Quantization Benchmarking with LLaMA:

Author: Damir Sarsenov
Course: Advanced Computer Architecture I
Term: Fall 2025

Abstract

This survey distills insights from two peer-reviewed research papers aligned with computer architecture: (1) GPTQ (ICLR 2023), a post-training, weight-only quantization method that compresses Transformer weights to 3–4 bits using approximate second-order information; and (2) Tender (ISCA 2024), an algorithm-hardware co-design that uses tensor decomposition and runtime requantization policies to enable efficient low-precision deployment. We analyze how each approach impacts model quality, performance (TTFT, tokens/s), memory footprint, and energy considerations for LLaMA-class models. We conclude with a concise protocol that future work can adopt to report system-level metrics alongside model quality.

1. Introduction & Background

Large Language Models (LLMs) like the LLaMA family stress memory capacity/bandwidth and raise latency, throughput, and energy concerns - all first-class topics in computer architecture. Quantization reduces numerical precision to decrease memory footprint and increase arithmetic intensity. This review focuses on two representative research directions: GPTQ - weight-only post-training quantization - and Tender - an algorithm-hardware co-design for low-precision inference.

2. Motivation

- Capacity & bandwidth: Lower-bit weights shrink HBM footprint and traffic.
- Latency/throughput: Reduced precision can increase tokens/s if kernels and scheduling are well supported.
- Energy considerations: Smaller data movement and higher utilization can reduce energy per token.

3. Problem Statement & Research Focus

Goal. Survey and compare GPTQ (ICLR 2023) and Tender (ISCA 2024) as two complementary paths for efficient LLM inference, with emphasis on quality, memory, performance, and energy.

Scope. Models: LLaMA-class models as representative. Metrics: quality (perplexity or task scores), performance (TTFT, tokens/s), weights-only memory footprint, and discussion of energy per token. Deliverables: a critical survey and a proposed protocol for future, fair benchmarks.

4. Related Research (Two Papers)

4.1 GPTQ (ICLR 2023) — Proposed Solution

Proposed solution. GPTQ is a one-shot, post-training, weight-only quantization method. It minimizes reconstruction error using approximate second-order (Hessian) information to preserve accuracy at 3–4 bits.

Architectural relevance. Weight-only INT3/INT4 substantially reduces memory bandwidth pressure and may increase throughput with suitable kernels. The authors report that GPTQ can quantize very large models within a few GPU hours and demonstrate non-trivial speedups.

4.2 Tender (ISCA 2024) — Proposed Solution

Proposed solution. Tender introduces decomposed quantization with power-of-two-spaced scales and runtime requantization policies, allowing accumulation across decomposed matrices without frequent de/quantization. This reduces overheads and aligns better with existing tensor hardware.

Architectural relevance. Tender's co-design targets efficient low-precision execution paths, aiming to improve accuracy and throughput with minimal hardware changes.

5. Memory Footprint: Back-of-the-Envelope

Model	FP16 (16-bit)	INT8 (8-bit)	INT4 (4-bit)
LLaMA-3 8B	~16 GB	~8 GB	~4 GB
LLaMA-3 70B	~140 GB	~70 GB	~35 GB

Formula: bytes ≈ params × (bits/8). Excludes KV cache/activations.

6. Findings from the Two Papers

6.1 Accuracy/Quality

- GPTQ shows that 3–4-bit weight-only PTQ can closely track full-precision quality on language modeling benchmarks for large Transformers.
- Tender reports higher accuracy than prior low-precision schemes under its decomposition and requantization strategy.

6.2 Performance (TTFT, tokens/s)

- GPTQ reports end-to-end inference speedups vs FP16 with optimized kernels on A100/A6000 in their evaluations.
- Tender reports throughput gains by avoiding frequent requantization when accumulating partial sums, improving hardware utilization.

6.3 Energy Considerations

- Both methods reduce data movement due to smaller weights, which generally correlates with lower energy per token; exact savings are workload-dependent.

7. Comparative Summary

Method	Precision Regime	Key Idea	Strengths	Caveats
GPTQ (ICLR'23)	W4 (3–4-bit weights only)	Hessian-aware one-shot PTQ	Strong accuracy-vs-size; simple deployment path	Throughput depends on kernel maturity; activations remain high precision
Tender (ISCA'24)	Low-precision with decomposed tensors	Power-of-two scale spacing avoids frequent requantization	Higher accuracy/throughput vs prior schemes; minimal HW changes	Requires co-design assumptions; reproduction may need research code

8. Discussion: A Proposed Protocol for Future Benchmarks

Based on our survey of GPTQ and Tender, we propose the following **future benchmark protocol** to enable fair, reproducible comparisons without claiming we executed a large study.

- Report TTFT, tokens/s, weights-only memory, and (when possible) energy per token together with quality.
- Keep serving engine and kernel versions constant; fix prompts/seeds; identical scheduling flags across runs.
- Include at least two precisions: FP16/BF16 (baseline) and a weight-only low-bit setting (e.g., 4-bit via GPTQ).
- If pursuing Tender-style ideas, document decomposition configuration and any runtime requantization behavior explicitly.

9. Conclusion & Future Work

Weight-only PTQ (GPTQ) and algorithm-hardware co-design (Tender) highlight two complementary paths to efficient LLaMA inference. GPTQ demonstrates that 3–4-bit weights can maintain quality while shrinking memory and enabling speedups with appropriate kernels. Tender proposes a principled route to low-precision accumulation with fewer requantization steps, improving accuracy and throughput with minimal hardware changes. Future work should standardize reporting of TTFT, tokens/s, memory, and energy per token.

References

1) Frantar, Ashkboos, Hoefler, Alistarh. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. ICLR 2023.

2) Lee, Lee, Sim. Tender: Accelerating Large Language Models via Tensor Decomposition and Runtime Requantization. ISCA 2024.