Architecture Guide: Hyper-Converged KV-Cache Offloading

1. Introduction

This document details the deployment of a high-performance, distributed caching layer for vLLM using a single-node Kvrocks setup backed by NVMe SSDs. This architecture addresses the high cost of GPU HBM by offloading the KV-cache.

2. System Architecture

The system utilizes a Hyper-Converged infrastructure where the compute (vLLM) and storage (Kvrocks) reside on the same high-performance node to minimize network latency.

- **Compute:** vLLM Inference Engine.
- **Caching Layer:** LMCache (managing the eviction policies).
- **Storage Backend:** Kvrocks (Redis-compatible, running on RocksDB).
- **Physical Storage:** RAID0 NVMe SSD Array (for maximum IOPS).

3. Configuration Details

Storage Setup (Kvrocks)

We utilize Kvrocks in a single-node configuration optimized for SSD throughput. **Path:** /mnt/nvme_raid/kvrocks (Generic Path) **Port:** 6666

Connection Topology

Instead of a complex proxy setup, we use a direct connection optimized for local execution: vLLM (Localhost) -> LMCache Connector -> Kvrocks (Port 6666)

4. Performance Tuning

- **Chunk Size:** Configured to 1024 to balance hit-rate and throughput.
- **Threads:** Increased worker threads in Kvrocks to handle parallel I/O requests from vLLM.
- **Compaction:** RocksDB compaction tuned for write-heavy workloads typical of KV-cache offloading.