

Parakeet vs Whisper Turbo and Distil on CPU

Asad Shehzad

Loudness.ai · LA-RB-2026-001 · 14 October 2026 · Version 1.0
DOI: [10.5281/zenodo.23374001](https://doi.org/10.5281/zenodo.23374001)

Abstract

We evaluated three Parakeet TDT 0.6B v3 configurations and two quantized Whisper variants on identical English reading audio using four CPU threads. On a 310.425-second recording, Parakeet FP32 with shorter chunks took 53.58 seconds with 2.81% word error rate (23 edits across 818 reference words) and 2.87 GiB peak process memory. Whisper Turbo INT8 took 343.26 seconds with 2.20% WER; Distil-Whisper INT8 took 279.10 seconds with 2.57% WER. Default Parakeet INT8 chunks omitted substantial speech; shorter chunks reduced WER from 39.12% to 4.65%. These single-request observations favor FP32 Parakeet for this CPU and sample, while the tested Whisper variants made fewer word edits. We publish all 18 completed runs, transcripts, fixture audio, references, checksums and reproduction settings.

Five configurations, one recording

Every comparison row uses the same 310.425-second, 818-word WAV, verified by SHA-256. All experiments used a four-core CPU quota and four inference threads on the same shared virtual server. The full Whisper large-v3 model was not tested; the comparison covers Turbo and Distil variants only.

Whisper Turbo took 6.41 times as long as FP32 Parakeet; Distil took 5.21 times as long. Their error counts were lower: 18 and 21 edits, versus 23 for FP32 Parakeet. Short-chunk INT8 Parakeet used less memory, but its 38 edits exceeded FP32's 23. These values describe completed requests, not estimated performance.

Configuration	Time (s)	WER	Edits	RSS (GiB)
Whisper Turbo INT8	343.26	2.20%	18/818	2.09
Distil-Whisper INT8	279.10	2.57%	21/818	1.96
Parakeet INT8 · default	68.22	39.12%	320/818	2.02
Parakeet INT8 · short	53.01	4.65%	38/818	1.46
Parakeet FP32 · short	53.58	2.81%	23/818	2.87

All completed runs

The study contains twelve Parakeet and six Whisper runs. Each long clean or noisy configuration was measured once. Only the short clean recordings include a first request and a warm repeat. No error bars or confidence intervals are supported by this design. The download includes full-precision metrics, per-run CPU time, sampled memory, background load averages and transcripts.

The case names 30sec and 5min are fixture labels; the exact durations are 36.85 and 310.425 seconds. A prepared 1,203.660-second fixture was retained, but its earlier Whisper run was aborted without a completed result and is excluded from the tables.

Configuration / case	Time (s)	Edits / ref.	WER	RSS (GiB)
INT8, default chunks / 30sec-first-request	6.11	25/93	26.88%	1.29
INT8, default chunks / 30sec-warm-repeat	5.89	25/93	26.88%	1.52
INT8, default chunks / 5min-warm	68.22	320/818	39.12%	2.02
INT8, default chunks / 5min-noise10db-warm	53.73	111/818	13.57%	2.18
INT8, short chunks / 30sec-first-request	8.27	9/93	9.68%	1.35
INT8, short chunks / 30sec-warm-repeat	5.81	9/93	9.68%	1.40
INT8, short chunks / 5min-warm	53.01	38/818	4.65%	1.46
INT8, short chunks / 5min-noise10db-warm	57.27	64/818	7.82%	1.46
FP32, short chunks / 30sec-first-request	9.51	3/93	3.23%	2.79
FP32, short chunks / 30sec-warm-repeat	6.05	3/93	3.23%	2.81
FP32, short chunks / 5min-warm	53.58	23/818	2.81%	2.87
FP32, short chunks / 5min-noise10db-warm	59.20	37/818	4.52%	2.87
Whisper Turbo INT8 / 30sec-first-request	70.48	2/93	2.15%	1.97
Whisper Turbo INT8 / 30sec-warm-repeat	66.04	2/93	2.15%	1.97
Whisper Turbo INT8 / 5min-warm	343.26	18/818	2.20%	2.09
Distil-Whisper INT8 / 30sec-first-request	49.85	2/93	2.15%	1.83
Distil-Whisper INT8 / 30sec-warm-repeat	42.57	2/93	2.15%	1.84
Distil-Whisper INT8 / 5min-warm	279.10	21/818	2.57%	1.96

Hardware and measurement boundaries

The KVM guest exposed 12 AMD EPYC virtual CPUs, x86-64 AVX2 and approximately 47 GiB RAM. VNNI was not exposed. This does not establish physical-core clocks or dedicated-machine performance. Inference had a four-core CPU quota, four inference threads, low scheduling priority and no swap allowance. Setup and model downloads ran separately; completed inference used cached files with external networking disabled.

Parakeet elapsed time covers local HTTP upload, multipart parsing, WAV decoding, pause-aware voice activity detection, chunking, model inference, stitching, timestamp processing, JSON serialization and receipt. Whisper time covers the direct OpenVINO pipeline call, including input conversion. Both exclude model startup, disk reading, scoring, Internet upload and queue delay. The different measurement boundaries make this an application-path comparison, not an identical-runtime kernel benchmark.

Parakeet process RSS was sampled every 50 milliseconds and includes mapped model and library pages. It excludes the client process and is distinct from aggregate operating-system memory accounting. The five-way table uses each request's sampled peak RSS. All tests ran sequentially at different times on a shared production host; background activity was observed, not controlled. No concurrency or simultaneous-user

capacity was measured.

Exact models and runtime settings

Parakeet used the [groxaxo FastAPI implementation](#) at commit 06464cd06a0f75426ce46e94c2d04ee7e26f48c0, with [istupakov's ONNX export](#) at revision 8f23f0c03c8761650bdb5b40aaf3e40d2c15f1ce. INT8 and FP32 loaded different precision files from that snapshot. The model and decoding source were unchanged; tuning used environment settings. Encoder and decoder-joint execution providers were verified as CPU-only.

Parakeet runtime: Python 3.12.3, ONNX Runtime CPU 1.23.2, onnx-asr 0.12.0, NumPy 1.26.4, PyTorch 2.8.0+cpu, SoundFile 0.13.1 and Silero VAD 6.2.3. Settings: four intra-operation threads, one inter-operation thread, sequential execution, full graph optimization, one inference worker, one audio worker, one web worker and cross-request batching disabled. BLAS and OpenMP pools were limited to one thread.

Default Parakeet chunks used minimum/target/maximum lengths of 20/60/75 seconds. Both short-chunk profiles used 5/12/15 seconds. Voice activity detection and pause boundaries are part of the tested behavior. The clean long INT8 short run produced 30 chunks; FP32 produced 31. This difference prevents attributing every quality change solely to arithmetic precision. Startup was approximately 8.8 seconds per Parakeet profile; first-use voice activity detection affected short first requests.

Whisper used [OpenVINO Whisper large-v3 Turbo INT8](#) at revision b568445dd5dc8c695bde596f8acbb4694fd6ba64 and [OpenVINO Distil-Whisper large-v3 INT8](#) at revision 7210c37ac7fb59669a9768fe194c5fe022ddc743. Runtime: OpenVINO 2026.1.0, OpenVINO GenAI 2026.1.0.0 and Python 3.12.3; four threads, one stream, latency mode, greedy decoding with English specified and segment timestamps. Startup was 4.88 seconds for Turbo and 3.90 for Distil, measured separately.

Audio provenance and scoring

The source is [LibriSpeech test-clean \(OpenSLR 12\)](#), by Vassil Panayotov and collaborators, derived from LibriVox audiobooks and distributed under [CC BY 4.0](#). The archive passed published MD5 32fa31d27d2e1cad72775fee3f4849a9 and SHA-256 39fde525e59672dc6d1551919b1478f724438a95aa55f874b576be21967e6c23. Fixtures use deterministic seed-42 shuffling, whole utterances, 0.25-second silent gaps, 16 kHz mono PCM16 WAV. Utterance IDs, speaker selection, original references and exact file hashes are supplied in the study package.

The short fixture contains 93 reference words across four utterances; the long fixture contains 818 reference words. The short IDs are 4970-29093-0015, 5639-40744-0030, 8555-292519-0011 and 121-127105-0034. Parakeet's clean files are byte-identical to the earlier Whisper files. These are concatenated, multi-speaker reading samples, not continuous meetings.

The noisy fixture adds seed-42 Gaussian white noise to the long recording at 10 dB whole-file energy SNR. The mix is uniformly attenuated if its peak would exceed 0.98, then encoded as PCM16. Recorded pre-encoding SNR was 9.9999996 dB and gain was 1.0. This is synthetic stationary noise; it is not traffic, room noise or a real meeting. Whisper was not tested on this noisy file.

Word error rate is exact Levenshtein word edit distance divided by reference word count. Text is lowercased; curly apostrophes are converted; tokenization retains alphanumeric words and internal apostrophes. The exact regular expression and scoring source are included. WER measures substitutions, deletions and insertions; it does not directly score punctuation, capitalization, speaker attribution, important

names or caption readability.

Omissions, chunking and noise

The default INT8 short transcript had 69 normalized words against 93 reference words and missed the final two utterances. The long transcript had 513 against 818. In the short request, the last real speech token ended around 25.5 seconds, while the wrapper extended final punctuation and segment end to the full 36.85-second input. An end timestamp matching the recording length therefore does not demonstrate complete recognition.

The response's words array contains tokenizer pieces such as *testi*, *mon* and *ies*, rather than regrouped whole words. Segment endings can be synthesized from chunk boundaries. These outputs are preserved for inspection; reliable word alignment and subtitle readability require separate evaluation.

Shorter chunks reduced clean long INT8 WER from 39.12% to 4.65%. FP32 reduced it further to 2.81%, with almost the same elapsed time. On the noisy recording, WER was 13.57% for default INT8, 7.82% for short INT8 and 4.52% for short FP32. The default INT8 result improved with added noise; this unusual result may involve VAD or decoding behavior and does not imply that noise generally improves recognition.

The [ONNX Runtime quantization documentation](#) describes possible U8S8 saturation on AVX2 hardware. It is a plausible hypothesis for some INT8 degradation, not a demonstrated cause here. We did not test reduced-range quantization, alternate exports or controlled identical chunk boundaries.

What these results support

For this virtual CPU, export and small English sample, short-chunk FP32 Parakeet offers the strongest observed speed and error tradeoff among the tested Parakeet profiles. It ran offline after setup. This is a configuration choice supported by measured requests, not a claim that Parakeet is universally more accurate than Whisper.

The Parakeet checkpoint covers English and 24 other European languages according to its [NVIDIA model card](#); this experiment evaluates English only. It supports no accuracy conclusion for Urdu, Hindi, Arabic, code switching or specific accents. Distil is English-only; Turbo is multilingual, but multilingual behavior was not measured.

Threats to validity include the small concatenated sample, abrupt speaker/context changes, single long-file measurements, uncontrolled shared-host load, unequal timing boundaries and nonidentical VAD boundaries. We measured neither full-corpus accuracy nor tail latency, concurrent capacity, physical CPU differences, real-world noise robustness or a full Whisper large-v3 baseline. A broader study should add independent corpora, repeated requests, fixed chunk controls, concurrency trials and a timestamp-specific evaluation.

Reproduce and inspect the study

Download the study package for all 18 runs, comparison CSV/JSON, twelve Parakeet API responses, six Whisper segment responses, raw transcripts, clean/noisy WAV fixtures, reference text, original utterance-selection records, model-file checksums, dependency locks and portable benchmark scripts. The scripts use the pinned upstream implementation and support the three tested chunk/precision profiles; they reproduce request and scoring boundaries. Download the models separately under their original licenses. CPU resource limits and isolated networking must be applied by the reproducing researcher.

Provision isolated Python environments from the supplied locks, obtain the pinned source and model revisions, verify checksums, and use the supplied WAVs. Set four inference threads and a four-core quota. Run each Parakeet profile separately with one worker and cached models, then the two Whisper pipelines with English specified. Keep startup separate from request timing, sample Parakeet RSS every 50 milliseconds and Whisper RSS every 100 milliseconds and score against the supplied references. Retain raw responses and runtime/provider checks before comparing summaries.

The Parakeet checkpoint's model card lists CC BY 4.0; the tested Whisper model cards list MIT. Model weights are not redistributed in this package. LibriSpeech audio and reference derivatives retain their CC BY 4.0 attribution; modifications are fixture concatenation and synthetic noise addition. The report is an authored technical report published by Loudness.ai, version 1.0. The complete original internal evidence archive is preserved separately from the public research package.

References

Panayotov, V., Chen, G., Povey, D. and Khudanpur, S. (2015). LibriSpeech: an ASR corpus based on public domain audio books. [ICASSP 2015](#), DOI: [10.1109/ICASSP.2015.7178964](#). Corpus: [OpenSLR 12](#).

NVIDIA. [Parakeet TDT 0.6B v3 model card](#). istupakov. [Parakeet ONNX export](#). groxaxo. [Parakeet FastAPI implementation](#). Pinned revisions are listed in Methods and in the package.

OpenVINO. [Whisper large-v3 Turbo INT8 model card](#); [Distil-Whisper large-v3 INT8 model card](#). ONNX Runtime. [Quantize ONNX models: performance and accuracy considerations](#). Sources accessed 14 October 2026.

Citation

Shehzad, Asad (2026). Parakeet vs Whisper Turbo and Distil on CPU. Loudness.ai technical report LA-RB-2026-001, version 1.0. <https://loudness.ai/research/parakeet-whisper-cpu-benchmark/>

DOI: [10.5281/zenodo.23374001](#)

© 2026 Asad Shehzad / Loudness.ai