Large-scale Quantum Circuit Emulation with MIMIQ

Guido Masella
CTO & Co-founder, QPerfect SAS

Overview

- Introduction to emulation: MPS and MIMIQ.
- Real-World Application.
- Pushing the Limits: QEC.
- From simulation to execution.

Emulating quantum systems

Why emulation?

Design, benchmark and optimize quantum algorihthms and quantum hardware

For a system of N qubits

$$|\psi\rangle \longrightarrow 2^N$$ **Complex Numbers**

On a classical computer

- State Vector Emulators

  + High accuracy, many operations

  - Low number of Qubits

- Clifford Emulators

  + High number of qubits

  - Restricted gate set

- Matrix Product States

  + High number of qubits

  - Expressivity (e.g. low entanglement)

Tensor Networks idea

Useful resource: https://tensornetworks.org

Compress entanglement between subsystems

$$|\psi\rangle = \sum_{i=0}^{\chi} \boxed{|A_i\rangle} - \boxed{|B_i\rangle}$$

Reduce the number of terms in the sum

$$|\psi\rangle = \sum_{\substack{n_1, n_2, \\ n_3, n_4}} \underbrace{\sum_{i,j,k} A_i^{n_1} A_{ij}^{n_2} A_{jk}^{n_3} A_k^{n_4}}_{\text{Matrix Product}} |n_1 n_2 n_3 n_4\rangle$$

Restrict size of A matrices
→ Compressed representation

Singular Value Decomposition

MPS can be constructed using Singular Value Decompositions

$$A = USV^\dagger$$

Diagonal
Singular Values

Truncation:
Keep only largest singular values

Strasbourg Cathedral

Original | 50% compressed | 95% compressed

MIMIQ

Quantum Circuit Emulator
by QPerfect

- Optimized for speed, scale and accuracy
- Easy to use
- Professional features for cutting-edge research

Benchmarking Large Scale Emulators

Leonteva et al., ACM Transactions in Quantum Computing (2025)

1) Algorithm selection

- 28 quantum algorithms
- 14 scalable up to 1024 qubits
- OpenQASM format
- Transpiled to a minimal gate set (u, cx)

2) Emulator selection

CPU only (for now)

3) Performance metrics

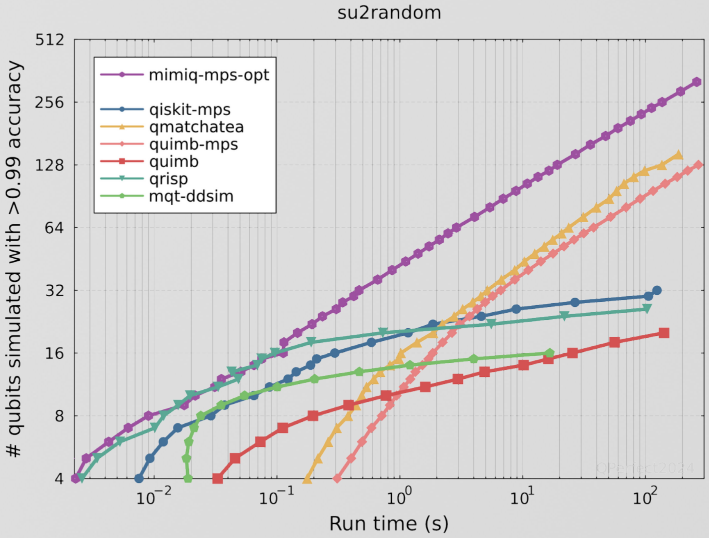
Scale : maximum number of qubits with fidelity >0.99 within 300 seconds

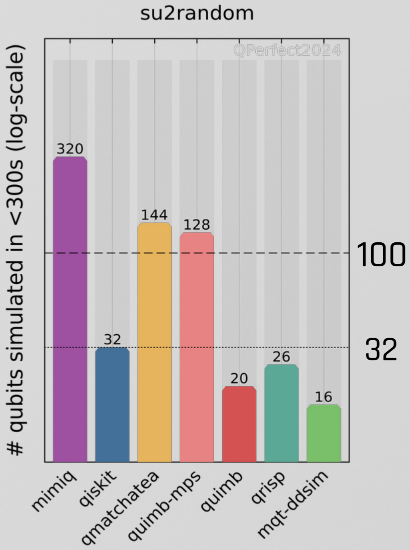
Accuracy: mirror circuit fidelity *[Nature Phys. 18,75-79 (2022)]*

Speed : minimum time to execute the circuit, including import and sampling

Benchmarking Large Scale Emulators

Leonteva et al., ACM Transactions in Quantum Computing (2025)





Key takeaways

- MPS Algorithms outperform others
- Most benchmarking algorithm scale polynomially for MPS

Elo ranking system: measure the performance of quantum computing systems across benchmarks

Algorithm	mimiq-mps	qiskit-mps	qmatchatea	quimb-mps	quimb	qrisp	mqt-dds
Elo Rating	1537 ± 17	1437 ± 41	1298 ± 43	1178 ± 28	981 ± 39	930 ± 52	989 ± 55

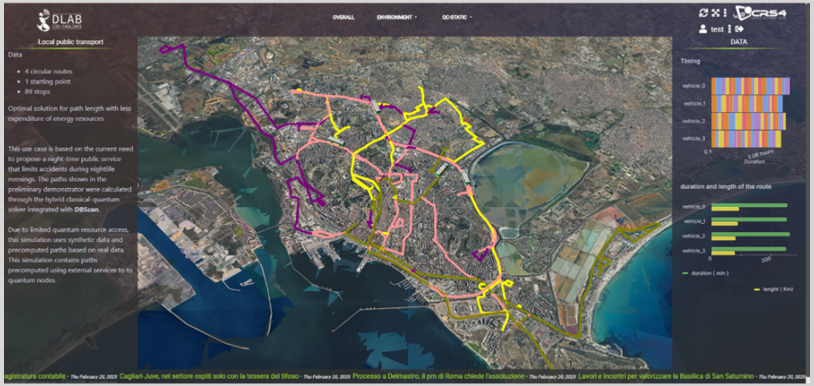
Use Case: Fleet Routing

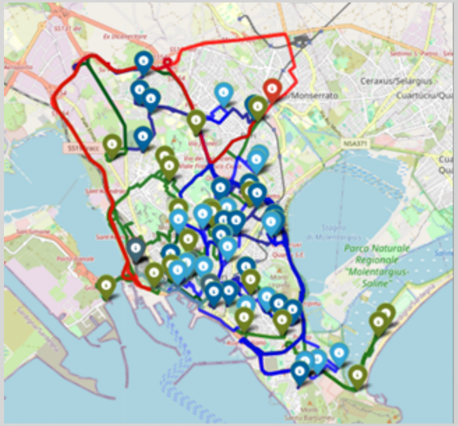
As part of the Cagliari Digital Lab project, CRS4 is conducting R&D into **fleet routing**. NISQ algorithms solve the MD-VRP (Multi-Depot Vehicle Routing Problem) and its variants, taking into account **real-world constraints** and using **real traffic data**.

The digital QC approach involves decomposing the problem into two stages: first, clustering customers to depots; then, solving VRP or CVRP instances on a QPU.

MIMIQ is used for **extensive testing across a large number of customers** using **stratified datasets**, enabling statistical analysis to identify conditions in which quantum solutions are more effective than classical approaches.

The solvers will be integrated in the smart city operational control platform





Use Case: Secure Communications

BTQ is developing quantum algorithms for securing communications and digital transactions.

The application exploits quantum computations and quantum mechanical principles to secure communications without the need of a quantum channel.

MIMIQ is deployed to simulate these complex circuits, allowing BTQ to **verify the security guarantee**s and **optimize the implementation strategy** to minimize the resources required for future digital quantum hardware.

Real-world applications
Digital Identity, Financial Transactions, Smart Contracts, Secure Communications, Decentralized Finance, Digital Right Management

Use Case: Hybrid Compute

Quobly is building a developer community to design and test applications before physical hardware is widespread.

The MIMIQ platform and its engines power QLEO, Quobly's user-facing emulator, and are integrated with the **NVIDIA CUDA-Q** stack.

Users leverage the **GPU-accelerated backend** to debug and optimize complex logical circuits in minutes rather than days.

Developers write standard **C++ and Python code** that runs on the emulator today and on QPUs tomorrow.

QLEO is available on OVHcloud

MIMIQ for Fault Tolerance

Collaboration with QuEra — Putting Quantum to Work

Today QC have errors~0.1%

Quantinuum, arXiv:2406.02501

QuEra, Nature 622, 268-272 (2023)

→ QEC

Needs 0.000000001%

Factoring 2048 bit RSA

Waintal, PNAS 121 (1) e2313269120 (2023)

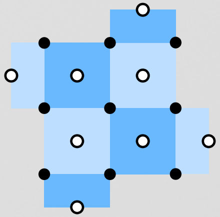
- QEC code?
- Logical gate implementation?
- Hardware implementation?
- Decoder?
- ...

Can we simulate large scale logical quantum algorithms?

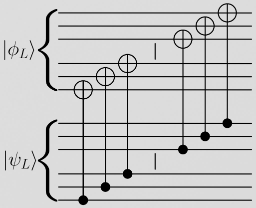
Magic State Distillation

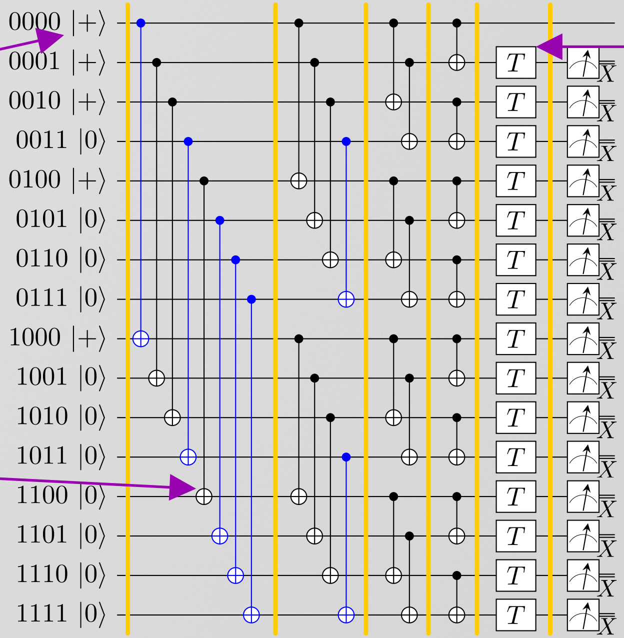
Collaboration with QuEra — Putting Quantum to Work

Rotated Surface Code



Logical CNOT



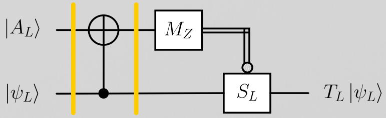


[Beverland, Kubica & Svore, PRX Quantum (2021)]

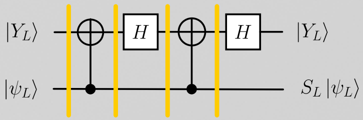
~~16 qubits, ~50 gates?~~

289 qubits, >30000 gates

Logical T



Logical S

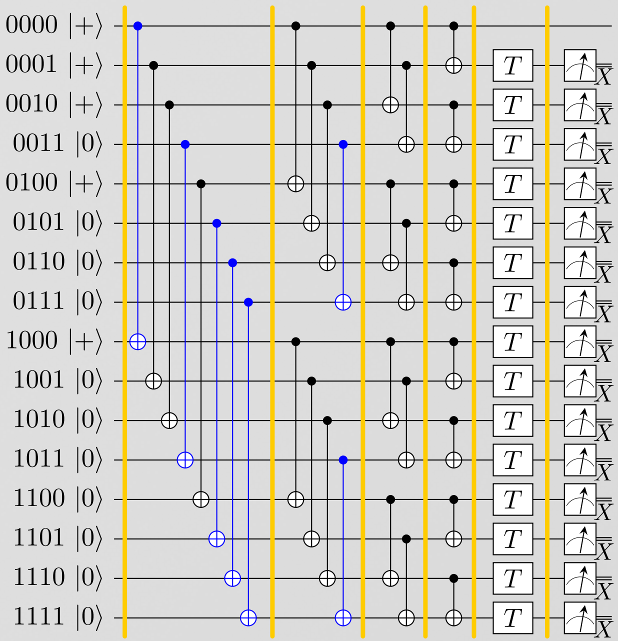


Rounds of QEC

Experimental Demonstration of Logical Magic State Distillation
Nature 645, 620–625 (2025)
by Harvard and QuEra

MSD Optimizations

Collaboration with QuEra



289 qubits, >30000 gates

- Ordering of physical qubits
  ×100 speedup
- Ordering of Physical Gates
  ×2 - ×10 speedup
- Ancillas and Ordering of Logical Gates
  ×2 - ×4 speedup
- Gate compression
- Tuning of MPS parameters

Runs with fidelity 1.0 on a single CPU

Qubits	Best
85	0.9 s
153	25 s
289	**11 min**

Solving the same hard problems

Optimizing Simulators		Optimizing Hardware Execution
Qubit Ordering	↔	Qubit Mapping
Gate Ordering	↔	Gate Mapping
Ancilla and logical gate placement	↔	Architecture choice and resource allocation

BTQ

Cryptography

aQCess

Neutral Atoms

Quantum Logic Unit QLU™

Bridging quantum advantage regime applications to early fault-tolerant quantum computers

- Compiling algorithms to efficient execution on neutral atom QPUs
- Incorporating industry-best gates for neutral atoms
- Application- and hardware-specific error correction with minimal overheads
- Cutting quantum resource requirements by orders of magnitude

Thank you for your attention!

QPerfect

Contact: **guido.masella@qperfect.io**

Follow us