANs), running over Fibre Channel (FC) hardware.[9]
Architecture	Uses Ethernet, layered stack (Application, Transport, Network, Link).[7] Has a send and ach mechanism for packets.	Switched fabric topology with point-to-point connections.[8], Supports RDMA without CPU Involvement	Runs over Fiber Channel, a dedicated storage network [9]. Uses a channel-based model for direct SCSI communication.
Latency	2-20 ms for network-bound database queries. TCP ACKs, which are small control packets without application payload, can consume a substantial portion of CPU cycles on the receive side[10]	0.5-5 µs end-to-end, RDMA bypasses OS reducing overhead [11]	Low latency, 10-50 µs, better than TCP/IP but higher than IB due to SCSI overhead.
Bandwidth	Upto 400 Gbps [12]	Up to 800 Gbps (NDR, 2024), with roadmap to 1600 Gbps. [13]	Up to 64 Gbps (32GFC/64GFC), sufficient for storage but lower than IB. [14]

Scalability and Reliability	Lossy by design and highly scalable less efficient for HPC	Highly scalable and ideal for HPC and AI clusters	Scalable for SAN but less flexible general networking
Use cases	General purpose networking (LAN,Cloud,Internet)	HPC, ULL network, NVMe-oF	Used for SAN and high-speed disk access for Databases
Cost	Low Cost	High, requires specialized hardware mainly NVIDIA	Moderate to high
Ease of Deployment	Easy to deploy, wide compatibility, easily available skill set	Complex, needs specialized skills	Familiar to SAN admins

As the study suggests TCP/IP is designed for compatibility and reliability [7], but its kernel-based processing and byte-stream model increase latency, making it less suitable for ultra-low latency. InfiniBand Offers microsecond latency and high bandwidth via RDMA, ideal for HPC edge AI [8], autonomous vehicles and real-time analytics, as emphasized in the table 2. FCP, Provides low latency for storage but is limited to SAN environments [9], with less flexibility than IB for general-purpose or AI-driven workloads. InfiniBand (IB) is a high-speed interconnect technology that is well-suited for ULL applications.

RDMA substantially reduces latencies compared to TCP by enabling direct data transfer between application memory on different machines without involving the CPU or operating system on the data path. This hardware-level support, seen in technologies like RDMA over InfiniBand and RDMA over Converged Ethernet (RoCE), can cut TCP's latency by up to 10 times [15].

In a data center it not entirely possible to build a ULL environment just on InfiniBand. The presentation tier mentioned in Fig 1, web servers, APIs, often relies on TCP/IP for compatibility with client devices (browsers, mobile apps) and external networks (internet, cloud). Most web applications are built using HTTP/HTTPS over TCP/IP, making it challenging to use InfiniBand directly for client-facing communication without additional integration.

For the Application tier, it's possible to design applications using InfiniBand's native verbs API or IP over InfiniBand (IPoIB), which allows TCP/IP applications to run over IB with reduced latency on Ethernet. For ULL applications (HPC, real-time analytics), developers can use RDMA-based frameworks (UCX, libfabric) to bypass TCP/IP entirely, reducing latency.

For the Data and Storage tier InfiniBand is the ideal solution for ULL. It can be used as NVMe-of (NVMe over Fabrics) or SRP (SCSI RDMA Protocol) for database-to-storage access. RDMA minimizes CPU overhead, making IB ideal for ultra-low latency.

4. Designing ULL environment in the data center

To achieve ULL in environments characterized by high data volume and complexity, it is essential to optimize every layer of the system through a combination of advanced hardware and intelligent architecture. This begins with maximizing compute capacity by utilizing high-core CPUs and accelerators such as FPGAs or GPUs, which enable efficient parallel processing and handle demanding workloads with ease. Low-latency interconnects and streamlined data paths, tailored to the needs of each system layer, further enhance performance by ensuring that data moves quickly and efficiently throughout the infrastructure.

A critical component in minimizing network latency is the adoption of RDMA technologies, such as InfiniBand or RoCEv2. RDMA provides high performance communication with a direct connection between a local

and remote memory. It performs zero-copy operations without the involvement of operating systems by exploiting a hardware NIC to process transport logic [16]. By allowing direct memory-to-memory data transfers between servers, RDMA eliminates traditional bottlenecks and significantly reduces both latency and CPU overhead, as shown in Fig 2. For storage, deploying NVMe SSDs and leveraging in-memory caching solutions like Redis or in-memory databases can dramatically decrease data access times, ensuring that applications receive the information they need without delay.

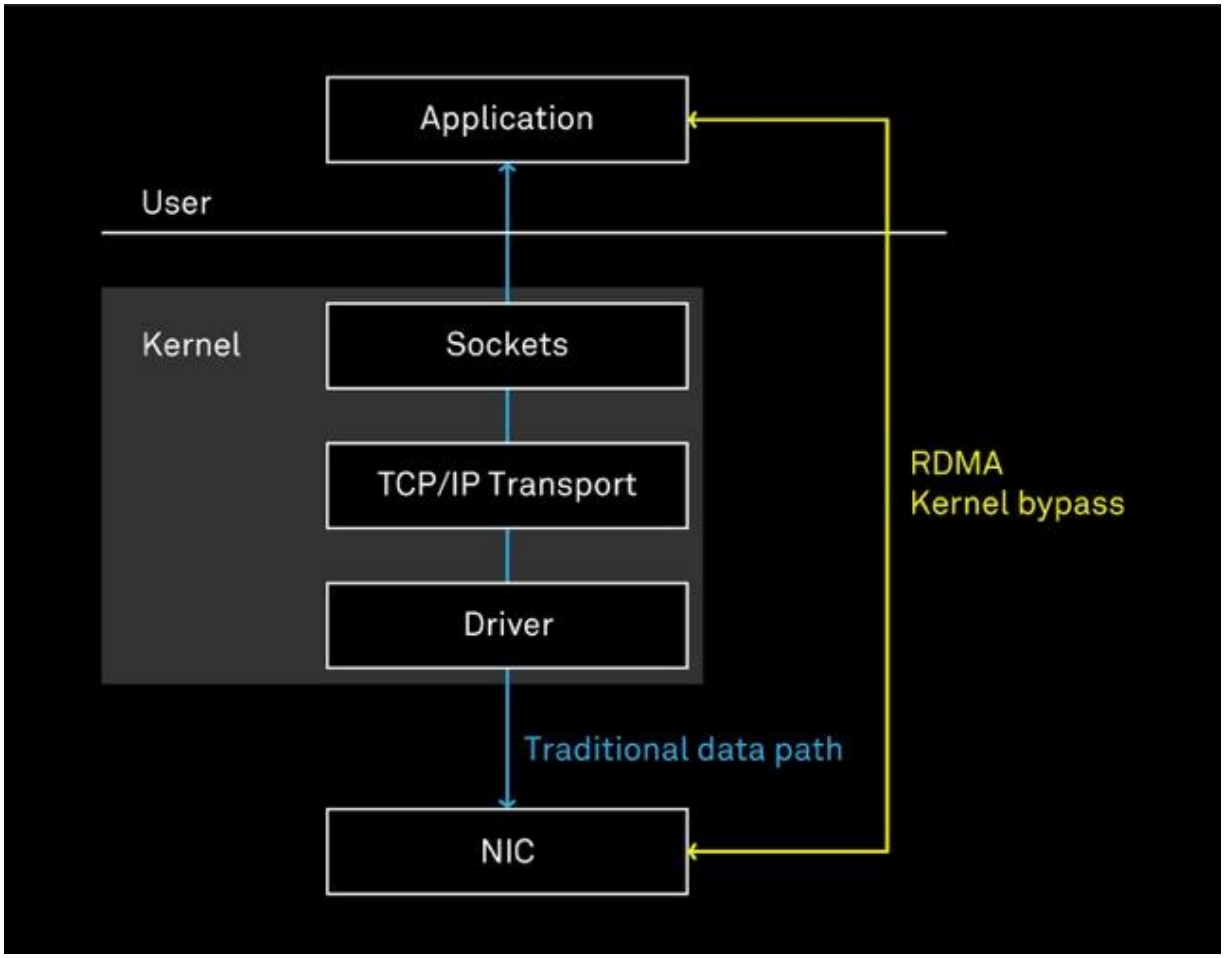

Fig 2. RDMA kernel bypass [17]

To maintain high performance during peak loads, such as handling 10,000 queries per second scaling the system with distributed architectures like Ceph is vital. This approach distributes the workload evenly and prevents any single component from becoming a bottleneck. Continuous monitoring of latency across all layers, using tools like Prometheus, allows for the proactive detection and resolution of performance issues, ensuring that ultra-low latency is consistently maintained. By integrating these strategies, organizations can build robust, high-throughput systems capable of supporting the most demanding, data-intensive applications.

4.1 Choosing the right hardware

Referring to Fig 1, Dell, HPE, and Supermicro do not manufacture dedicated hardware load balancers as standalone products, but they provide server hardware and software solutions that can be configured for load balancing in ULL environments, such as a real-time fraud detection system processing 10,000 concurrent queries per second or more. These solutions leverage high-performance servers with software load balancers and accelerators (BlueField-3 DPU) to achieve extremely low load balancing latency in the Global Traffic Management (GTM) and Presentation Tiers, contributing to a total system latency of ~3 ms (3000 µs). Table 3 discusses a all tiers mentioned in Fig 1 and provides hardware choices for each layer.

Table 3. Hardware choices for a tiered architecture

Layer	Compute Capacity Requirements	Hardware Choices	Latency Impact	Notes
Global Traffic Manager/Cloud LB	Clustered nodes to distribute 10,000 queries/s across regions, high-core CPU for routing, 256 GB RAM, FPGA/DPU for offloading , dual 100G Ethernet NICs,client-facing RDMA internal.	Dell R660/HPE DL360/Supermicro SYS-621U, EPYC 7232P/Xeon 4310, 128G GB DDR5, Xilinx Alveo U50 /NVIDIA BlueField-3, Intel E810-CAM2/NVIDIA ConnectX-7.	~100 µs (50 µs processing + 50 µs queuing)	Manages global traffic, hands off to Presentation LB, requires RDMA to internal tiers for ULL.
Presentation Tier (Web Servers)	10 nodes to handle HTTP/API traffic, 16-24 core CPUs, 256 GB DDR5, FPGA/DPU for packet processing, 100G Ethernet/RDMA.	Dell R660/Lenovo SR650/Supermicro SYS-121U, EPYC 7313P/Xeon 5318Y, 128GB DDR5, Xilinx Alveo U200 /BlueField-2, ConnectX-7/Thor 2.	~200 µs (100 µs processing + 90 µs queuing + 10 µs network)	10 nodes scale for 10,000 queries/s, RDMA to Application Tier ensures ULL.
Application Tier (Logic)	5 nodes , 24-32 core CPUs, GPUs/FPGAs for inference, 256GB DDR5, RDMA to Data Tier.	Dell R760/HPE DL380/Lenovo SR670, EPYC 7413/Xeon 8358, 256GB DDR5, NVIDIA A100 /AMD MI300X /Xilinx U280, ConnectX-7/Thor 2.	~1000 µs (600 µs processing + 350 µs queuing + 50 µs network)	5 nodes handle complex logic, GPUs accelerate AI, RDMA maintains ULL.
Data Tier (NVMe Optimized)	2 HA nodes for SQL/NoSQL (PostgreSQL, MongoDB, Cassandra) and caches (Redis, Memcached), 32-core CPUs, 512GB DDR5, NVMe SSDs,	Dell R760/Supermicro SYS-221U/Lenovo SR660, EPYC 7543/Xeon 8362, 512GB DDR5, NVIDIA A100/AMD MI250/Xilinx U250, 4x Intel Optane P5800X /Samsung	~800 µs (300 µs processing + 200 µs queuing + 200 µs I/O + 50 µs network + 50 µs application + 0 µs security)	NVMe-optimized for ULL (~111.5 µs total), HA with replication, accelerators reduce query latency.

	GPUs/FPGAs for query acceleration, RDMA.	PM9A3, ConnectX-7/Thor 2.		
Storage Tier (Block, File, Object)	100 nodes (distributed) or 1-2 arrays, 16-core CPUs, 256GB DDR5, NVMe SSDs RDMA for access.	Supermicro SYS-621U/HPE Apollo 4200/Pure FlashArray, EPYC 7232P/Xeon 4310, 256GB DDR5, 24x Kioxia CM7 /Intel Optane, ConnectX-7/Thor 2.	~900 µs (200 µs I/O + 600 µs processing/queuing + 100 µs network)	Distributed (Ceph/ONTAP) or arrays (Pure/Dell) scale for high volume, ULL with NVMe-oF.

5. Discussion

As a study, a real time fraud detection that handles approximately 10000 queries per second is being discussed in the following section.

5.1 Real time fraud detection

Fig 3 discusses a real-time fraud detection system processing approximately 10,000 concurrent queries per second with an ULL target of ~3 ms. This also tries to provide the cost estimates and trade-offs.

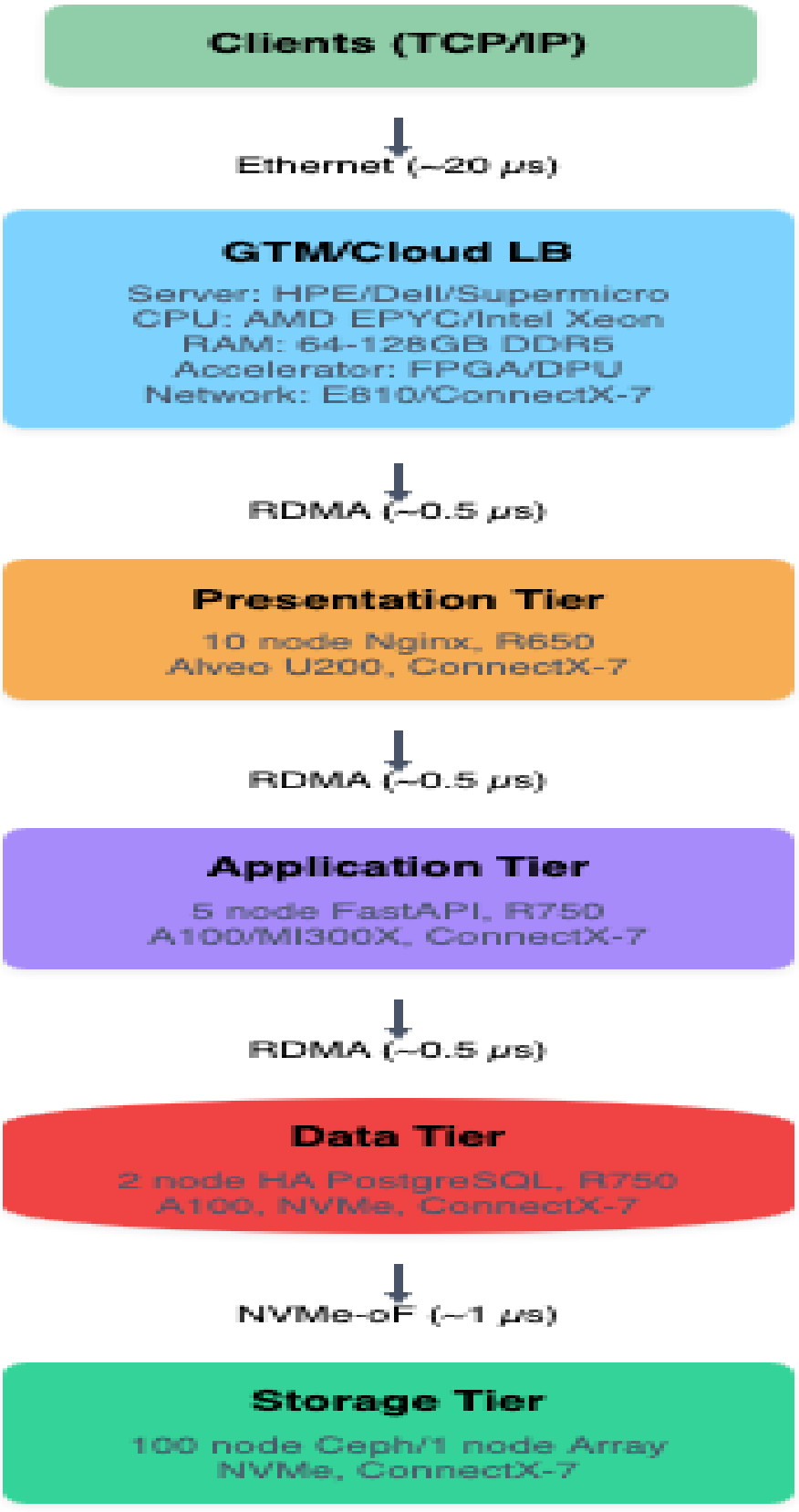

Fig 3. Tiere architecture of real time fraud detection system

The Global Traffic Manager or Cloud Load Balancer (GTM/Cloud LB) distributes 10,000 queries per second globally across regions or cloud providers, ensuring high availability and low latency. For improved routing capacity, upgrading to the EPYC 7313P with 16 cores is recommended over the EPYC 7232P with 8 cores.

Enhanced offloading is supported by FPGA or DPU options such as the Alveo U55 and BlueField-3. For network performance, the ConnectX-8 network interface card, which is 200G capable and costs approximately $2,800, can be used to improve RDMA performance and maintain latency at 100 microseconds.

The Presentation Tier (Web Servers) handles HTTP and API requests at a rate of 10,000 queries per second. For CPUs, retaining either the EPYC 7313P or Xeon 5318Y with 16 to 24 cores is sufficient for a 10-node deployment. For packet processing, upgrading to the Alveo U200 (2025 version, approximately $3,500) and BlueField-3 is recommended. Network performance is enhanced with ConnectX-8 or Thor 3 100G NICs utilizing RDMA, which reduces network latency to 10 microseconds. This architecture can scale to 10,000 or more queries per second, with RDMA ensuring total latency remains around 200 microseconds.

The Application Tier (Logic) executes AI-driven fraud detection. For CPUs, upgrading to the EPYC 7443P with 24 cores or the Xeon 8358 with 32 cores provides strong AI inference capabilities. For GPU and FPGA acceleration, retaining the A100, MI300X, or U280 is recommended, with the A100 80GB model suited for 2025 AI workloads. Network performance is supported by ConnectX-8 or Thor 3 NICs with RDMA, maintaining latency at 1,000 microseconds. This tier handles complex logic across five nodes, with GPUs accelerating fraud detection processes.

The Data Tier (Persistence) manages SQL and NoSQL databases as well as caching systems. For CPUs, retaining the EPYC 7543 or Xeon 8362 with 32 cores is recommended for high-availability nodes. For query acceleration, updating to GPUs or FPGAs such as the A100 80GB, MI250, or U250 provides significant performance benefits. Storage performance is enhanced by upgrading to Optane P5801X (2025 model, approximately $12,500) and PM9A3 ($2,200) SSDs, which deliver I/O latencies as low as 200 microseconds. Network connectivity is supported by ConnectX-8 NICs with RDMA, maintaining network latency at 800 microseconds. This architecture is optimized for NVMe-oF, supporting a total latency of 3 milliseconds, while high availability ensures reliability and consistent data access.

The Storage Tier (Block, File, Object) provides scalable storage for the system. For CPUs, upgrading to the EPYC 7313P or Xeon 4316 with 16 cores is recommended for deployments of up to 100 nodes. Storage devices include retained Kioxia CM7 and Optane P5810 SSDs, with Optane models priced at approximately $75,000 for high-capacity needs. Network connectivity is handled by ConnectX-8 NICs for 100G RDMA, supporting network latency of 900 microseconds. This configuration is designed to scale distributed storage effectively while maintaining ultra-low latency through NVMe-oF optimization.

5.2 Latency estimate and validation

Total Latency= 100 µs (GTM) + 200 µs (Presentation) + 1000 µs (Application) + 800 µs (Data) + 900 µs (Storage) = 3000 µs (3 ms), meeting the target.

The hardware configuration supports scalability for 10,000 queries per second across all tiers, with deployments including 10 Presentation nodes, 5 Application nodes, 2 high-availability Data nodes, and 100 Storage nodes. Ultra-low latency is achieved using RDMA with ConnectX-8 NICs and NVMe-oF with Optane P5801X SSDs, ensuring consistent performance across the entire architecture.

5.3 Cost Estimation and trade off

The cost estimate ranges from $500,000 for basic configurations to approximately $2 million for setups with premium GPUs and SSDs. Trade-offs include the higher expense associated with high-end CPUs and GPUs, as well as the additional infrastructure required to support RDMA, all of which contribute to increased overall costs.

6. Conclusion

In this paper, I have provided a comprehensive framework for achieving end to end Ultra Low Latency (ULL) in the range of 1-5 ms, an essential requirement for modern, real-time applications such as fraud detection and personalized e-commerce.

By analyzing each layer of the technology stack, from hardware and networking to software and storage, I have demonstrated that ULL is attainable through

targeted investments in high-performance components like NVMe SSDs, FPGAs, GPUs, and InfiniBand with RDMA. Real-world scenarios and cost estimates highlight both the technical feasibility and the financial considerations of deploying such systems at scale. While the pursuit of ULL demands careful planning and significant resources, the actionable strategies outlined here empower organizations to design and implement high-performance, data-intensive architectures capable of meeting the most demanding latency requirements.

References

[1] Hazarika, Ananya, and Mehdi Rahmati. "Towards an evolved immersive experience: Exploring 5G-and beyond-enabled ultra-low-latency communications for augmented and virtual reality." *Sensors* 23.7 (2023): 3682.

[2] Murthy, Pranav, and Aditya Mehra. "Exploring neuromorphic computing for ultra-low latency transaction processing in edge database architectures." *Journal of Emerging Technologies and Innovative Research* 8.1 (2021): 25-26.

[3] Cardwell, Neal, et al. "Bbr: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time." *Queue* 14.5 (2016): 20-53. Link: https://dl.acm.org/doi/pdf/10.1145/3012426.3022184

[4] Title: What is Latency? IBM resource, Link: https://www.ibm.com/think/topics/latency

[5] A. Khan *et al*., "Hvac: Removing I/O Bottleneck for Large-Scale Deep Learning Applications," *2022 IEEE International Conference on Cluster Computing (CLUSTER)*, Heidelberg, Germany, 2022, pp. 324-335, doi: 10.1109/CLUSTER51413.2022.00044.

[6] M. I. Ashraf, M. Guizani, V. G. Menon and S. Mumtaz, "Series Editorial: Ultra-Low-Latency and Reliable Communications for Future Wireless Networks," in *IEEE Communications Standards Magazine*, vol. 6, no. 1, pp. 42-43,

[7] Tyagi, Ashima. "Tcp/ip protocol suite." *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol* 6.4 (2020): 59-71

[8] Sinha, U., Dash, R., Kopri, N.J. (2018). A Gateway Virtual Network Function for InfiniBand and Ethernet Networks. In: Negi, A., Bhatnagar, R., Parida, L. (eds) Distributed Computing and Internet Technology. ICDCIT 2018. Lecture Notes in Computer Science(), vol 10722. Springer,

[9] J. R. Heath and P. J. Yakutis, "High speed storage area networks using a fibre channel arbitrated loop interconnect," in *IEEE Network*, vol. 14, no. 2, pp. 51-56, March-April 2000, doi: 10.1109/65.826372.

[10] W. Bai, K. Chen, H. Wu, W. Lan and Y. Zhao, "PAC: Taming TCP Incast Congestion Using Proactive ACK Control," *2014 IEEE 22nd International Conference on Network Protocols*, Raleigh, NC, USA, 2014, pp. 385-396, doi: 10.1109/ICNP.2014.62.

[11] Gamess, Eric, and Humberto Ortiz-Zuazaga. "Low level performance evaluation of InfiniBand with benchmarking tools." *International Journal of Computer Network and Information Security* 8.10 (2016): 12.

[12] ConnectX NICs NVIDIA Ethernet Adapters Overview Link: https://www.nvidia.com/en-us/networking/ethernet-adapters/

[13] NVIDIA ConnectX InfiniBand Adapters NVIDIA InfiniBand Adapters Overview Link: https://www.nvidia.com/en-us/networking/infiniband-adapters/

[14] LPe38102 SecureHBA FC Adapter Broadcom LPe38102 Fibre Channel HBA Overview Link: https://www.broadcom.com/products/storage/fibre-channel-host-bus-adapters/lpe38102

[15] Xue, Jaichen, et al. "Fast congestion control in RDMA-based datacenter networks." *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos*. 2018.

[16] S. Lee, Y. Kim, H. Woo and I. Yeom, "Efficient User-Level Multi-Path Utilization in RDMA Networks," in *IEEE Access*, vol. 9, pp. 127619-127629, 2021, doi: 10.1109/ACCESS.2021.3110840.

[17] An Introduction to RDMA (Remote Direct Memory Access) Filip Milovanovic Link:

https://elements.tv/blog/an-introduction-to-rdma-remote-direct-memory-access/