Performance Analysis

This report summarizes the performance results of a vLLM engine running on a distributed setup of four A10 GPUs. The analysis reveals that the system is stable and handles load balancing effectively,

Test Environment

The load test was conducted with the following configuration:

- The system consisted of 4 scheduler nodes, each running a dedicated vLLM replica.
- Each replica was deployed on an NVIDIA A10 GPU with 16GB of RAM.
- All requests targeted the TinyLlama model via the *v1/chat/tinyllama* endpoint.
- Client side metrics were captured using Locust .

The response was utmost 500 tokens per request.

Metrics and Results

System Load Distribution

- The total number of active requests handled by the system stabilized at approximately **950**, indicating a high concurrency workload

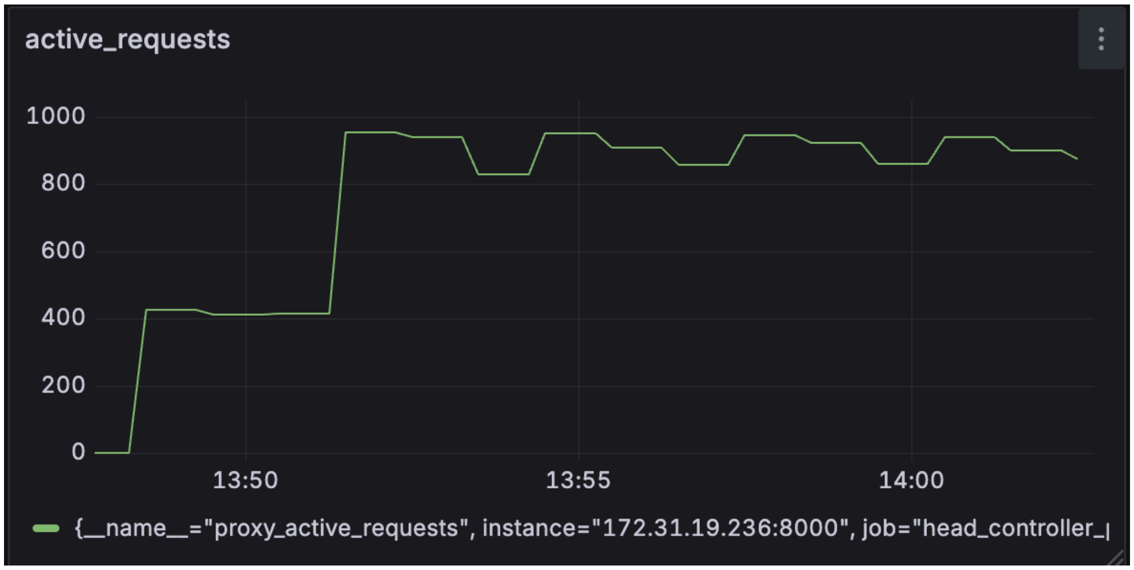

Fig1 : The number of active requests running in the system

- This load was distributed almost perfectly evenly across the four GPU replicas. Each replica consistently processed around **225-240 active requests**, which shows the least loaded load balancing was working effectively

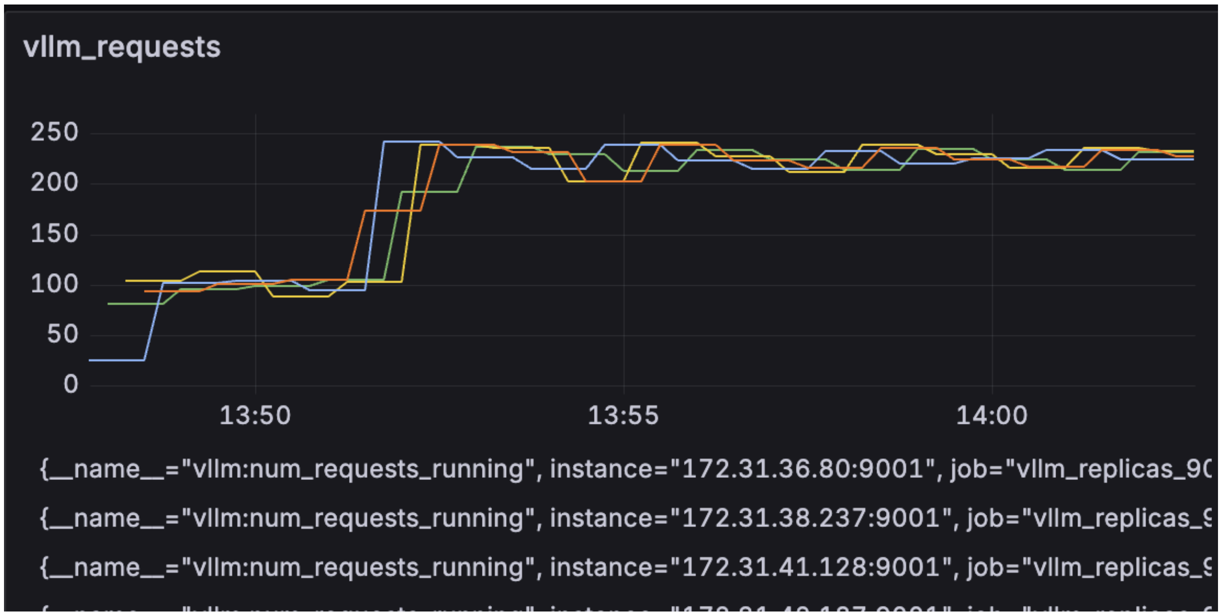

Fig2: Number of active requests in each GPU

- Under peak load, the GPU cache utilization settled between **50% and 60%**. This is a healthy level, showing that the GPUs had sufficient memory and were not overwhelmed by the demand

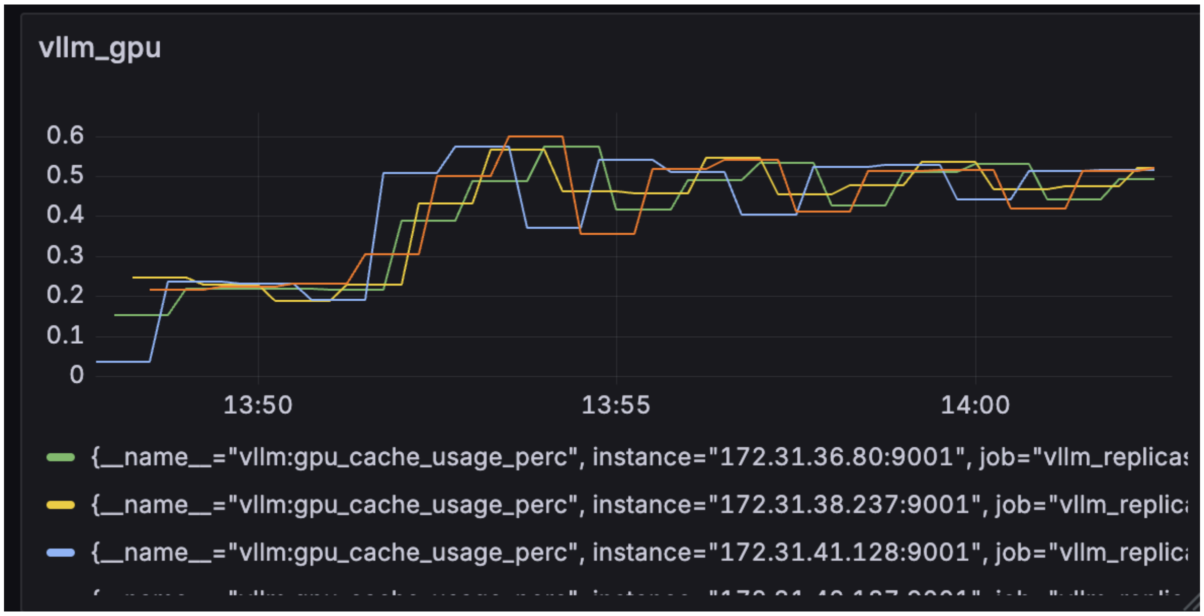

Fig 3: The percentage of GPU utilization

Latency Analysis

- The server-side performance was exceptionally fast and consistent. The **inter token latency**—the time to generate each new token—was stable at just under **0.1 seconds**. This confirms that the vLLM engine and the A10 GPUs are highly efficient. The latency was the same at the proxy showing an extremely robust backend without any queuing delays.

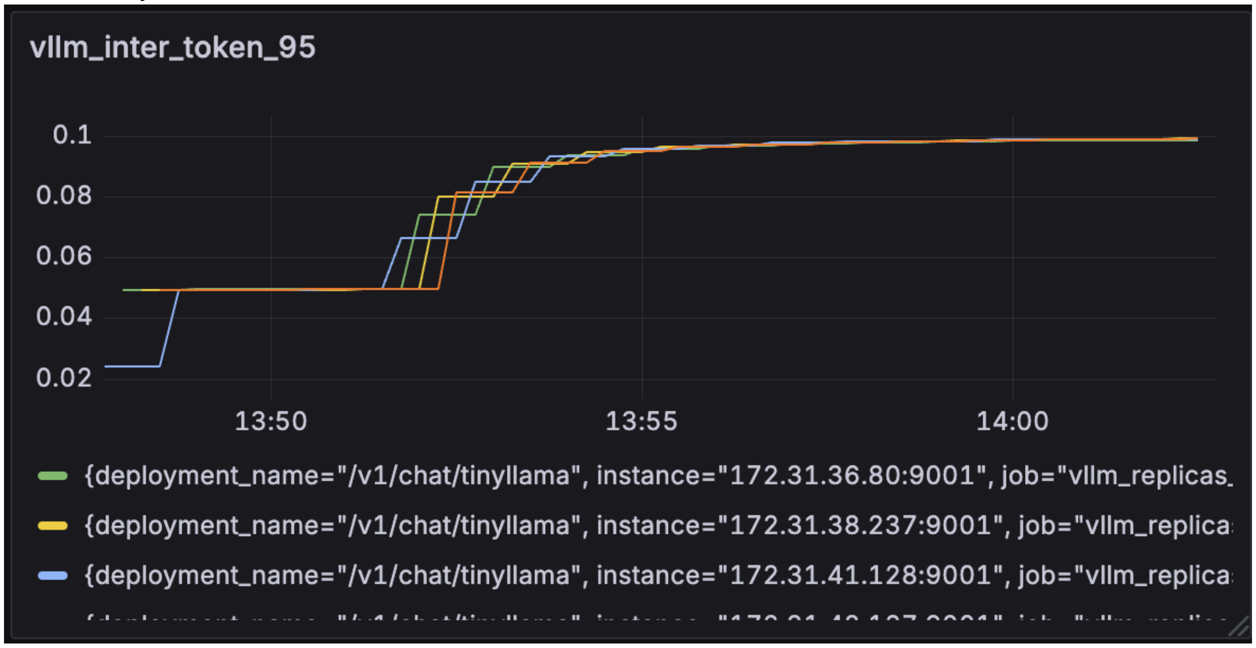

Fig 5: 95 percentile Inter token latency at vllm

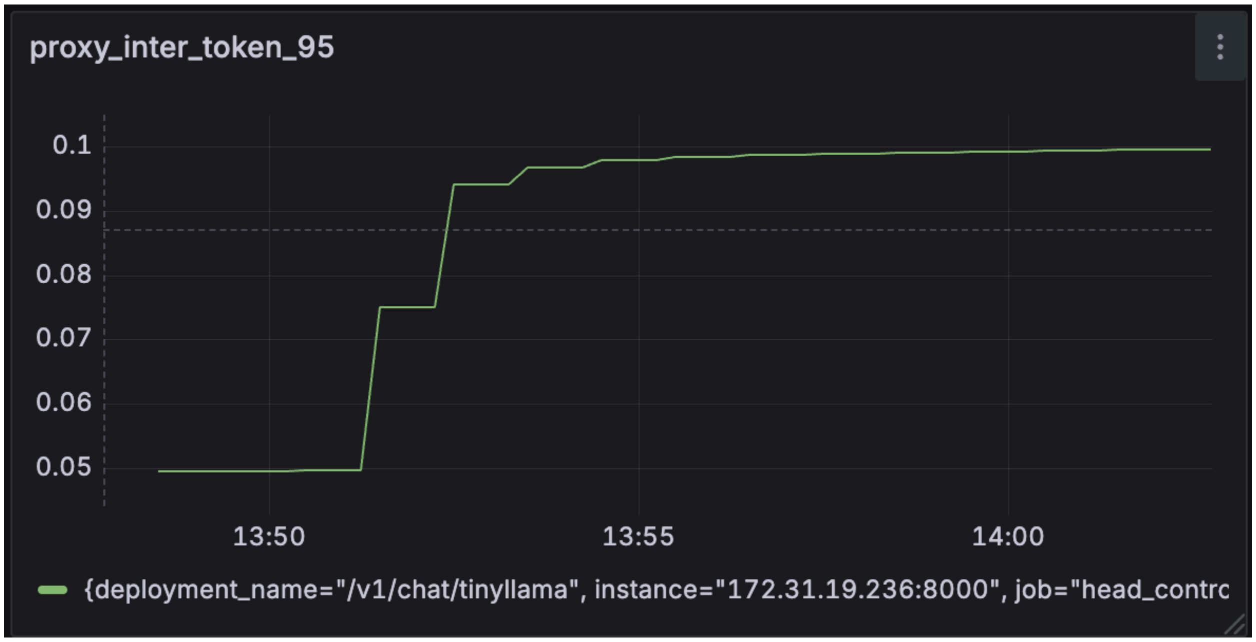

Fig 6: 95 percentile Inter token latency at proxy

- Both the system's internal metrics and the client-side Locust test show the 95th percentile latency was reaching **40-50 seconds.**

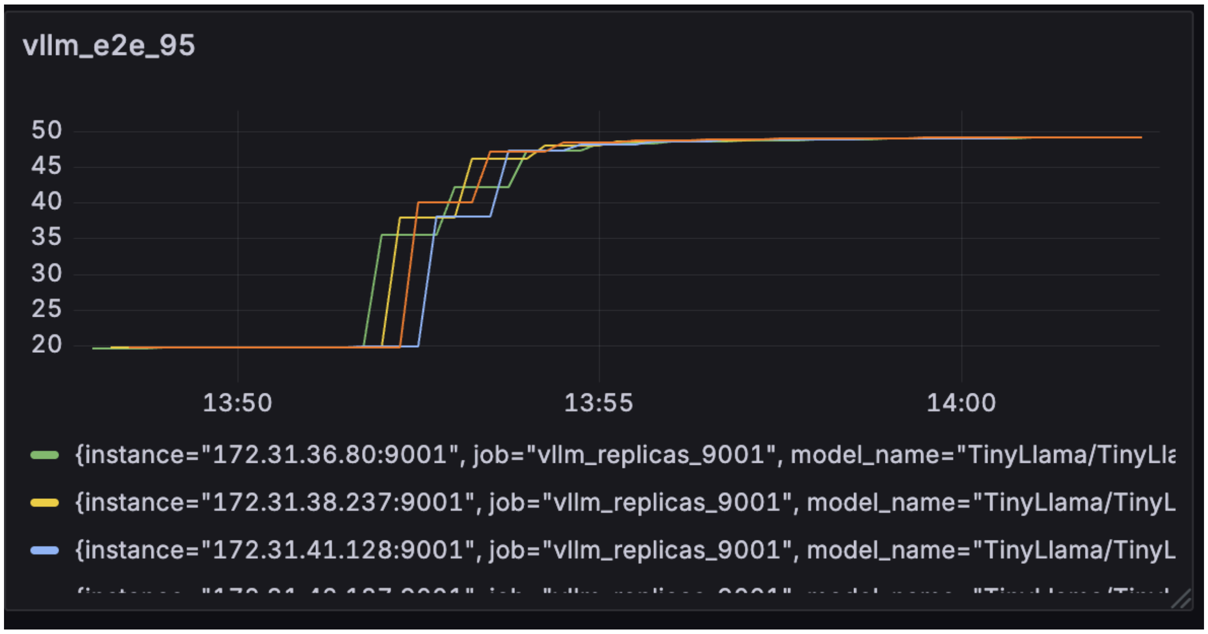

Figure 7: vllm end to end 95 percentile

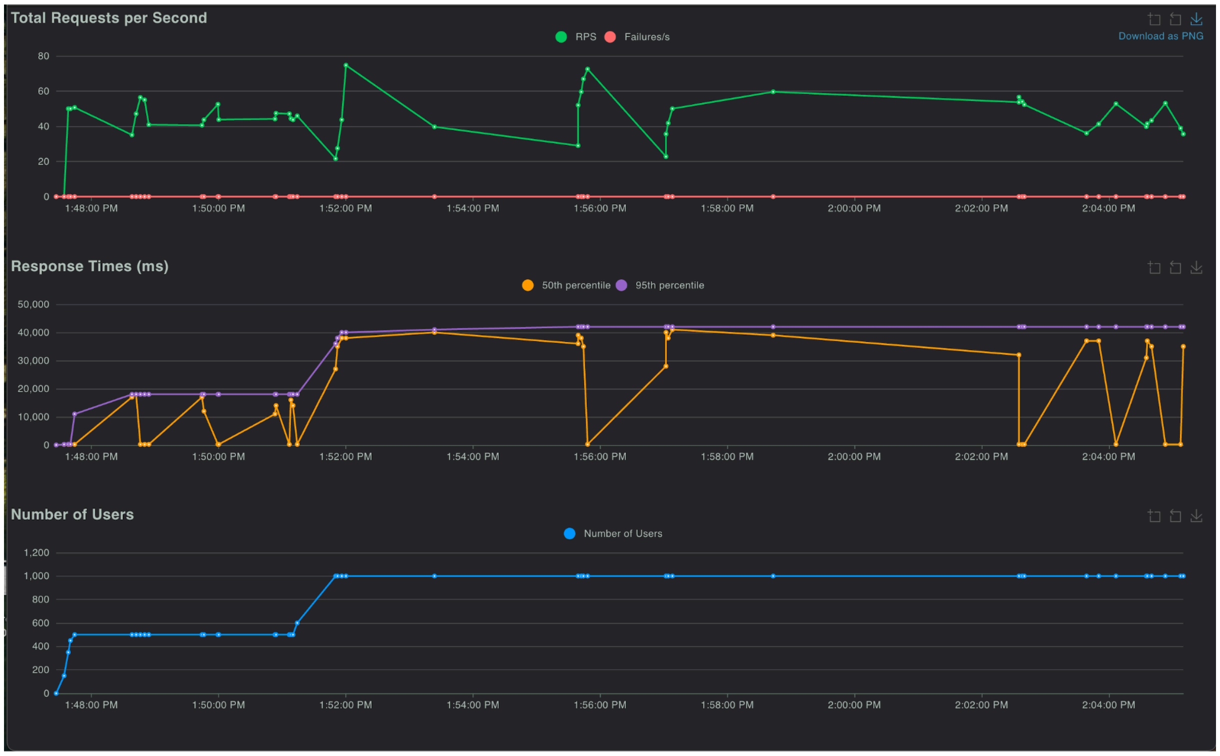

Fig 8: Locust test plotting end to end latencies

Conclusion

The system performance is in sync with performance of the vLLM engine, showing that the system has no queuing delays and the entire asynchronous implementation works well. The primary cause of the high latency is the long generation time required for each request.

While the GPU generates tokens very quickly (0.1 seconds per token), a request requiring 400-500 tokens takes a time of

450 tokens * 0.1 seconds/token = 45 seconds

This calculation almost perfectly matches the measured end-to-end latency.