FIT 5225 Assignment 1

You Ni

35358076

May 1st, 2026

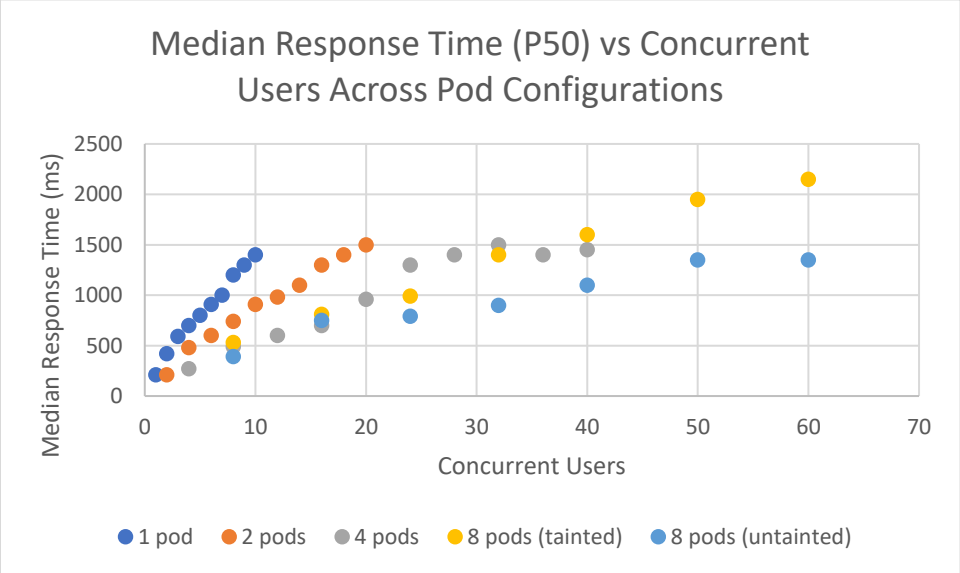
Empirical Benchmark Report

This study evaluates the performance and scalability of a FastAPI-based YOLO inference service deployed on Kubernetes. Experiments were conducted using Locust by increasing concurrent users across 1, 2, 4, and 8 pod configurations, with each pod limited to 1 vCPU. Two variants of the 8-pod setup were tested: worker-node-only (tainted control-plane) and mixed-node deployment (untainted control-plane).

Pods	Max Stable Users	Throughput (RPS)	P50 (ms)	P95 (ms)
1	10	5.95	1400	1500
2	20	10.99	1500	1900
4	40	20.32	1450	2200
8 (tainted)	32–40	19.2–20.92	1400–2150	2200–3350
8 (untainted)	60	31.03	1350	2100

Table 1: Summary of Benchmark Results

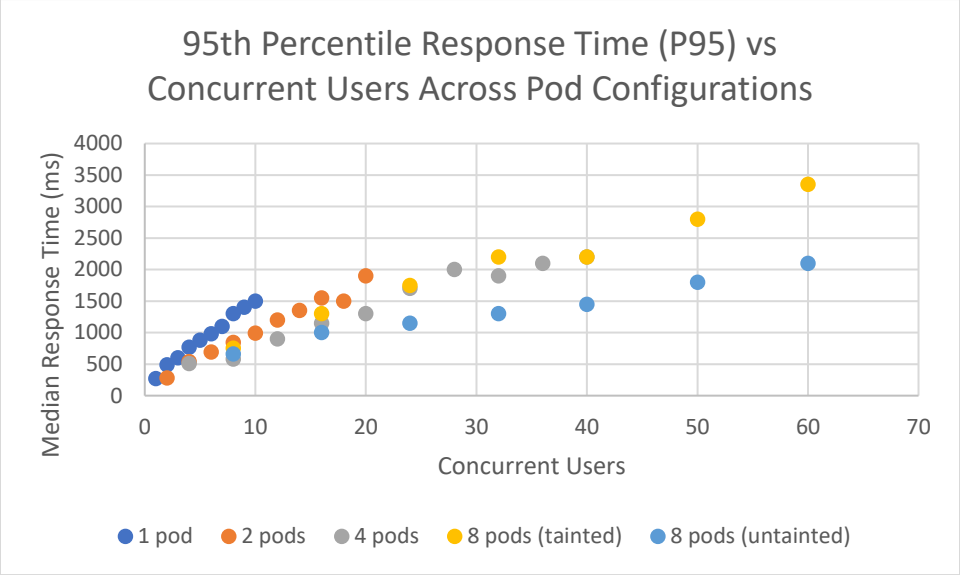
The results show that throughput increases with both concurrency and pod count, but with diminishing returns. A single pod achieved a maximum of approximately 5.95 RPS at 10 users, while 2 pods reached ~10.99 RPS at 20 users. The 4-pod configuration scaled to ~20.32 RPS at 40 users, demonstrating near-linear scaling. However, the 8-pod configuration revealed a critical distinction. When restricted to worker nodes, throughput plateaued at ~20.92 RPS, like the 4-pod setup, indicating resource constraints. When the control-plane node was made schedulable, throughput increased significantly to ~31.03 RPS, confirming that the limitation was due to insufficient cluster resources rather than application inefficiency.



(Median Response Time (P50) vs Concurrent Users)

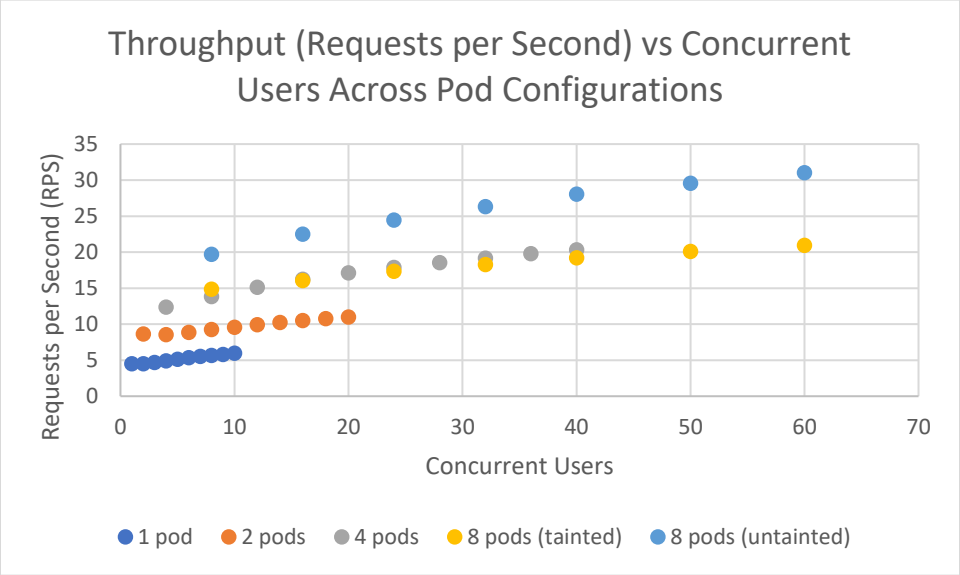
Median latency increases steadily with concurrency across all configurations. For example, in the 1-pod setup, P50 latency rises from approximately 210 ms to 1400 ms, while in the 4-pod

configuration it increases from ~270 ms to ~1450 ms. The 8-pod (untainted) configuration maintains comparatively lower latency at equivalent load levels, demonstrating improved load distribution.



(95th Percentile Response Time (P95) vs Concurrent Users)

Tail latency (P95) exhibits stronger growth, reaching up to 3350 ms in the 8-pod (tainted) configuration. Unlike systems that exhibit sharp latency spikes, the observed behaviour is characterised by gradual latency growth with moderate fluctuations. This indicates that the system enters a saturated but stable regime, where requests are queued rather than dropped.



(Throughput vs Concurrent Users)

The observed performance can be explained using queueing theory and Little's Law ($L = \lambda W$). As concurrency (L) increases, throughput (λ) initially scales with available CPU resources. However,

once the system approaches its service capacity, throughput stabilises while incoming requests continue to increase. Consequently, response time (W) must increase, resulting in the observed latency growth. This is further supported by the throughput plateau observed in higher concurrency levels, where additional users contribute primarily to queueing delay rather than increased processing.

The primary bottleneck is CPU-bound YOLO inference. Although FastAPI supports asynchronous request handling, inference execution dominates processing time and limits the service rate of each pod. Horizontal scaling mitigates this by increasing parallel processing capacity. However, scaling efficiency is constrained by cluster resource availability, as demonstrated by the limited performance of the 8-pod (tainted) configuration. Enabling scheduling on the control-plane node increased available resources and restored scaling efficiency.

In conclusion, the system demonstrates effective horizontal scaling up to available resource limits. While throughput improves with additional pods, latency increases under high concurrency due to queueing effects inherent in CPU-bound workloads. This highlights the trade-off between throughput and response time in cloud-based machine learning inference systems.

Appendix 1: Table (Raw Data):

Pods count	User	RPS (4min mark)	Failure	Response time (50th percentile) low	Response time (50th percentile) high	Response time (95th percentile) low	Response time (95th percentile) high	Fluctuation
1	1	4.49	0	210	210	270	280	none
1	2	4.49	0	420	480	480	500	low
1	3	4.68	0	590	600	600	610	none
1	4	4.9	0	700	710	770	780	low
1	5	5.12	0	800	810	880	890	low
1	6	5.33	0	910	920	980	990	high
1	7	5.5	0	1000	1100	1100	1100	low
1	8	5.67	0	1200	1200	1200	1500	high
1	9	5.81	0	1300	1300	1300	1700	high
1	10	5.95	0	1400	1400	1400	1800	medium
2	2	8.62	0	210	210	280	290	none
2	4	8.55	0	480	490	500	570	low
2	6	8.85	0	600	610	690	700	none
2	8	9.24	0	710	770	810	880	low
2	10	9.58	0	910	910	990	1000	low
2	12	9.93	0	980	990	1100	1300	high
2	14	10.24	0	1100	1200	1200	1500	high
2	16	10.49	0	1200	1300	1400	1700	high
2	18	10.75	0	1300	1500	1500	1600	high
2	20	10.99	0	1300	1600	1700	2100	high
4	4	12.38	0	270	280	510	520	none
4	8	13.82	0	490	500	580	590	none
4	12	15.11	0	600	600	880	920	none
4	16	16.21	0	700	710	1100	1200	some
4	20	17.12	0	790	1100	1300	1300	medium
4	24	17.86	0	1200	1300	1600	1700	medium
4	28	18.53	0	1400	1500	1800	2200	medium
4	32	19.15	0	1500	1700	1900	2300	medium
4	36	19.78	0	1200	1600	1900	2300	heavy
4	40	20.32	0	1100	1800	1900	2500	heavy
8 (tainted)	8	14.87	0	500	560	720	790	low
8 (tainted)	16	16.05	0	790	830	1200	1400	low
8 (tainted)	24	17.35	0	980	1000	1700	1800	low
8 (tainted)	32	18.29	0	1400	1500	2100	2300	low
8 (tainted)	40	19.2	0	1600	1700	2100	2300	low
8 (tainted)	50	20.11	0	1900	2000	2700	2900	medium
8 (tainted)	60	20.92	0	2100	2200	3100	3600	medium
8(untainted)	8	19.69	0	310	470	630	700	low
8(untainted)	16	22.5	0	700	800	680	1300	medium
8(untainted)	24	24.43	0	780	810	1100	1200	low
8(untainted)	32	26.31	0	890	920	1300	1400	low
8(untainted)	40	28.04	0	1000	1100	1400	1500	low
8(untainted)	50	29.56	0	1300	1400	1500	1900	medium
8(untainted)	60	31.03	0	1300	1400	2000	2300	medium

Appendix 2: Screenshot from Locust