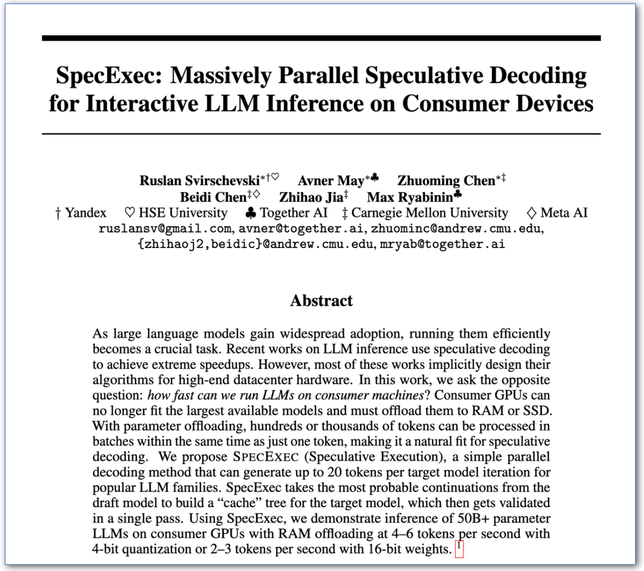
SpecExec: Massively Parallel Speculative Decoding for Interactive LLM Inference on Consumer Devices

Ruslan Svirschevski* (Yandex, HSE University)
Avner May* (Together AI)
Zhuoming Chen* (Carnegie Mellon University)
Beidi Chen (Carnegie Mellon University Meta AI)
Zhihao Jia(Carnegie Mellon University)
Max Ryabinin(Together AI)

Method highlights

	SpecInfer	SpecExec(ours)
Token source	Sampled from draft	Cherry-picked from draft
Repeat sampling prob. adjustment	Required	Not required
Acceptance probability	Depends on P_target / P_draft ratio	Depends on P_target only
Draft tree shape	Only based on random draft sampling	Any
Best setup	Aligned distributions	Spiked distributions


SpecExec: Massively Parallel Speculative Decoding for Interactive LLM Inference on Consumer Devices

Ruslan Svirschevski[*†♡] Avner May[*♣] Zhuoming Chen[*‡]
Beidi Chen[‡◇] Zhihao Jia[‡] Max Ryabinin[♣]
† Yandex ♡ HSE University ♣ Together AI ‡ Carnegie Mellon University ◇ Meta AI
ruslansv@gmail.com, avner@together.ai, zhuominc@andrew.cmu.edu,
{zhihaoj2,beidic}@andrew.cmu.edu, mryab@together.ai

Abstract

As large language models gain widespread adoption, running them efficiently becomes a crucial task. Recent works on LLM inference use speculative decoding to achieve extreme speedups. However, most of these works implicitly design their algorithms for high-end datacenter hardware. In this work, we ask the opposite question: *how fast can we run LLMs on consumer machines*? Consumer GPUs can no longer fit the largest available models and must offload them to RAM or SSD. With parameter offloading, hundreds or thousands of tokens can be processed in batches within the same time as just one token, making it a natural fit for speculative decoding. We propose SpecExec (Speculative Execution), a simple parallel decoding method that can generate up to 20 tokens per target model iteration for popular LLM families. SpecExec takes the most probable continuations from the draft model to build a "cache" tree for the target model, which then gets validated in a single pass. Using SpecExec, we demonstrate inference of 50B+ parameter LLMs on consumer GPUs with RAM offloading at 4–6 tokens per second with 4-bit quantization or 2–3 tokens per second with 16-bit weights.

High acceptance length

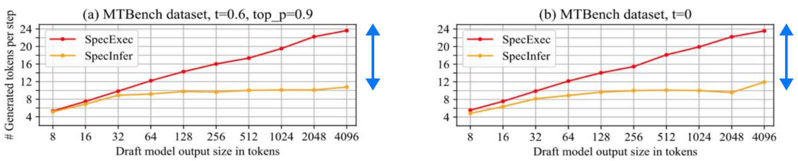


Figure 3: Generation rate vs draft size for Llama 2-7B/70B chat models, MTBench [63] dataset.

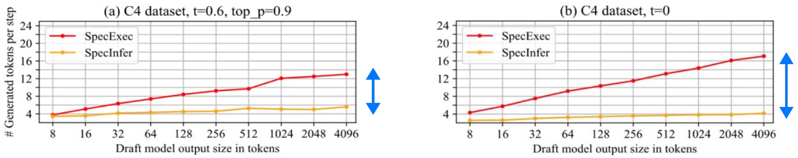
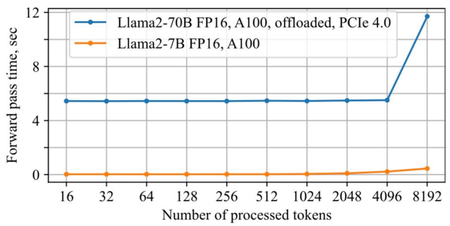


Figure 4: Generation rate vs draft size for Llama 2-7B/70B models, C4 dataset.

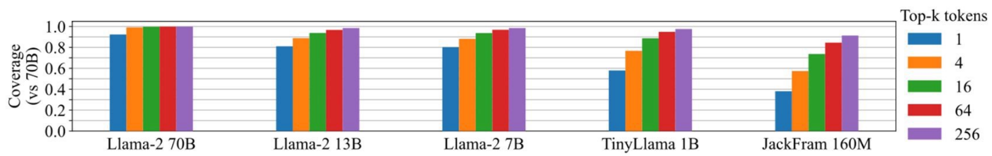
Paper: arxiv.org/abs/2406.02532

Demo: github.com/yandex-research/specexec

Performance Factors: parallel decoding



Performance Factors: distribution alignment



Method Performance

(1) Inference speed with RAM offloading, A100 GPU, Chat / Instruct models, using SpecExec (SX) vs SpecInfer (SI) methods.

Draft / Target models	Dataset	t	Method	Budget	Gen. rate	Speed, tok/s	Speedup
Llama 2-7B / 70B	OAsst	0.6	SX	2048	20.60	**3.12**	**18.7x**
		0.6	SI	1024	8.41	1.34	8.0x
		0	SX	1024	18.8	**2.74**	**16.4x**
		0	SI	1024	7.86	1.18	7.1x
Llama 2-7B / 70B GPTQ	OAsst	0.6	SX	128	12.10	6.02	8.9x
		0	SX	256	13.43	6.17	9.1x
Mistral-7B / Mixtral-8x7B	OAsst	0.6	SX	256	12.38	3.58	3.5x
Llama 3-8B / 70B		0.6	SX	1024	18.88	2.62	15.6x
Llama 3-8B / 70B	MTBench	0.6	SX	1024	18.16	2.79	16.6x
		0	SX	2048	21.58	2.94	17.5x

(2) Inference speed with RAM offloading. A100 GPU, base models SpecExec (SX) vs SpecInfer (SI).

Draft / Target models	Dataset	t	Method	Budget	Gen. rate	Speed, tok/s	Speedup
Llama 2-7B / 70B	C4	0.6	SX	2048	12.9	**1.97**	**11.8x**
		0.6	SI	1024	6.48	1.03	6.2x
		0	SX	2048	16.1	**2.38**	**14.3x**
		0	SI	1024	4.78	0.75	4.5x
Llama 2-7B / 70B	WikiText-2	0.6	SX	2048	9.57	**1.54**	**9.2x**
		0.6	SI	1024	4.69	0.77	4.6x
		0	SX	2048	11.74	**1.88**	**11.3x**
		0	SI	1024	3.71	0.62	3.6x
Llama 2-7B / 70B GPTQ	WikiText-2	0.6	SX	256	6.99	3.72	5.5x
		0	SX	256	8.81	4.54	6.7x
Mistral-7B / Mixtral-8x7B	WikiText-2	0.6	SX	128	6.56	3.23	3.2x

(3) Inference speed on consumer GPUs with offloading, chat/instruct models, Llama 2 70B-GPTQ target, t = 0.6, OpenAssistant dataset.

GPU	Draft model	Budget	Gen. rate	Speed, tok/s	Speedup
RTX 4090	Llama 2-7B	256	13.46	5.66	8.3x
RTX 4060		128	9.70	3.28	4.6x
RTX 3090		256	14.3	3.68	10.6x
RTX 2080Ti	ShearedLlama-1.3B	128	7.34	1.86	6.1x

(4) Inference speed without offloading, A100 GPU.

Draft / Target models	Dataset	t	Method	Budget	Gen. rate	Speed, tok/s	Speedup
SL-1.3B / Vicuna-33B	OASST-1	0.6	SX	128	5.33	31.6	2.15x
	OASST-1	0	SX	128	5.4	32.94	2.24x
	C4	0.6	SX	128	5.1	33.3	2.26x
	C4	0	SX	128	5.36	35.62	2.42x
	WikiText-2	0.6	SX	128	4.87	30.19	1.90x
	WikiText-2	0	SX	128	5.24	33.15	2.08x