Worker vs. Verifier Latency Benchmark Report

Single-request latency for worker generation against verifier prefill on Qwen3-8B and one RTX 4090: 48 input x output combinations, per-token decode cost, prefill throughput and the worker/verifier cost ratio.

This report is based on the timing results from `tools/vllm_worker_verify_pipeline/benchmark_worker_verifier_latency.py`. It quantifies the time-cost gap between the two paths and analyzes how input / output length affects that gap. Two runs are covered here — a smaller model on a consumer GPU and a larger model on a datacenter GPU — to show that the conclusion holds across model size and hardware. The collected data lives under `experiments/worker-verifier-latency/data/qwen3_8b/` and `experiments/worker-verifier-latency/data/qwen3_32b/`.

- Script: `tools/vllm_worker_verify_pipeline/benchmark_worker_verifier_latency.py`
- Models / hardware:
- **Qwen3-8B** — single RTX 4090, `vllm==0.29.0`
- **Qwen3-32B** — single H100 NVL, `vllm==0.29.0`
- Common config: quant=bf16, dtype=bfloat16, tensor_parallel_size=1
- Sampling: top_k logprobs = 20, warmup=2, repeats=5, batch=1
- Backend: vLLM offline `LLM` class (in-process, V1 EngineCore)

Both paths share a single `vllm.LLM` instance (same model, same GPU state):

- **worker**: normal generation, `prefill(input) + decode(output_len)`, `SamplingParams(max_tokens=output_len, logprobs=20)`; output length is pinned exactly via `ignore_eos + min_tokens`.
- **verifier**: a single prefill over the worker's `input_ids + output_ids` to recover per-position top-k logprobs, `SamplingParams(max_tokens=1, prompt_logprobs=20)`.

1. Executive Summary

Core conclusion: the verifier's cost is far below the worker's generation cost — in typical multi-token output scenarios, verifying a full generation takes only about 1% of the cost to produce it (worker/verifier ≈ 80×–117×). This holds for both Qwen3-8B (RTX 4090) and Qwen3-32B (H100 NVL), i.e. it is robust to model size and hardware. This is direct evidence that the worker/verifier protocol is viable in terms of time cost.

The gap stems from the fundamentally different nature of the two paths. The worker spends nearly all its time on **serial decode**, emitting tokens one at a time — ~17.5 ms/token for the 8B on a 4090, and ~22.7 ms/token for the 32B on an H100 NVL — essentially independent of input length. The verifier performs a **single parallel prefill**, computing all positions of input+output at once at several thousand tok/s. The ~two-orders-of-magnitude efficiency gap between serial decode and parallel prefill is the source of the cost difference. Consequently, the longer the output, the worse the worker fares and the more favorable verification becomes: the ratio peaks around 100×–117× when output is large.

Notably, the 32B model is only ~30% slower per decode token than the 8B despite being ~4× larger, because the H100 NVL's much higher memory bandwidth largely offsets the extra weight traffic. The verifier's parallel prefill throughput is comparable across the two setups, so the overall worker/verifier ratio lands in the same band.

The only exception is when the output is extremely short (=1), where the verifier is actually slightly slower (ratio 0.62–0.90) — it must compute top-k logprobs at every position, and that fixed overhead slightly outweighs the tiny decode savings. This is a corner case and does not affect the main conclusion.

2. Worker Cost Model: Output-Dominated, Serial Decode

With input fixed at 128, worker time scales almost perfectly linearly with output, and the per-token cost holds steady — ~17.5 ms for the 8B, ~22.7 ms for the 32B:

output	8B worker med (s)	8B per token (ms)	32B worker med (s)	32B per token (ms)
1024	17.63	17.2	23.13	22.6
2048	35.40	17.3	46.33	22.6
4096	71.49	17.5	93.09	22.7
8192	145.77	17.8	187.66	22.9

Input has little effect on the worker — with output fixed at 1024, growing input from 128 to 8192 raises worker time by only ~10–12%:

input	8B worker med (s) @ out=1024	32B worker med (s) @ out=1024
128	17.63	23.13
512	17.71	23.21
1024	17.83	23.43
2048	18.14	23.67
4096	18.76	24.22
8192	19.80	25.44

The pure prefill cost can be read approximately from the output=1 rows (8B: input=8192 / out=1 → 0.886s; 32B: input=8192 / out=1 → 1.428s).

Conclusion: worker cost ≈ output_len × (per-token decode cost), consistent with decode being serial and memory-bandwidth-bound. The larger model pays more per token, but only modestly, thanks to the faster GPU's bandwidth.

3. Verifier Cost Model: Total-Dominated, Parallel Prefill

The verifier is a single prefill; its time scales linearly with total (input+output). Effective throughput is a few thousand tok/s for both setups:

total	8B verifier med (s)	8B tok/s	32B verifier med (s)	32B tok/s
129	0.033	(fixed overhead)	0.038	(fixed overhead)
1152	0.170	~6800	0.221	~5200
8320	1.488	~5600	1.924	~4325
16384	3.129	~5240	4.038	~4057

Conclusion: verifier cost ≈ total_tokens / (a few thousand tok/s) + fixed overhead. Prefill computes all positions in one parallel pass, making it two orders of magnitude faster than serial decode. The 32B verifier is only moderately slower than the 8B's despite the larger weights, again because the H100 NVL absorbs the extra bandwidth demand.

4. Key Metric: Worker/Verifier Cost Ratio

This is the most valuable output of the experiment. The ratio grows sharply with output before flattening. Representative scenarios for both models:

Scenario (input/output)	8B ratio	32B ratio
512 / 1024	**81.5×**	**85.7×**
128 / 2048	**113.6×**	73.6× [1]
2048 / 4096	**89.2×**	66.9×
1024 / 4096	**104.6×**	96.0×
128 / 4096	84.4×	**117.1×**

[1] The 32B `128/2048` cell is a noisy outlier: its verifier median (0.630s) sits above the mean (0.559s) and even the p10 (0.424s), so the ratio there is depressed by verifier-side jitter rather than a real effect.

4.1 Qwen3-8B (RTX 4090) — full ratio matrix

output →	1	16	64	256	1024	2048	4096	8192
input=128	0.82	8.36	26.27	70.10	103.78	113.56	84.45	97.99
input=512	0.65	3.67	13.20	39.35	81.51	99.84	76.71	97.24
input=1024	0.64	2.47	7.33	25.02	62.45	84.70	104.62	92.44
input=2048	0.70	1.59	4.39	14.60	42.78	64.15	89.18	82.83
input=4096	0.78	1.20	1.90	8.33	19.58	46.38	51.76	62.43
input=8192	0.62	0.81	1.40	3.88	12.19	21.16	36.65	50.13

4.2 Qwen3-32B (H100 NVL) — full ratio matrix

output →	1	16	64	256	1024	2048	4096	8192
input=128	0.90	9.67	30.06	75.97	104.79	73.59 [1]	117.05	97.56
input=512	0.89	4.27	13.93	42.31	85.69	98.90	106.36	96.07
input=1024	0.81	2.71	7.51	24.52	62.38	82.43	95.96	90.47
input=2048	0.89	1.78	4.14	14.44	41.38	48.15	66.90	81.38
input=4096	0.67	1.33	2.64	6.06	20.10	33.73	52.05	68.58
input=8192	0.80	0.97	1.52	3.86	11.88	21.14	34.78	48.49

How to read the tables (both matrices show the same structure):

- **The output=1 column is entirely < 1**: the verifier is slightly slower than the worker due to the per-position logprobs overhead.

- **Each row rises monotonically with output**: the longer the output, the more cost-effective verification becomes.
- **Each column decreases as input grows**: with longer input, the verifier must pay for the extra input-segment prefill, narrowing the ratio.
- **For a fixed input the ratio rises then flattens/dips slightly**: when output ≫ input, the verifier's total ≈ output also scales ∝ output, so the ratio asymptotes to `(per-token decode) / (per-token prefill) ≈ 90–110×`.
- **The two models track each other closely**: same shape, same magnitude, peaks in the 100×–117× range. The worker/verifier advantage is not an artifact of one particular model or GPU.

5. Notes and Limitations

- **The TTFT / decode split columns are empty (`-`).** Both runs used the vLLM V1 engine (the `EngineCore` in the logs), which does not expose the legacy `RequestMetrics.first_token_time`, so the script cannot separate TTFT from decode. This does not affect the total timings or any conclusion above; for a prefill approximation, use the output=1 rows.
- **This is single-request latency at batch=1.** In real online serving, the worker amortizes per-token cost via continuous batching, and the verifier can verify in batches, so absolute numbers will change. However, the fundamental difference between prefill (parallel) and decode (serial) remains, so the order-of-magnitude relationship — "verification is far cheaper than generation" — still holds.
- **Cross-run comparison caveat.** The 8B and 32B runs use different GPUs (RTX 4090 vs H100 NVL), so per-model absolute latencies are not directly comparable; what *is* comparable, and what this report emphasizes, is the worker/verifier *ratio* within each run.
- **A few individual cells are noisy** (notably 32B `128/2048`, see §4). Verifier times for small totals are dominated by fixed overhead and are more susceptible to jitter; treat single-cell anomalies with caution and rely on the overall trend.

6. Reproduction

Qwen3-8B (RTX 4090):

```
cd tools/vllm_worker_verify_pipeline
./benchmark_worker_verifier_latency.py \
  --model Qwen/Qwen3-8B \
  --dtype bfloat16 --trust-remote-code \
  --max-model-len 32768 \
  --gpu-memory-utilization 0.95 \
  --input-lengths "128 512 1024 2048 4096 8192" \
  --output-lengths "1 16 64 256 1024 2048 4096 8192" \
  --top-k 20 --warmup 2 --repeats 5 \
  --gpu "4090" \
  --output-dir qwen3_8b
```

Qwen3-32B (H100 NVL):

```
cd tools/vllm_worker_verify_pipeline
./benchmark_worker_verifier_latency.py \
  --model Qwen/Qwen3-32B \
  --dtype bfloat16 --trust-remote-code \
  --max-model-len 32768 \
  --gpu-memory-utilization 0.95 \
  --input-lengths "128 512 1024 2048 4096 8192" \
  --output-lengths "1 16 64 256 1024 2048 4096 8192" \
  --top-k 20 --warmup 2 --repeats 5 \
  --gpu "h100nvl" \
  --output-dir qwen3_32b
```

(These are the actual arguments that produced the committed data; the results are stored under `experiments/worker-verifier-latency/data/qwen3_8b/` and `experiments/worker-verifier-latency/data/qwen3_32b/`. Note `--max-model-len 32768` is required — the largest cells reach total = 16384 tokens, which an 8192 context cannot hold. The `--gpu` value is a free-form label recorded verbatim into `metadata.json`, not a detected device.)

Artifacts (per model dir): `latency_summary.csv` (aggregated metrics), `latency_raw.jsonl` (per-iteration raw timings), `summary.md` (Markdown table), `metadata.json` (args and environment).

Appendix A: Full Data Table — Qwen3-8B (RTX 4090)

input	output	total	worker med (s)	verifier med (s)	worker/verifier
128	1	129	0.0272	0.0332	0.82
128	16	144	0.2839	0.0340	8.36
128	64	192	1.1085	0.0422	26.27
128	256	384	4.4061	0.0629	70.10
128	1024	1152	17.6304	0.1699	103.78
128	2048	2176	35.3973	0.3117	113.56
128	4096	4224	71.4902	0.8465	84.45
128	8192	8320	145.7669	1.4876	97.99
512	1	513	0.0549	0.0849	0.65
512	16	528	0.3137	0.0854	3.67
512	64	576	1.1387	0.0863	13.20
512	256	768	4.4452	0.1130	39.35
512	1024	1536	17.7100	0.2173	81.51
512	2048	2560	35.5538	0.3561	99.84
512	4096	4608	71.8107	0.9361	76.71
512	8192	8704	146.3028	1.5046	97.24
1024	1	1025	0.1016	0.1586	0.64
1024	16	1040	0.3624	0.1467	2.47
1024	64	1088	1.1905	0.1625	7.33
1024	256	1280	4.5075	0.1802	25.02
1024	1024	2048	17.8318	0.2855	62.45
1024	2048	3072	35.7824	0.4224	84.70
1024	4096	5120	72.2362	0.6905	104.62
1024	8192	9216	147.0083	1.5903	92.44
2048	1	2049	0.2040	0.2910	0.70
2048	16	2064	0.4668	0.2937	1.59
2048	64	2112	1.3059	0.2973	4.39
2048	256	2304	4.6611	0.3193	14.60
2048	1024	3072	18.1371	0.4239	42.78
2048	2048	4096	36.2336	0.5648	64.15
2048	4096	6144	73.0383	0.8190	89.18
2048	8192	10240	148.4488	1.7922	82.83

input	output	total	worker med (s)	verifier med (s)	worker/verifier
4096	1	4097	0.4181	0.5344	0.78
4096	16	4112	0.6856	0.5693	1.20
4096	64	4160	1.5416	0.8107	1.90
4096	256	4352	4.9817	0.5982	8.33
4096	1024	5120	18.7645	0.9585	19.58
4096	2048	6144	37.2070	0.8023	46.38
4096	4096	8192	74.6061	1.4413	51.76
4096	8192	12288	151.2620	2.4228	62.43
8192	1	8193	0.8863	1.4333	0.62
8192	16	8208	1.1655	1.4470	0.81
8192	64	8256	2.0467	1.4572	1.40
8192	256	8448	5.5912	1.4422	3.88
8192	1024	9216	19.7963	1.6246	12.19
8192	2048	10240	38.8759	1.8373	21.16
8192	4096	12288	77.5542	2.1159	36.65
8192	8192	16384	156.8459	3.1287	50.13

Appendix B: Full Data Table — Qwen3-32B (H100 NVL)

input	output	total	worker med (s)	verifier med (s)	worker/verifier
128	1	129	0.0341	0.0380	0.90
128	16	144	0.3699	0.0383	9.67
128	64	192	1.4316	0.0476	30.06
128	256	384	5.7371	0.0755	75.97
128	1024	1152	23.1320	0.2208	104.79
128	2048	2176	46.3294	0.6296	73.59
128	4096	4224	93.0909	0.7953	117.05
128	8192	8320	187.6646	1.9236	97.56
512	1	513	0.0869	0.0973	0.89
512	16	528	0.4369	0.1024	4.27
512	64	576	1.5196	0.1091	13.93
512	256	768	5.8605	0.1385	42.31
512	1024	1536	23.2122	0.2709	85.69
512	2048	2560	46.5517	0.4707	98.90
512	4096	4608	93.1237	0.8756	106.36
512	8192	8704	188.3937	1.9609	96.07
1024	1	1025	0.1595	0.1962	0.81
1024	16	1040	0.5115	0.1884	2.71
1024	64	1088	1.6060	0.2138	7.51
1024	256	1280	5.9531	0.2428	24.52
1024	1024	2048	23.4337	0.3757	62.38
1024	2048	3072	46.7594	0.5673	82.43
1024	4096	5120	93.7020	0.9765	95.96
1024	8192	9216	188.9899	2.0891	90.47
2048	1	2049	0.3371	0.3772	0.89
2048	16	2064	0.6822	0.3837	1.78
2048	64	2112	1.7670	0.4273	4.14
2048	256	2304	6.1526	0.4262	14.44
2048	1024	3072	23.6689	0.5720	41.38
2048	2048	4096	46.8743	0.9736	48.15
2048	4096	6144	94.0470	1.4059	66.90
2048	8192	10240	189.6197	2.3300	81.38

input	output	total	worker med (s)	verifier med (s)	worker/verifier
4096	1	4097	0.6376	0.9557	0.67
4096	16	4112	0.9802	0.7394	1.33
4096	64	4160	2.1058	0.7964	2.64
4096	256	4352	6.5474	1.0798	6.06
4096	1024	5120	24.2233	1.2049	20.10
4096	2048	6144	47.9006	1.4200	33.73
4096	4096	8192	95.5773	1.8361	52.05
4096	8192	12288	191.8452	2.7974	68.58
8192	1	8193	1.4284	1.7834	0.80
8192	16	8208	1.8025	1.8587	0.97
8192	64	8256	2.9443	1.9347	1.52
8192	256	8448	7.4379	1.9247	3.86
8192	1024	9216	25.4369	2.1414	11.88
8192	2048	10240	49.5643	2.3441	21.14
8192	4096	12288	98.0349	2.8185	34.78
8192	8192	16384	195.8385	4.0383	48.49

Compare with the earlier Qwen2.5-7B verification-cost experiment →

Back to papers and experiments →