TECHNICAL DOSSIER // AI & SOFTWARE

Grouped-Query Attention (GQA): LLM KV Cache Memory Optimization

Hardware Memory Bandwidth Limits, Arithmetic Intensity, and Serving Latency Physics

Published: October 29, 2026 | Distribution: Open Engineering Community | Document ID: LT-2026-B15

EXECUTIVE SUMMARY & MICROARCHITECTURAL SCOPE

This whitepaper evaluates Grouped-Query Attention (GQA) and Multi-Query Attention (MQA) against standard Multi-Head Attention (MHA) under hardware memory bandwidth roofline constraints. During autoregressive LLM decoding, memory bandwidth to High-Bandwidth Memory (HBM) becomes the dominant bottleneck. GQA partitions query heads into groups sharing single key-value projection heads, achieving up to an 8x reduction in KV cache memory traffic while maintaining model quality comparable to full MHA.

Attention Architecture	Multi-Head Attention (MHA)	Grouped-Query Attention (GQA-8)	Multi-Query Attention (MQA)
KV Cache Memory Footprint	100% (Baseline)	12.5% to 25% (4x-8x Reduction)	3.1% (32x Reduction)
Decoding Arithmetic Intensity	0.25 to 1.0 FLOP / Byte	2.0 to 8.0 FLOP / Byte	8.0 to 32.0 FLOP / Byte
HBM Bandwidth Saturation	Severe (Memory-Bound)	Balanced (High Throughput)	Compute-Bound (Small Batches)
Perplexity / Quality Gap	Optimal (Baseline)	Negligible (< 0.2% delta)	Measurable degradation on long-context

1. The Memory Wall in Autoregressive LLM Inference

Large Language Model execution divides into two distinct operational regimes: the compute-bound prefill phase (processing the input prompt) and the memory-bound decoding phase (generating tokens one by one). In decoding, each new token requires loading the entire accumulated Key-Value (KV) cache for all previous context tokens from off-chip HBM into on-chip SRAM. At long context lengths (e.g., 32k to 128k tokens), KV cache memory access completely saturates the memory bus.

2. Multi-Head vs. Multi-Query Attention Architecture

Standard MHA allocates independent projection matrices for queries, keys, and values across H heads (e.g., H=32, d_k=128). In Multi-Query Attention (MQA), all H query heads share a single key head and value head. While MQA slashes KV cache memory footprint by a factor of H, it frequently causes noticeable degradation in complex reasoning, multi-turn dialogue, and code generation due to constrained subspace capacity.

3. Grouped-Query Attention (GQA) Structural Mechanics

GQA generalizes MHA and MQA by grouping H query heads into G groups (where 1 < G < H). Each group of (H/G) query heads shares a single key head and value head. For instance, in LLaMA-2/3 70B (H=64, G=8), 8 query heads share each KV head, providing an 8x reduction in KV cache memory storage and bandwidth requirements while retaining 8 independent subspace key-value representations, yielding quality virtually indistinguishable from MHA.

4. Roofline Model Shifts & Serving Throughput Scaling

On NVIDIA H100/H200 and AMD MI300X accelerators, GQA shifts the operational point along the Roofline curve. By reducing bytes transferred per token, GQA increases arithmetic intensity (FLOPs / Byte), moving decoding closer to the compute roofline. In production serving engines (vLLM, TensorRT-LLM), the reduced footprint enables 4x to 8x larger serving batch sizes within the same VRAM, dramatically increasing overall system token throughput.

STRATEGIC ENGINEERING TAKEAWAYS & DEPLOYMENT CHECKLIST

- **Bandwidth Amplification:** GQA provides the optimal trade-off between KV cache memory reduction and model accuracy across long-context reasoning tasks.
- **Serving Density Multiplier:** Transitioning from MHA to GQA-8 allows cloud providers to scale concurrent user batch sizes by up to 400% on identical accelerator clusters.
- **Paging Integration:** Coupling GQA with PagedAttention and FP8 KV cache quantization virtually eliminates memory fragmentation and VRAM over-allocation.