The FluxQuapacitors' presentation

TEAM MEMBER DETAILS

Project Lead (Architect):- Vedika Saravanan

GPU Acceleration PIC (Builder)-Lim Wee Keat and Gabriel Ortega

Quality Assurance PIC (Verifier):-Gabriel Ortega and Yash Singh

Technical Marketing PIC (Storyteller):Travis Martin

The LABS Problem

The Low Autocorrelation Binary Sequence (LABS) problem is a fundamental challenge in combinatorial optimization and signal processing. The objective is to find a binary sequence of length N that minimizes the side-lobes of its aperiodic autocorrelation function, effectively maximizing the 'merit factor.' Mathematically, this involves navigating a massive search space of 2^n possible configurations to find a state of minimum energy. Because the energy landscape is extremely 'rugged' and non-convex, classical algorithms often get trapped in local minima. Solving LABS is critical for developing high-precision radar, deep-space pulse compression, and robust cryptographic keys, making it a perfect candidate for the hybrid acceleration provided by CUDA-Q.

TOTAL ENERGY

$E = \sum_{k=1}^{N-1} C_k^2$

3

Lower energy indicates better autocorrelation properties.

This energy value defines the sequence's quality; reaching the global minimum is what enables high-precision signal detection in hardware.

Plan and Pivot

Original PRD strat:

For Phase 2, the strategy focuses on scaling and refining the QE-MTS workflow for real-world application on larger problem sizes. This involves optimizing the QAOA component by incorporating advanced parameter optimization techniques like COBYLA or SPSA to fine-tune the variational angles (gamma, beta) rather than relying on fixed values, thereby improving the quality of the initial candidate pool. Additionally, the team plans to migrate the simulation backend to NVIDIA's cuQuantum SDK to leverage GPU acceleration, enabling the simulation of deeper circuits and larger qubit counts beyond the current limits. Finally, the Memetic Tabu Search will be enhanced with adaptive tenure and diversification mechanisms to prevent premature convergence in the increasingly complex energy landscapes of high-N LABS sequences.

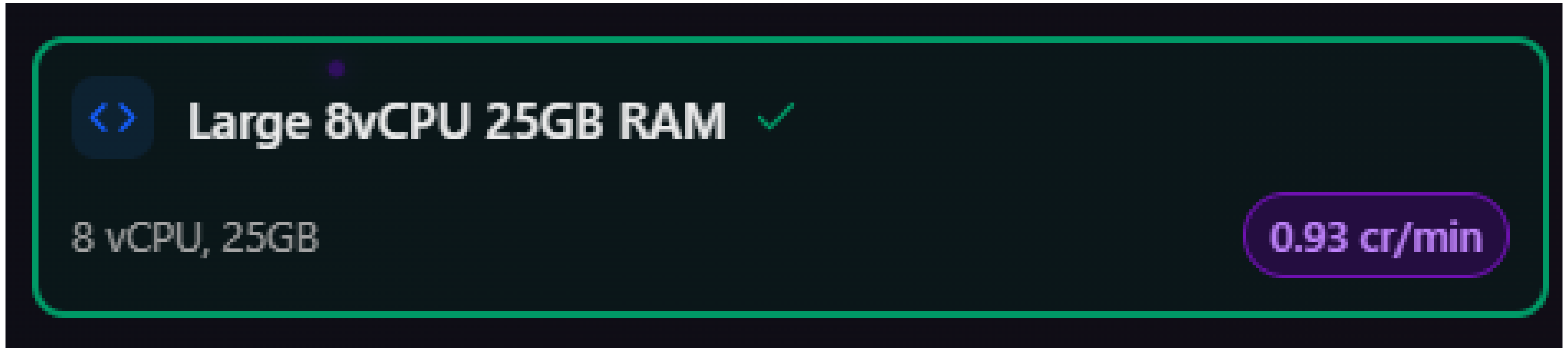
What changed?

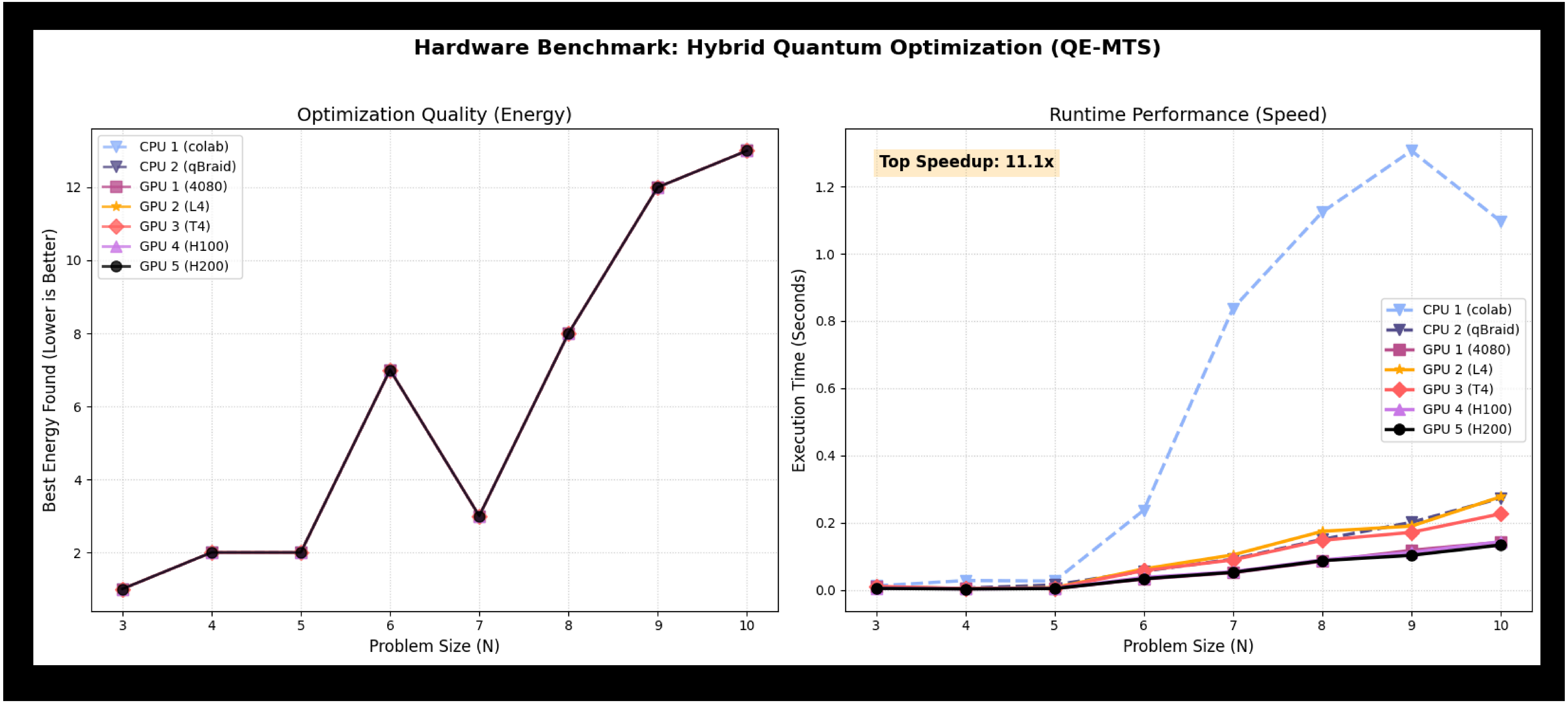
While our PRD strategy outlined a fully variational QAOA loop utilizing COBYLA or SPSA to dynamically train parameters, our final product utilized fixed ansatz parameters to prioritize rapid sampling and isolation of hardware benchmarking speed. Additionally, the planned adaptive tenure mechanism for the Tabu Search was streamlined to a fixed-tenure approach to reduce classical overhead during the short hackathon timeframe. Finally, the implementation required an unforeseen manual decomposition of R_{ZZ} gates into CNOT-R_Z-CNOT chains to ensure full compatibility with the specific gate set available in the CUDA-Q kernel.

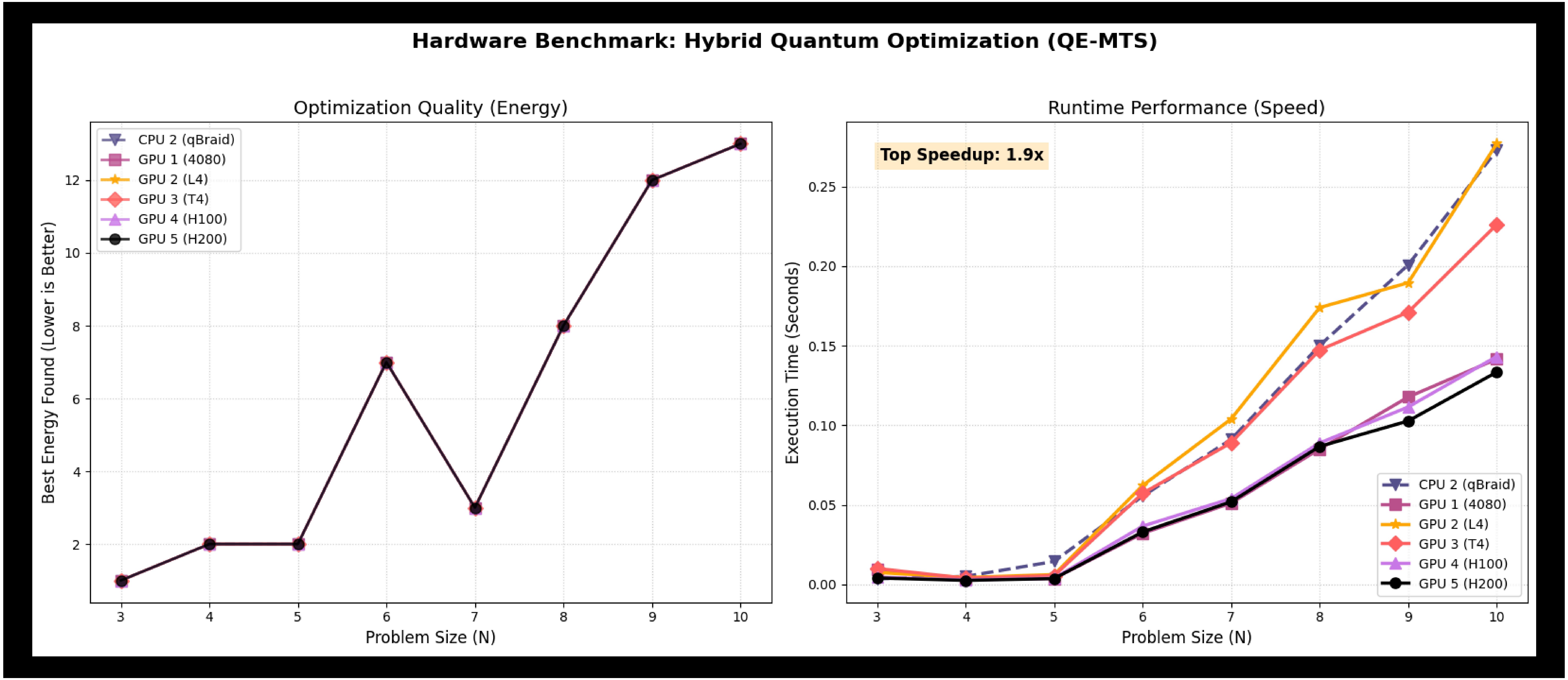
Results!!!

We compared CPU vs GPU performance through measuring both Solution Quality and Runtime Comparison, using one of our teammate's local GPU (an RTX 4080) as a control to not only compare NVIDIA Brev GPUs against qBraid's Large 8vCPU, but also against a physical GPU (the RTX 4080).

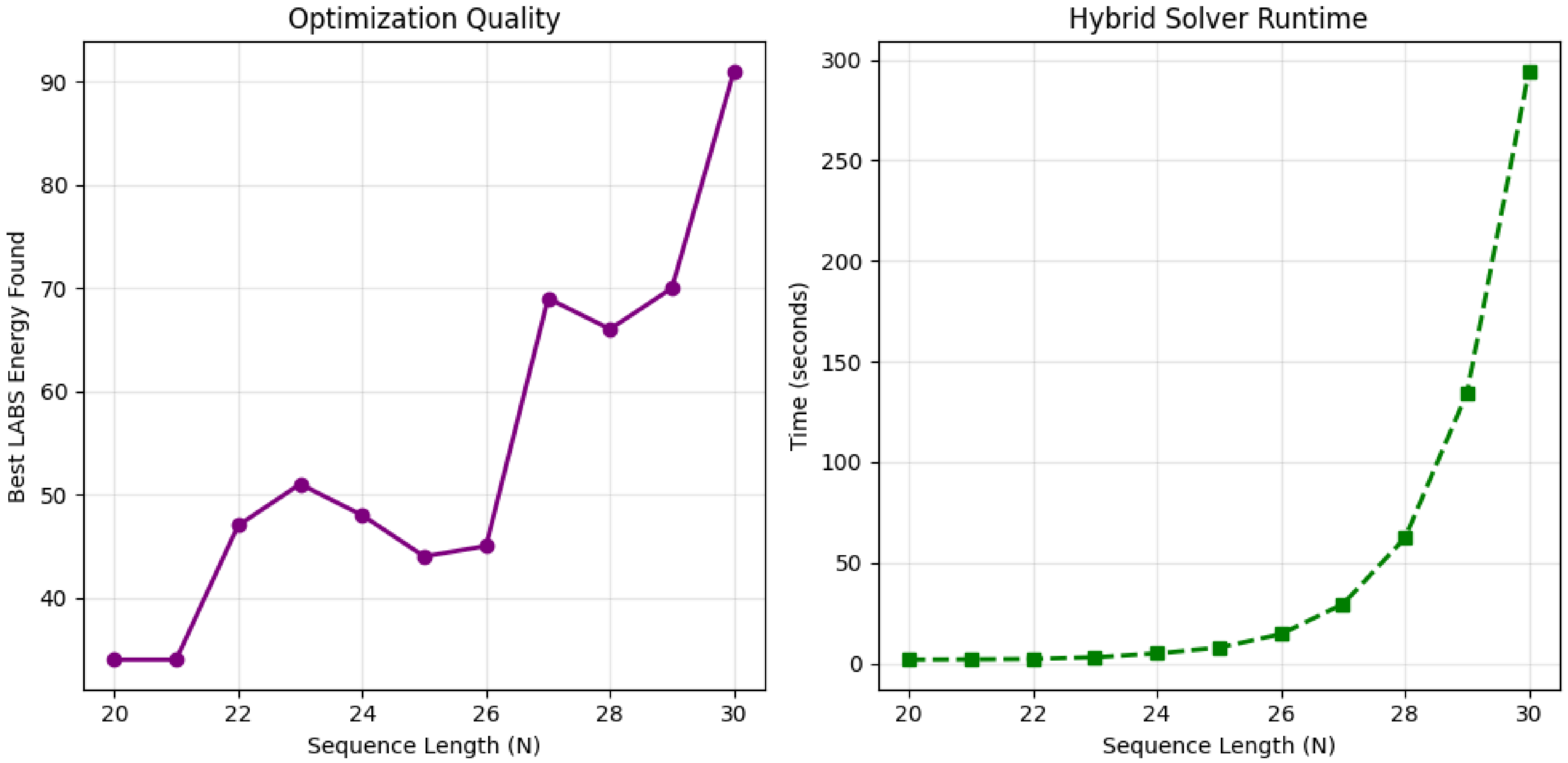
We decided to keep the solution graph on the side to ensure the only difference is purely hardware, no shady tricks under the hood.



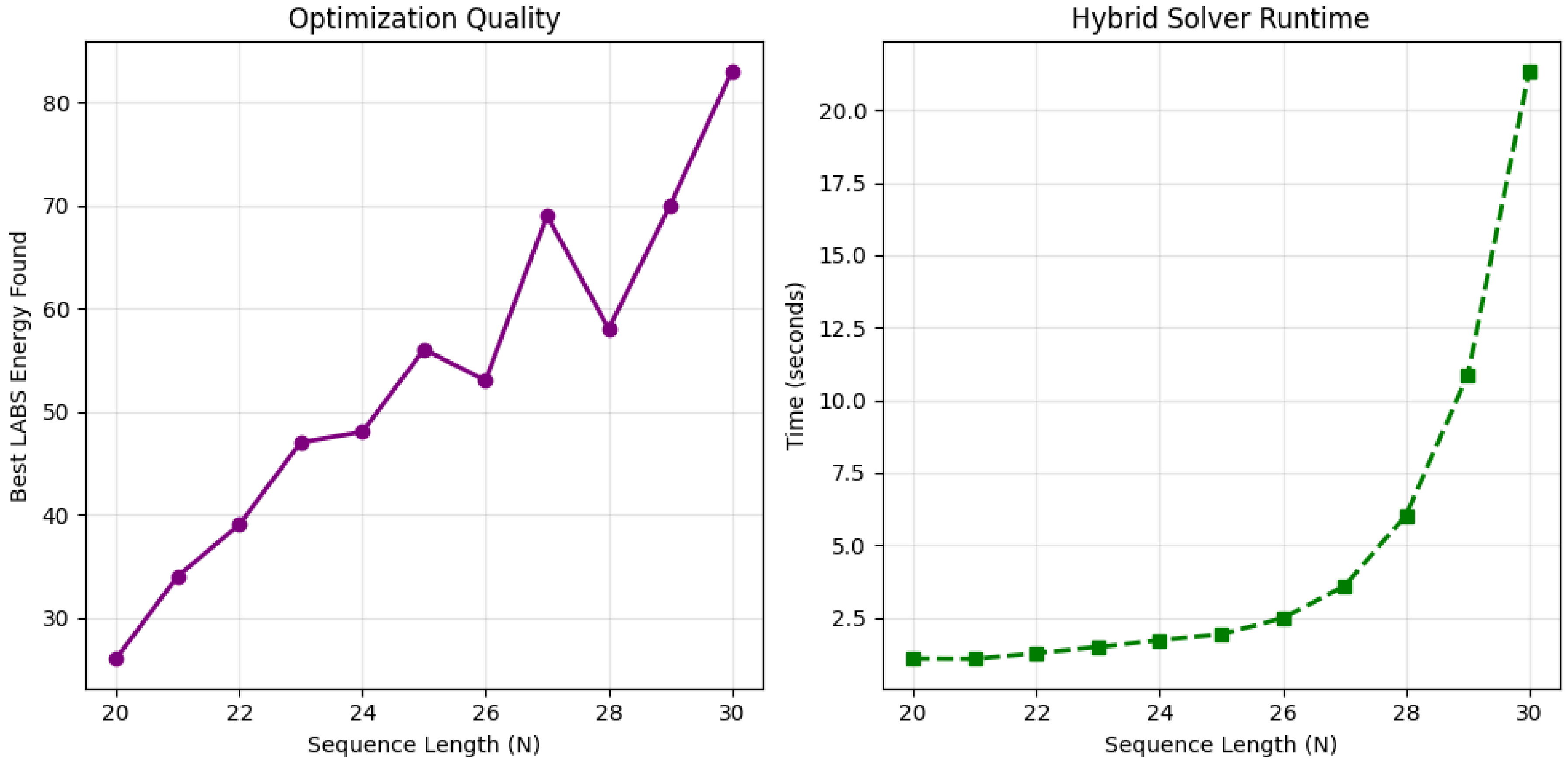

Hardware Benchmark: Hybrid Quantum Optimization (QE-MTS)
Optimization Quality (Energy)
Best Energy Found (Lower is Better)
Problem Size (N)
CPU 1 (colab)
CPU 2 (qBraid)
GPU 1 (4080)
GPU 2 (L4)
GPU 3 (T4)
GPU 4 (H100)
GPU 5 (H200)
Runtime Performance (Speed)
Top Speedup: 11.1x
Execution Time (Seconds)
Problem Size (N)



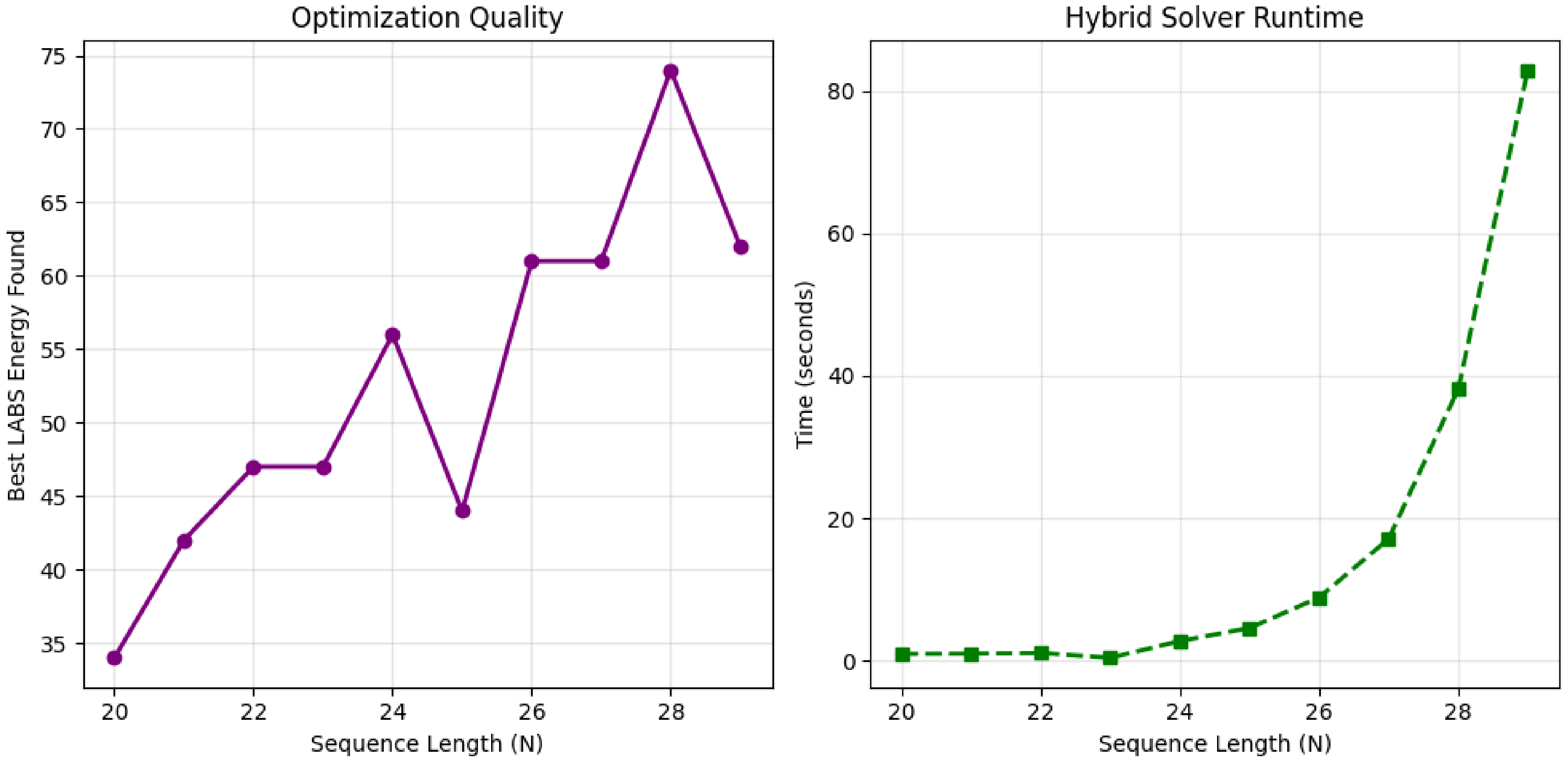
Hardware Benchmark: Hybrid Quantum Optimization (QE-MTS)
Optimization Quality (Energy)
Best Energy Found (Lower is Better)
Problem Size (N)
CPU 2 (qBraid)
GPU 1 (4080)
GPU 2 (L4)
GPU 3 (T4)
GPU 4 (H100)
GPU 5 (H200)
Runtime Performance (Speed)
Top Speedup: 1.9x
Execution Time (Seconds)
Problem Size (N)



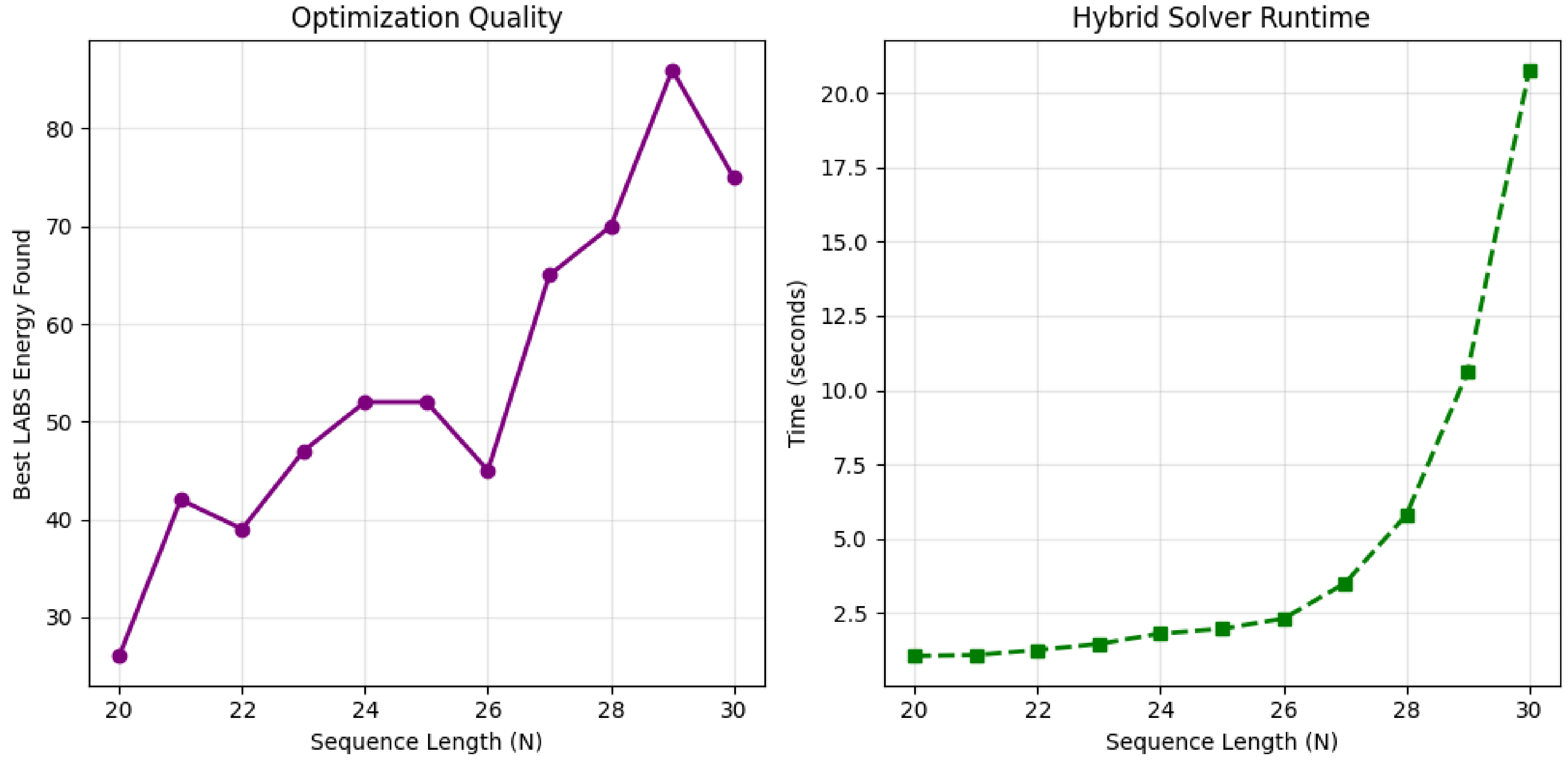
L4



H100



RTX 4080



H200

Summary

Key Observations & Takeaways

- **Algorithm quality alone does not determine performance at scale**
- **Hardware characteristics become dominant as problem size increases**
- **Small scale (n = 3 → 10):**
  - **No meaningful difference between CPU and GPU results**
  - **Output is stable across hardware types**
- **Larger scale (n = 20 → 30):**
  - **Significant variation in outputs for the same algorithm**
  - **Differences driven primarily by GPU quality and architecture**
- **Runtime performance is not a reliable predictor of solution quality**
- **Example:**
  - **RTX 4080 vs H200 show similar runtimes**
  - **Yet produce different energy outputs**
- **Implication:**
  - **GPU choice impacts numerical outcomes, not just speed**
  - **Hardware-aware evaluation is critical for large-scale workloads**