Google Skills

# Reviewing the core TPU concepts

## What are cloud TPUs?

As Don mentioned, Tensor Processing Units are custom-designed application-specific integrated circuits (ASICs) built by Google to accelerate machine learning computations. Code written for TPUs is compiled by the Accelerator Linear Algebra (XLA) compiler, which translates the computational graph into machine code optimized for TPUs.

**How it works:** XLA acts as a just-in-time compiler, meaning it takes the computational graph generated by a machine learning framework and translates the linear algebra operations, loss functions, and gradient calculations into machine code specifically for the TPU. The remaining parts of your program will run on the TPU host machine. The XLA compiler itself is included within the TPU VM image that operates on this host machine. You'll learn more about this in the upcoming lessons.

**Want to learn more** about Tensor Processing Units? See how to think about TPUs.

## When to use TPUs

Cloud TPUs are ideal for computationally intensive tasks, including:

- **Training** large, complex deep learning models like large language models (LLMs).
- **Inference** workloads that rely heavily on embeddings, such as recommendation systems, which benefit from the specialized SparseCores. Ironwood TPUs, for example, were purpose-built for inference workloads. Ironwood is our most powerful, capable and energy efficient TPU yet, purpose-built for power thinking, inferential AI models at scale. It supports this next phase of generative AI and its tremendous computational and communication requirements, and can scale up to 9,216 liquid cooled chips linked with breakthrough Inter-Chip Interconnect (ICI) networking spanning nearly 10 MW.

Don's video didn't describe all TPU generations and versions - that decision will vary based on the nature of your workload. As an introductory course, we won't deep dive into deployment, but here's a breakdown of the available options:

# At a glance: TPU across generations

| Feature | v5e | v5p | Trillium (vs v5e) | Ironwood (vs v5p) |
|---|---|---|---|---|
| Number of chips per pod | 256 | 8960 | 256 | 9216 [TPU7x], 256 [TPU7] |
| Chip Bf16 TFLOPs | 197 | 459 | 918 (4.7x) | 2307 (5x) |
| Chip Int8 or FP8 TOPs | 394 | 918 | 1836 (4.7x) | 4614 (10x, fp8) |
| HBM/chip (GB) | 16 | 96 | 32 (2x) | 192 (2x) |
| HBM BW (GB/s) | 820 | 2765 | 1640 (2x) | 7380 (2.7x) |
| Host DRAM (GB) | 512 | 512 | 1536 | 1152 |
| ICI BW per chip (GB/s) | 400 bi-dir | 1200 bi-dir | 800 bi-dir (2x) | 1200 bi-dir |
| DCN BW per chip (Gb/s) | 25 | 50 | 100 (4x) | 100 (TPU7x), 200 (TPU7) |
| SparseCore | No | Yes | Yes (New) | Yes |

## Cloud TPUs can offer distinct benefits for inference, including:

- Great performance and cost-efficiency for demanding AI workloads
- Support for major AI frameworks, including PyTorch and JAX
- The ease to switch from GPUs to TPUs for inference, or use both in a multi-accelerator deployment - all with just a few configuration changes without major code rewrites. We'll explore this in more detail later

# A recap: TPU system architecture

TPUs are specifically engineered with four key components to accelerate machine learning workloads:

- **TensorCore:** This is the primary processing unit, handling the bulk of the computational acceleration
- **BarnaCore/SparseCore:** This specializes in sparse computations to significantly speed up specific deep learning tasks
- **High Bandwidth Memory (HBM) access interface:** This provides fast access to memory for the TPU
- **Inter-chip-interconnect (ICI):** A proprietary Google technology, ICI enables seamless communication between multiple TPU chips

To handle large-scale ML operations, Google developed custom machines that house multiple TPUs, allowing for massive increases in processing power.

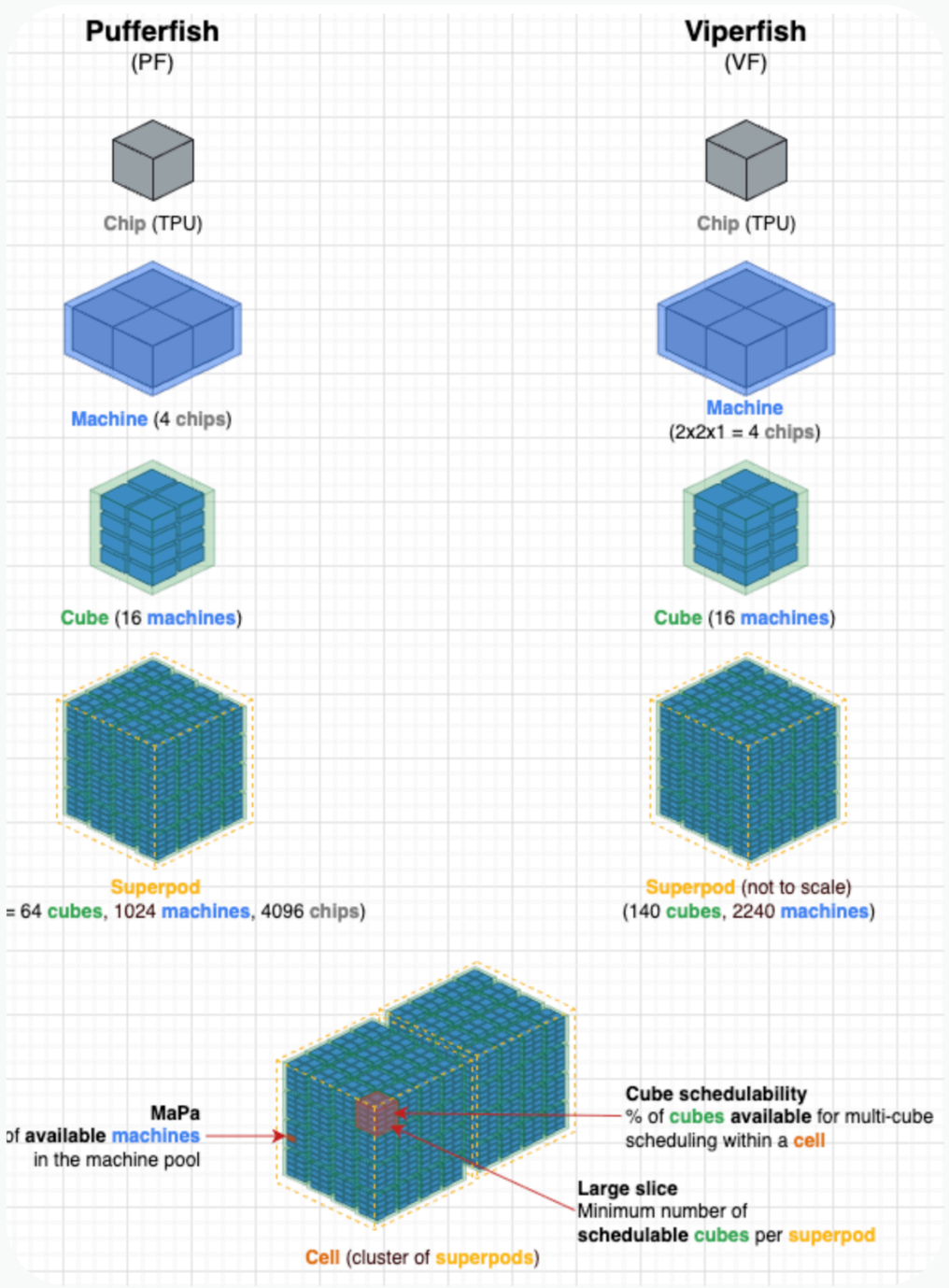
## What's inside a TPU chip?

Each chip contains one or more TensorCores, which are comprised of:

- **Matrix-multiply units (MXUs):** These are the computational cores, structured as systolic arrays, that perform thousands of multiply-accumulate operations per cycle
- **Vector units:** These handle general computations like activation functions
- **Scalar units:** These manage control flow and other housekeeping tasks

To scale up, multiple TPUs are grouped into a **TPU pod**—a collection of TPUs connected bya high-speed network.

While a single TPU chip is powerful, real-world AI models often require even more computing power. This is where **TPU pods** and **slices** come in.



## What is a slice?

A **slice** is a subset of chips within a single pod, connected by extremely fast Inter-Chip Interconnects (ICI).

Slices are described in terms of either the number of chips or the number of TensorCores, depending on the TPU version. You might hear terms like **"chip shape"** and **"chip topology"** which refer to how chips within a slice are arranged and interconnected.

## Specialized components

- **TPU cube:** A 4x4x4 topology of interconnected chips introduced with TPU v4 to optimize communication
- **SparseCores:** Specialized processors included in TPUs that accelerate computations involving sparse data commonly found in recommendation models
- **ICI Resiliency:** A feature that improves the fault tolerance and scheduling availability of TPU slices by rerouting traffic around network faults. It is enabled by default for v4 and v5p slices of one cube or larger

## TPU cloud architecture

TPUs are made available on Google Cloud as compute resources, primarily through **TPU VMs.** You can manage these VMs directly or use them through managed services like Google Kubernetes Engine (GKE) and Vertex AI. (Diagram 1)

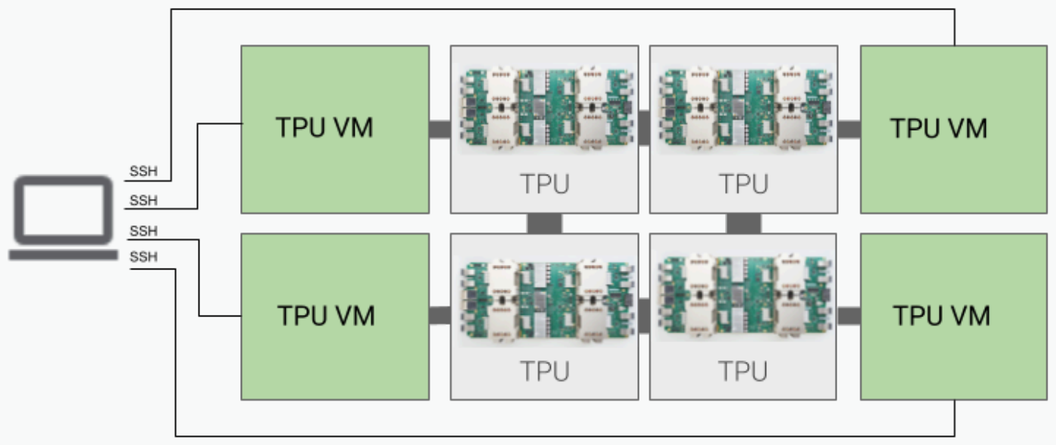



Diagram 1

A **TPU host** is the VM that runs on a physical machine with attached TPU hardware. Workloads can be configured in several ways:

- **Single-host:** The entire job runs on a single TPU VM—suitable for smaller models and experiments
- **Multi-host:** The job is distributed across multiple TPU VMs, treating them as a single atomic unit—an essential configuration for large-scale inference workloads. The following diagram shows a v5litepod-16 (v5e) multi-host TPU slice on Google Kubernetes Engine. This TPU slice has four VMs, each with four TPU v5e chips connected via high-speed interconnects (ICI), and each TPU v5e chip has one TensorCore. (Diagram 2)
- **Sub-host:** The job uses only a portion of the available chips on a single TPU VM, allowing for cost optimization on smaller tasks

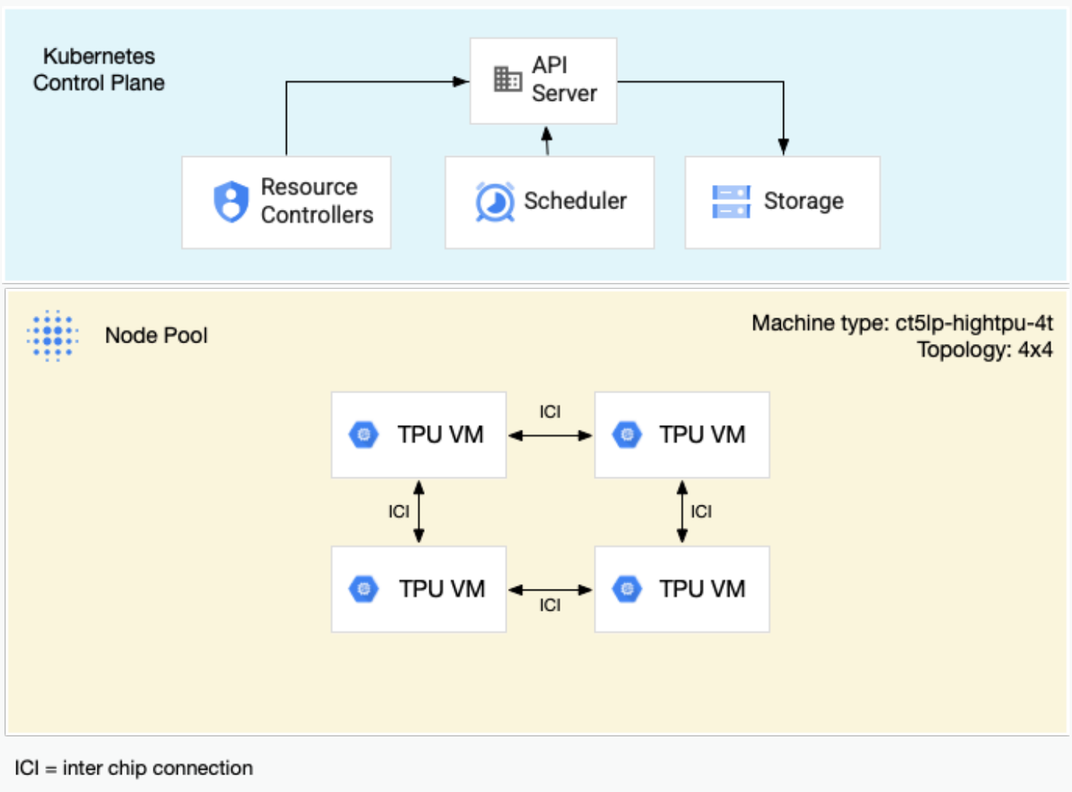



Diagram 2