Google Skills

Reviewing important inference concepts

Our last video, ‘Serving AI Models at Scale with vLLM,’ focused on vLLM but touched on two larger topics that require forethought when architecting your inference stack.

Novel techniques

Many emerging techniques have improved the performance and cost of inference. These include speculative decoding or low level kernel optimizations (like FlashAttention, PagedAttention, and FlashInfer) that improve hardware utilization. Another key approach is disaggregated serving, a technique for running large language models (LLMs) that separates the prefill and decode stages into different processes, potentially on different machines.

- **Prefill**: this stage processes the input prompt and generates an intermediate representation (like a key-value cache). It's often compute intensive.
- **Decode:** this stage generates the output tokens, one by one, using the prefill representation. It is typically memory-bandwidth bound.

By separating these stages, disaggregated serving allows for prefill and decode to run in parallel, improving throughput and latency while leading to improvements in performance and efficiency - particularly for large models.

Learn more: Though not included in our reference architecture, you can check out this recipe to see how to use disaggregated serving.

Accelerator lock-in

One of the most common fears we hear when someone is experimenting with TPUs and GPUs is “Is this a one-way door decision?” In other words, they’re worried about lock-in. The answer is no. vLLM is well known for being a fast and efficient library for inference. Earlier in 2025 we announced that you can now run inference on TPUs with vLLM in addition to existing support for GPUs. That means you have the flexibility to use TPUs or GPUs while effortlessly switching between them with just a few configuration changes.

This is another area where custom ComputeClasses in GKE can help you define what kind of virtual machine nodes and consumption models to use, and in what order of preference. For example, you could start with pre-reserved GPUs (Spot VMs), followed by GPUs on-demand, followed by TPUs on-demand, all in the **same deployment**, and GKE autoscaler takes care of the rest.

Learn more: vLLM Performance Tuning: The Ultimate Guide to xPU Inference Configuration.