White paper **Juniper Networks + VAST Data**

Retrieval-Augmented Generation (RAG) Architectures for Generative AI

Deploy and manage local RAG-based architectures

JUNIPER NETWORKS

www.juniper.net

 2000833-001-EN April 2025

Contents

01 Executive summary

Generative AI (GenAI) is transforming the way we do business. In enterprise, retrieval augmented generation (RAG) is a convenient and powerful tool for bringing localized knowledge to pretrained, consumable language models. Implementation of a RAG architecture requires access to local vector databases, and while there are a variety of tools out there for RAG deployments, discussion around the specific implications to the network design are often omitted. This paper explores and illustrates how a local RAG-based architecture can be easily deployed and managed using Juniper switches and network fabrics. It also discusses using Apstra Data Center Director to design the same network fabric to accommodate both inference queries and RAG-based database queries with minimal complexity.

This white paper is intended for IT infrastructure decision-makers and technical influencers in an enterprise involved in deploying AI infrastructure at scale, specifically network architects, infrastructure engineers, enterprise architects, and AI platform owners. Whether you're designing high-performance fabrics to support RAG workloads or seeking to streamline deployment and management of AI-ready networks, this paper offers practical insights into how Juniper Networks and VAST Data can help simplify and scale your RAG infrastructure.

02 Introduction

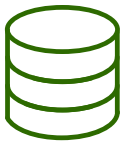
Large language models (LLMs) bring huge advantages to enterprise businesses, providing both general knowledge and localized knowledge through fine-tuning. LLMs can serve a wide range of purposes—from acting as conversational chatbots for customer service to functioning as powerful knowledgebase tools and much more. But LLMs on their own have severe limitations: They can suffer from outdated or incorrect information, they (without fine-tuning) lack proprietary knowledge, and they can be prone to providing answers that are only partially correct—sometimes fabricating aspects of a reply, a process colloquially known as 'hallucinations.'

RAG facilitates the creation of customized responses from a pretrained model, most commonly an existing foundational model such as Meta's Llama, Google's Gemma, etc. RAG allows an external data source to be used to augment the query prompt with contextually relevant data. This method allows the LLM to utilize this additional data when forming its response to the query.

For its external data source, RAG most often (but not always) utilizes a vector database, storing localized content in high-dimensional vector embeddings. These databases are highly scalable and enable efficient search, making them well suited for powering the retrieval stage of RAG.

The retrieval of data from the vector database is a high-speed, latency-sensitive operation. Therefore, it is critical that both the system and storage on which the database resides, and the network that the queries will traverse, are extremely high-performance.

03 The vector database



Impact on the network fabric

In small deployments, the vector database can be stored locally on the node that will be processing the inference request. However, in larger deployments with multiple inference nodes, access to the vector database from all nodes simultaneously will be required. In these scenarios, it is generally desirable to store the vector database in an external shared storage system using protocols such as NFS, S3, etc., to facilitate access to the database.

From a network perspective, the critical factor in an at-scale RAG environment is latency to and from the storage system housing the vector database. Inference requests are real-time operations, and there is a delay during the time a RAG query embedding is created and the retrieval of close-proximity chunks from the vector database that are passed back to the inference server. If the RAG database queries are combined with the main frontend network traffic, undue network latency can slow down inference requests, potentially resulting in poor user experience.

A simple solution is to create a separate storage fabric for the RAG database I/O, but this approach requires additional hardware, cabling, etc., and is likely unnecessary in most deployment scenarios. A better solution in most use cases is to converge the vector database and inference traffic onto the same fabric, but to use VXLAN to separate the traffic.

The benefits of VAST Data storage

RAG inference performance depends not only on the network fabric but also on the storage used for the vector database, which is a critical component. VAST Data storage systems can be used for highly scalable vector databases such as Milvus or Pinecone, offering specialized solutions for RAG, including the VAST Database and VAST Datastore.

Ensuring that vector database lookups are performed as quickly as possible is critical, just like minimizing latency when transporting those lookups across the network fabric. While neither can make a RAG query complete faster, they can slow it down.

In our labs, we've tested RAG workflows with multiple vector databases using both local drives and VAST Data as the vector store. By isolating storage traffic into a dedicated VXLAN, cleanly managed by Apstra Data Center Director, and combined with the simplicity of the VAST Data storage environment, RAG vector database operations can be handled and managed with ease.

04 Solution use cases

RAG can be applied across a wide range of use cases, including enterprise, financial services, healthcare, industrial, and more. As previously discussed, RAG enables an LLM to access localized data without requiring model fine-tuning. While this reduces complexity, eliminating the need for fine-tuning the model for example, it adds complexity at the infrastructure level due to the need for a vector database to store the vector embeddings.

Beyond the benefits of RAG that we have discussed, deploying RAG with Juniper switches and VAST Data storage provides the following advantages:

- **"One-click" deployment** of the network fabric using Apstra Data Center Director, with the flexibility to support either separate frontend and storage fabrics or operate a single fabric in a converged capacity while separating inference query traffic from vector database traffic using VXLAN tunnels
- **Maximized throughput** from VAST Data across the Juniper fabric by leveraging Juniper QFX switches. The VAST Data 2x1 system can deliver over 50GB/second with 500,000 IOPS, and the Juniper QFX fabric can scale accordingly. With 16x100G upstream connectivity from VAST Data, the Juniper QFX5130 provides a 100G/400G fabric

Example use cases

1. Engineering document repositories
2. Customer service chatbots
3. Financial services

05 Architecture used

The solution utilizes the following components:

Hardware

- SuperMicro X13 system with dual Nvidia RTX 5000 ADA GPUs
  - VectorDB manager, ingest, and vectorizer
- SuperMicro AMD MI300X Instinct with eight GPUs
  - vLLM Inference Server for query and response
- VAST Data 2x1 with 480TB logical capacity / 240TB usable and 488,000 IOPS (read)
  - Storage repository for vector database
- Juniper Networks QFX5130 and QFX5230 400G switch
  - Used for leaf and spine switches connecting to the frontend fabrics
- Juniper Networks QFX5220 400G switch
  - Used as leaf and spine switches for the storage fabric

RAG software workflow

- LangChain for document preprocessing and chunking
- Milvus Vector Database
- Snowflake Arctic 2 embedding model
- Google Gemma 2 9b
- vLLM Inference Engine
- Ubuntu 24.04 LTS

06 Solution design

Using Data Center Director to seamlessly and efficiently build and scale RAG inference servers

Apstra Data Center Director transforms this paradigm by reducing complex network fabric deployment to just a few clicks. The intent-based networking automation solution eliminates the complexity typically associated with building and scaling network fabrics for RAG workloads. What once required weeks of planning, configuration development, and careful coordination between network and application teams can now be accomplished in minutes through Data Center Director's intuitive interface.

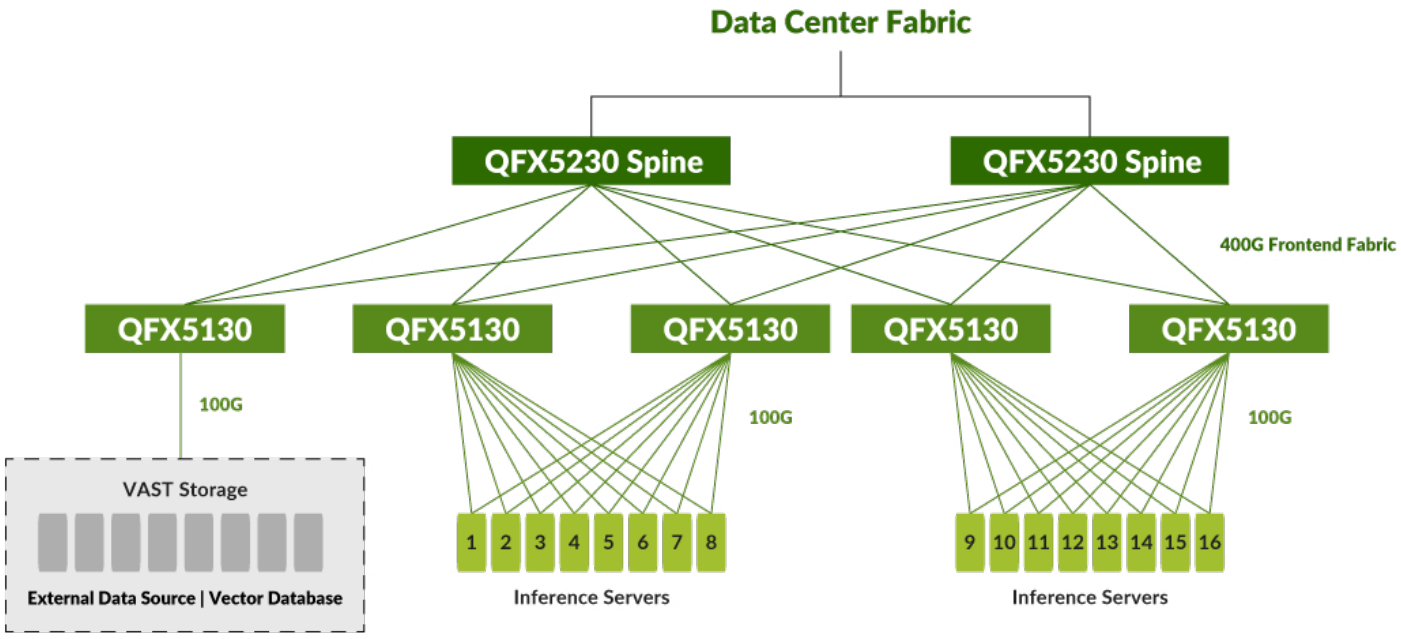


FIGURE 1

Optimizing performance with RAG architecture in the data center

Creating new infrastructure for RAG deployments begins with Data Center Director's visual rack design approach, as demonstrated in the interface above. The screenshot shows how administrators can design complete storage racks for vector databases in minutes using the intuitive drag-and-drop rack builder. The logical device configuration panel displays simple requirements like "20 ports, 10Gig" while the ESI redundancy protocol dropdown shows how built-in redundancy protocols are automatically configured for dual-homed connectivity. The visual representation clearly shows dual-homed VAST storage servers connected to leaf switches, illustrating how even small teams without deep networking expertise can model complete rack topologies before any physical deployment occurs. Rather than wrestling with complex configuration syntax or requiring weeks of planning, network administrators simply drag and drop components to define their requirements and let Data Center Director handle the underlying technical complexity of EVPN ESI configuration.

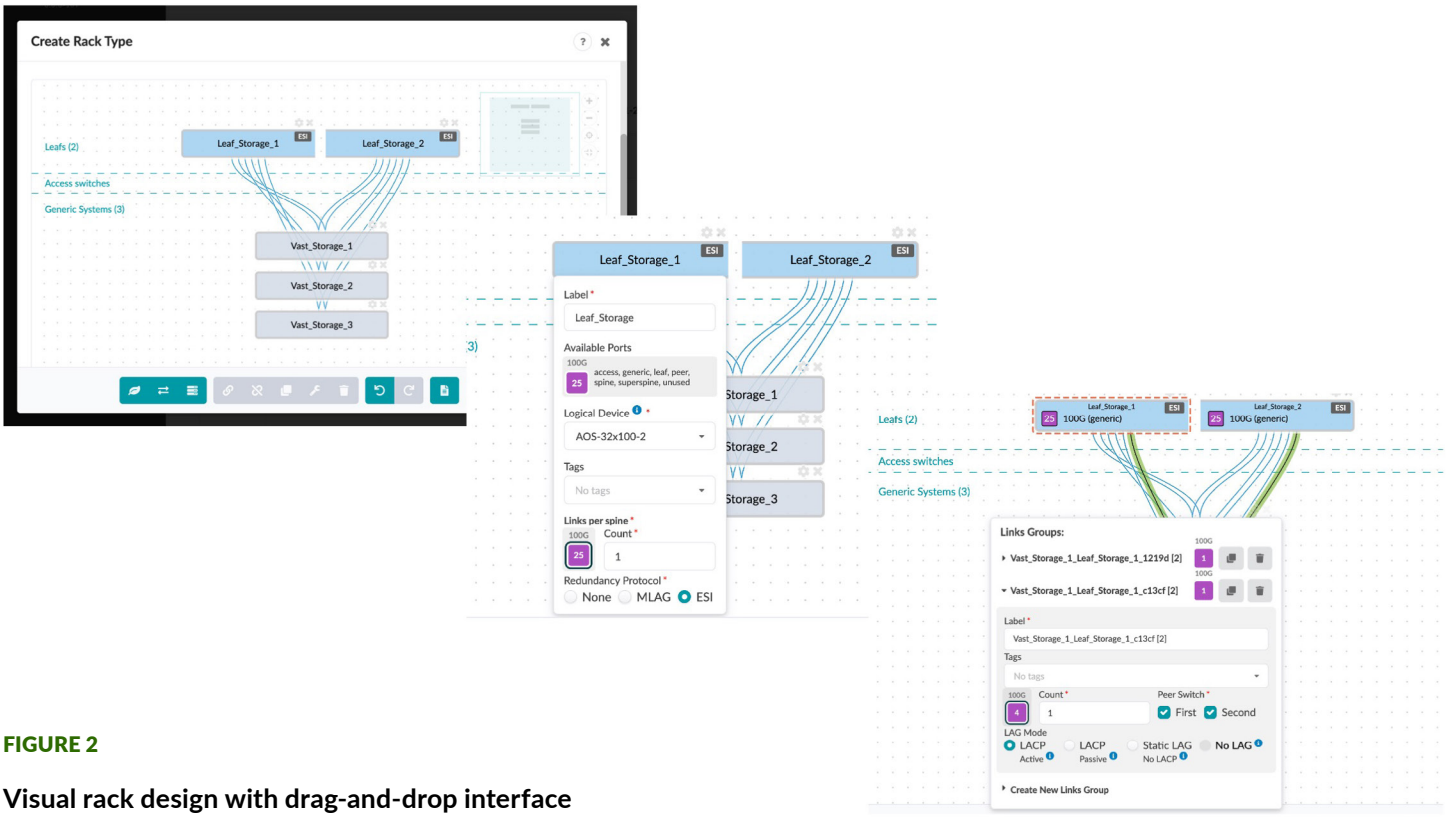


FIGURE 2

Visual rack design with drag-and-drop interface

Expanding RAG infrastructure demonstrates Data Center Director's true operational advantage: enabling deployment in just a few clicks. The screenshot illustrates how the "add rack" dialogue allows administrators to select from prevalidated rack designs, with Data Center Director automatically validating that the requested expansion is feasible within the existing fabric constraints. The interface showcases the platform's intelligent capacity planning by indicating exactly how many additional racks the current spine infrastructure can support—in this case, up to eight racks, based on available spine port capacity. This means that even a single network administrator can scale RAG infrastructure rapidly, without complex calculations, coordination meetings, or risk of deployment errors. The rack selection interface displays the new VAST storage rack template ready for deployment, turning what traditionally requires weeks of planning and coordination into a process completed in minutes.

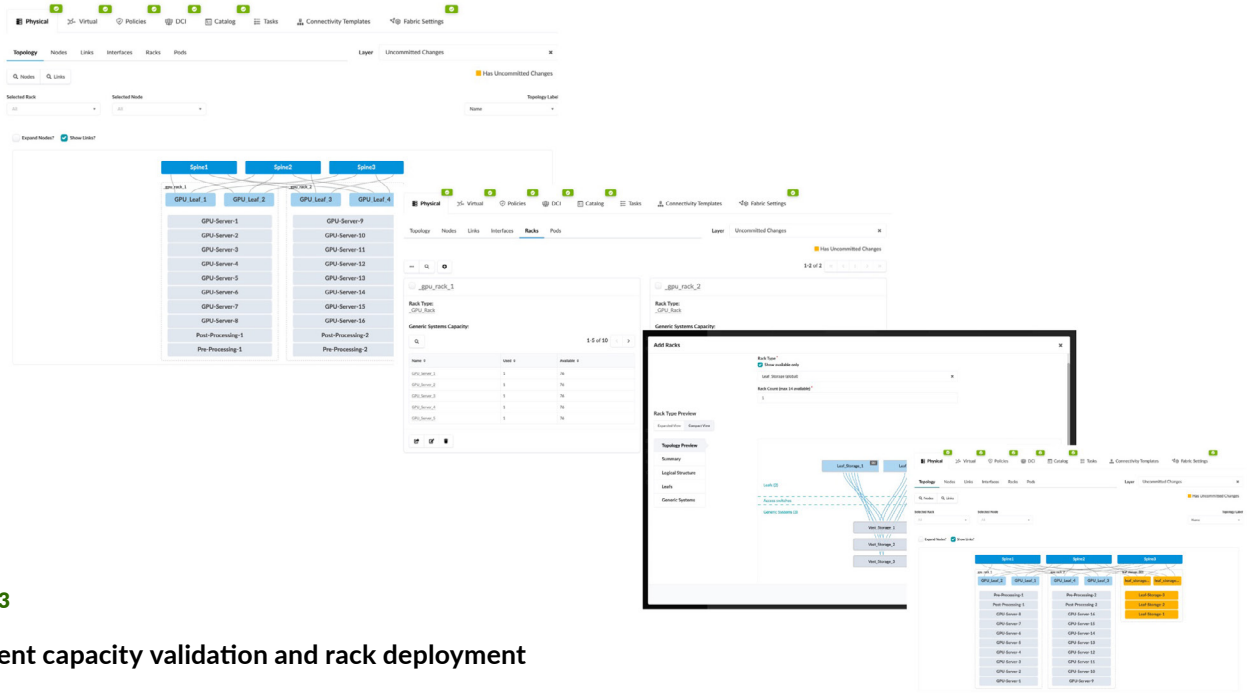


FIGURE 3

Intelligent capacity validation and rack deployment

Network segmentation for RAG workloads becomes equally straightforward, requiring just a few clicks to create dedicated VXLAN tunnels that isolate vector database traffic from inference queries. The screenshot shows the virtual network creation screen for “RAG Storage Net,” where administrators can specify high-level requirements while Data Center Director automatically manages the complex underlying implementation. The interface shows VRF selection, automatic subnet allocation (/22), and tagged connectivity template configuration, with VXLAN IDs, VLAN tags, and IP addresses automatically assigned from predefined resource pools. This enables small teams to deploy sophisticated network segmentation without the manual coordination typically required between network and application teams while ensuring optimal traffic isolation for RAG performance through dedicated pathways for vector database queries.

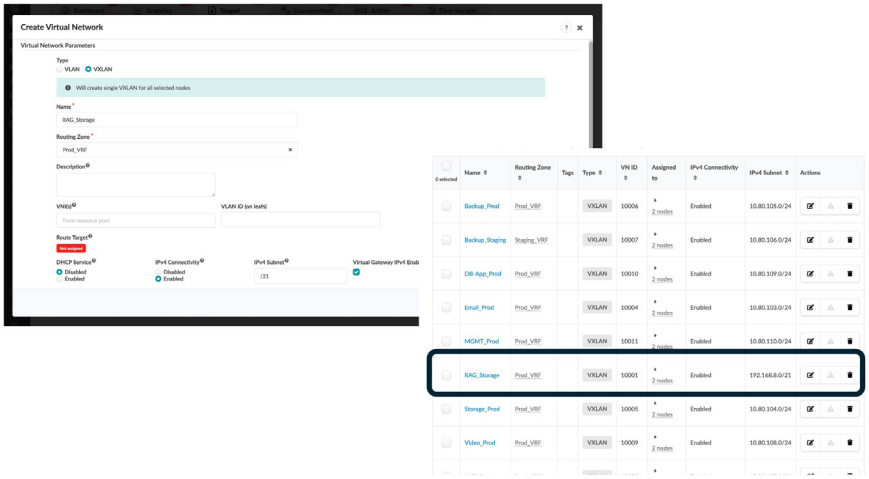


FIGURE 4

Automated VXLAN overlay creation and resource allocation

Behind this simplified interface lies sophisticated automation that generates hundreds of lines of optimized configuration in seconds, as shown in the configuration output above. The screenshot displays complex EVPN-VXLAN syntax, BGP configuration, interface settings, and ESI-LAG parameters that Data Center Director generates automatically from the simple interface interactions shown in previous figures. This represents over 300 lines of configuration for a single switch that would traditionally require deep networking expertise and weeks of testing time.

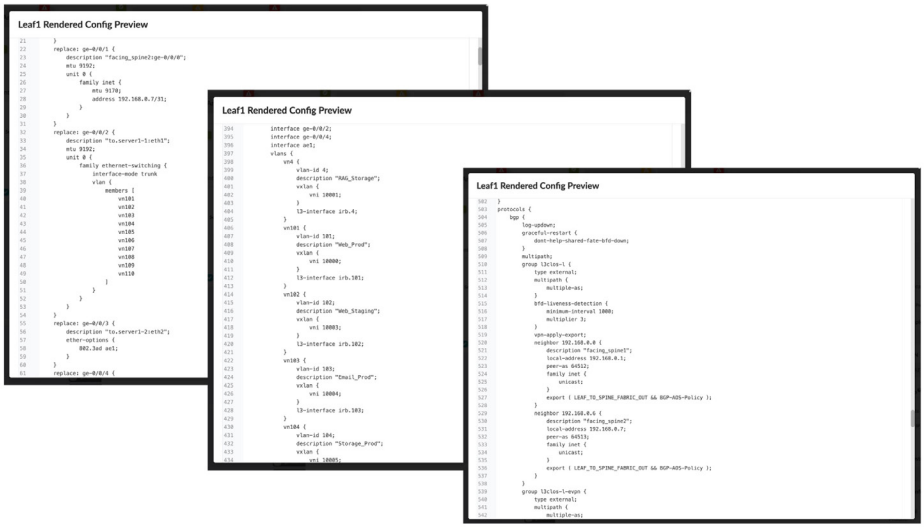


FIGURE 5

Auto-generated EVPN-VXLAN configuration output

Instead, small teams can deploy complex EVPN-VXLAN fabrics with just a few clicks while the platform automatically creates EVPN route targets, configures BGP policies across the spine-leaf topology, and establishes ESI-LAG parameters for storage redundancy. What once required extensive coordination and manual effort becomes a transparent, automated process that ensures consistency and adherence to best practices, eliminating the configuration drift that often plagues complex network environments.

Beyond initial deployment, Data Center Director delivers significant operational advantages through the RAG infrastructure life cycle. The platform's single source of truth approach ensures that, regardless of which team member implements changes, the outcome remains consistent and repeatable. This repeatability breeds reliability, eliminating human error and preserving configuration integrity over time.

Small teams can operate sophisticated RAG network fabrics with confidence because Data Center Director continuously monitors both the intended network state and the actual deployed configuration. When changes occur, whether planned or unexpected, the platform immediately identifies discrepancies between design intent and reality. This capability proves invaluable for RAG deployments, where network performance directly impacts inference quality and user experience.

This comprehensive lifecycle management transforms the network from a potential bottleneck into a strategic enabler of rapid RAG innovation, allowing organizations to focus resources on optimizing their AI workloads rather than managing the complexities of the underlying network infrastructure.

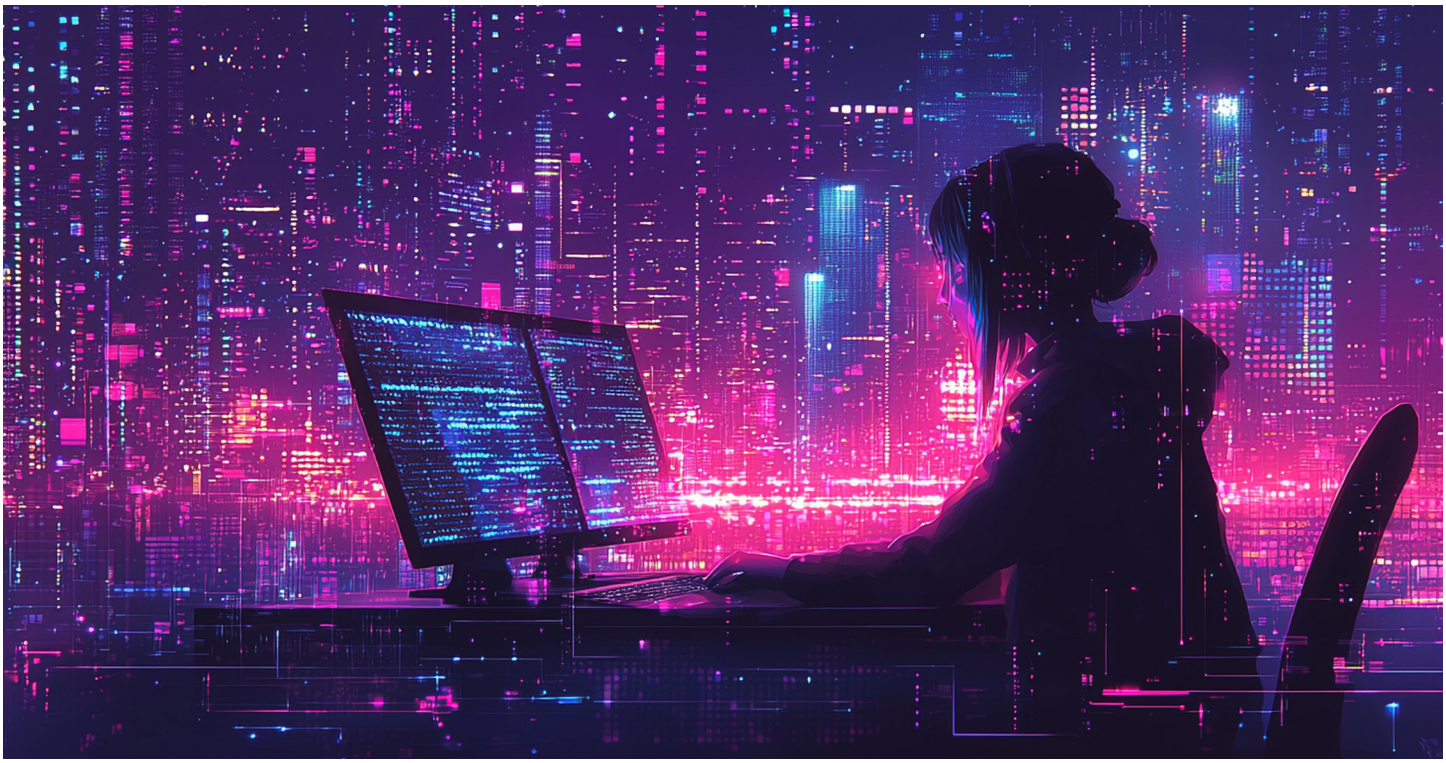
07 Conclusion

The benefits to an organization implementing a RAG architecture as part of its production AI deployment are clear: relevant access to localized data, enhanced data privacy, reduced hallucination risk, and a better overall user experience.

While the RAG-based deployments can be complex, the network component of that deployment doesn't have to be difficult or time-consuming to set up and implement. Apstra Data Center Director simplifies the process for network administrators in creating and deploying fabrics for RAG inference architectures, with the flexibility to scale as needed. Juniper QFX switches, including the QFX5130, are ideally suited for frontend inference fabrics. Additionally, with their larger buffer size, these switches excel at handling low-latency RAG-based vector database storage traffic.

It's also worth noting that, as the storage deployments for RAG clusters become more demanding and the proliferation of these clusters increases, there is a use case for distributed storage across wide area networks utilizing Data Center Interconnect (DCI).

As AI adoption grows in production environments and scales across enterprises, RAG inference and (soon) distributed inference clusters leveraging RAG will become a critical part of any company's AI strategy. Juniper Networks is well positioned to meet the networking demands of that evolution.



Why Juniper

HPE Juniper Networking is leading the convergence of AI and networking. Mist™ Juniper's AI-native networking platform, is purpose-built to run AI workloads and simplify IT operations, assuring exceptional and secure user and application experiences—from the edge to the data center to the cloud. Additional information can be found at www.juniper.net), X, LinkedIn, and Facebook.

JUNIPER NETWORKS

Take the next step

See our AI data center solution in action

Visit the Ops4AI Lab →

Explore our AI networking capabilities

Discover solutions →

Explore demos

Discover more →

Get insights

Subscribe to The Feed →

www.juniper.net

Juniper Networks Inc.
1133 Innovation Way
Sunnyvale, CA 94089

2000838-001 EN August 2025

JUNIPER NETWORKS