Title- Hardware/Software Co-Design for LLM Quantization
Team 01: Cipher Core (Utkarsh Lubal & Sami Abedi)

Objective- This project investigates how **hardware/software co-design** principles can enhance the efficiency of **Large Language Models (LLMs)** through **quantization**, the process of reducing numerical precision (e.g., FP16 → INT8 → INT4) to lower computational and memory costs. The goal was to empirically evaluate performance, memory, and accuracy trade-offs across frameworks and GPU architectures, providing actionable insights for efficient model deployment.

Methodology-

1. **Hardware Setup:**
   - ○ Primary GPU – *NVIDIA Tesla T4* (2,560 CUDA cores, 320 Tensor cores, 15.8 GB GDDR6).
2. **Models Tested:**
   - ○ distilgpt2 (82 M), DialoGPT-small (124 M), TinyLlama (1.1 B), Llama-3.2 (1 B)
3. **Frameworks:**
   - ○ **BitsAndBytes (INT8)** – PyTorch integration for 8-bit quantization
   - ○ **ONNX Runtime (INT8)** – hardware-assisted inference with KV cache
   - ○ **GGUF/llama.cpp (INT4)** – 4-bit quantization for large models
4. **Pipeline:** Load Model → Quantize → Warm-up → Benchmark (100 runs) → Measure Memory → Compute Perplexity → Record Results

Metric	**Observation**	**Insight**
Speed	Llama-3.2-1B INT4→4.55× faster; ONNX INT8→1.69× speedup	Quantization gains scale with model size.
Memory	Up to 75 % reduction (INT4), 50 % for INT8 models	Enables deployment on limited-VRAM devices.
Accuracy	Perplexity increase ≤ 3 %	Minimal quality degradation.
Energy	Power draw ↓ from 45 W → 38 W (14.4 % reduction)	Quantization improves energy efficiency.
Hardware Utilization	GPU utilization ↑ to 78 % for large INT4 models	Larger models exploit tensor-core throughput.

Comparative Insights

- **Framework Impact:** ONNX Runtime outperformed BitsAndBytes by ~69 % on identical hardware, proving implementation optimization is as critical as precision choice.
- **Model Size Threshold:** Quantization benefits become significant > 1 B parameters; smaller models can show overhead.
- **Hardware Sensitivity:** Older GPUs (T4) yield modest gains; A100/H100 architectures amplify speedups.
- **Co-Design Validation:** Performance depends on aligning quantization granularity with hardware tensor-core capabilities.

Conclusions

- Quantization combined with hardware-aware deployment delivers substantial efficiency gains without sacrificing accuracy.
- The project demonstrates that **INT4 quantization** is ideal for 1B+ models, while **ONNX Runtime INT8** is suited for smaller deployments.
- Results validate that **co-designing software quantization strategies with hardware architecture** can transform LLM deployment economics reducing speed, memory, and energy costs simultaneously.

Final Outcome: Professional-grade quantization framework analysis achieving ≈ **4.5× speedup**, **75 % memory savings**, and **14 % energy efficiency gain** with accuracy maintained within 3 %.