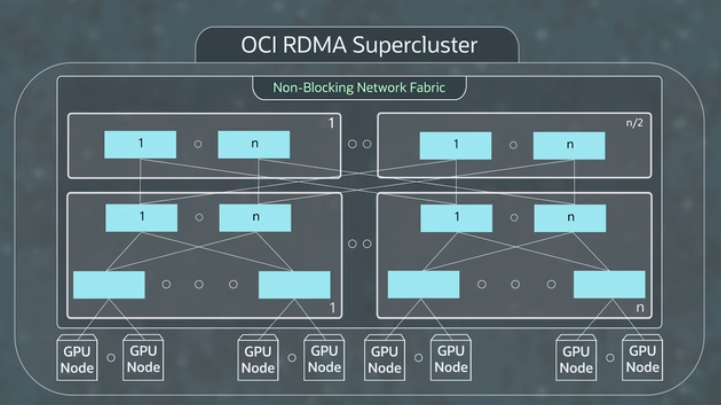


The **Supercluster** being described consists of powerful GPU nodes that are connected in a way to maximize performance. Let’s break it down:

1. **GPU Nodes**: At the bottom of the schematic, there are two **GPU nodes**. Each of these nodes contains **eight NVIDIA A100 GPUs**. These GPUs are **interconnected using NVIDIA’s NVLinks**, which means they can communicate quickly with each other to share data. This setup helps the GPUs work together more efficiently.
2. **Network Fabric**: The GPU nodes are then connected to a larger system called the **network fabric**. You can think of the network fabric as a super-fast highway that connects all the GPUs together. It allows all the GPUs to communicate without any slowdowns, so none of the GPUs have to wait for data.
3. **High-Speed Connection**: Each GPU node connects to the network fabric at a speed of **1.6 terabits per second** (or 1,600 gigabits per second), which is extremely fast. This ensures that all the GPUs in the cluster can exchange data quickly, maximizing performance when running complex AI models or high-performance computing tasks.

In simple terms, the **Supercluster** consists of multiple groups of powerful GPUs that are connected together with a very fast network, making sure that everything works smoothly and quickly when handling big AI or computing tasks.

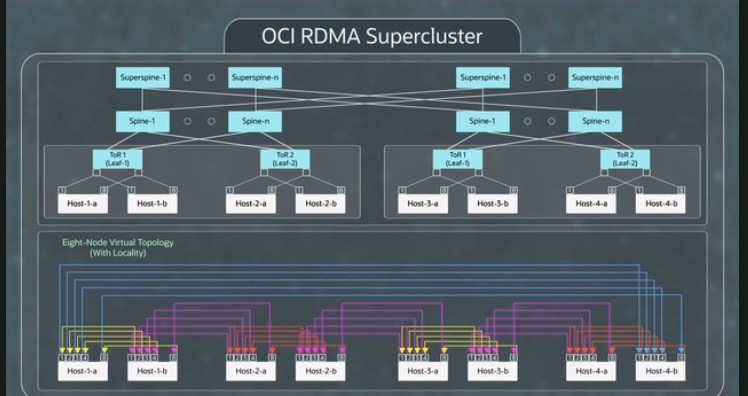
Top of Form

llllllkkkkk

1. **GPU Bandwidth and Network Fabric**:
   * In this system, each GPU gets **200 gigabits per second** of network bandwidth. The total network bandwidth is **1.6 terabits per second** (1,600 gigabits), and this bandwidth is divided across the GPUs.
   * The **network fabric** connects all GPUs in a way that any GPU can communicate with any other GPU at the same time without delay. This is called **nonblocking**, meaning no data bottlenecks.
2. **Supercluster and Network Design**:
   * A **Supercluster** is a very large network of GPUs—much bigger than a typical cluster. It’s designed to handle **tens of thousands of GPUs** and can potentially scale to **over 100,000 GPUs**.
   * The network is organized in blocks, and each block has a special three-layer network design called a **Clo network**. This design ensures that all GPUs can communicate efficiently, even at such a large scale.
3. **Managing Latency**:
   * When a GPU in one block wants to communicate with a GPU in a faraway block, there’s a slight delay in communication, known as **latency**. This is because the network is so large and the GPUs are physically spread out. The worst-case delay is about **20 microseconds**, which is still very fast.
   * To minimize issues caused by latency, the system uses **quality of service (QoS)** features. These include intelligent switches and extra memory buffers to ensure smooth data flow, preventing any dropped data (a "lossless" network).
4. **Handling Low-Latency Workloads**:
   * While the **20-microsecond** latency is very low compared to other cloud networks, some applications (like **high-performance computing (HPC)** or **databases**) need even faster communication—around **6-7 microseconds**.
   * For such cases, the system has a **control plane** that automatically assigns these workloads to a smaller, faster block of GPUs, ensuring they experience the lower latency required.

**Summary:**

The **Supercluster** is a very large, high-speed network of GPUs designed to scale up to tens of thousands of GPUs. It ensures fast, nonblocking communication between GPUs, but in very large setups, there can be a slight delay (latency). The system uses special techniques to minimize this delay and even has options to assign certain workloads to smaller, faster blocks for lower-latency performance, ensuring everything runs smoothly regardless of scale.



 **Managing Latency in Large GPU Workloads**:

* When GPU workloads span across different "blocks" in a Supercluster, they may sometimes experience a **higher latency** (around 20 microseconds). However, this doesn’t happen all the time. To minimize this, the system uses something called **network locality hints**.

 **What Are Network Locality Hints?**:

* These hints help customers know how close their GPUs are to each other within the network. With this information, customers can arrange their GPUs in a way that **keeps most of the traffic local**, meaning most of the data stays within a block, avoiding unnecessary travel across blocks. This results in lower latency for most of the communication between GPUs.

 **How Does This Work?**:

* For example, if you have GPUs in **two different blocks**, the system is smart enough to organize traffic so that about **85% of it stays local** to one block, minimizing latency.
* Occasionally, some data does need to travel between blocks, which can have higher latency (up to 20 microseconds), but this happens less frequently.
* On average, most data transfers happen very quickly, with some traffic even staying below **6.5 microseconds** in latency when it stays local to the top of the rack (the immediate network connection).

 **Benefits of Localized Traffic**:

* Keeping traffic local not only reduces latency but also helps prevent **flow collisions**, which are like traffic jams on a network when multiple data streams try to use the same connection. By keeping data within a block, these collisions are less likely to happen, which increases overall **throughput** (the speed at which data moves across the network).

 **Optimizations in the Supercluster**:

* The Supercluster network is designed with several smart optimizations:
  1. **Buffers**: The system uses special memory buffers that are fine-tuned to handle the network’s latency, ensuring that data is never lost (this is called a **lossless network**).
  2. **Placement Mechanism**: When GPU nodes or other resources are launched, the system tries to place them as close as possible to each other, reducing latency and avoiding flow collisions.
  3. **Placement Hints**: For workloads that need to span multiple blocks, **placement hints** help algorithms keep traffic as local as possible, minimizing the time data spends traveling across the network.

Bottom of Form