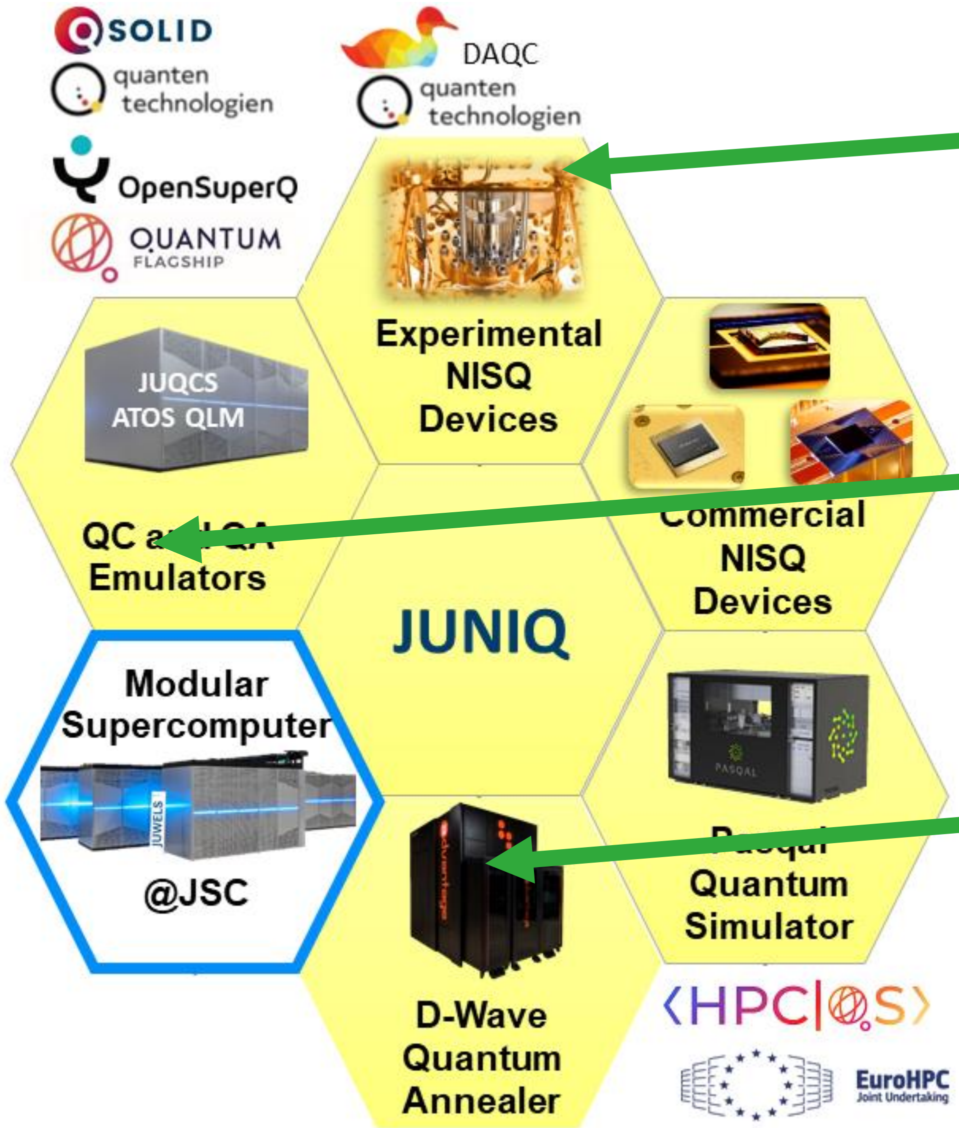
GPU-BASED QUANTUM COMPUTER SIMULATORS BEYOND JUQCS

JUNE 21, 2022 I DENNIS WILLSCH

CONTENTS



1. **JUQMES**
   Quantum **M**aster **E**quation **S**imulator
2. **JUQFAS**
   Quantum **F**actoring **A**lgorithm **S**imulator
3. **JUQAS**
   Quantum **A**nnealing **S**imulator

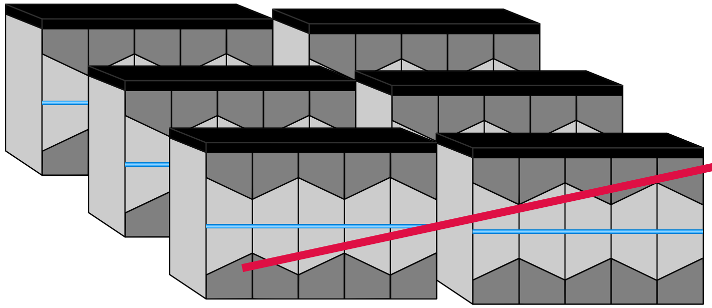
JUQ***: SIMULATING QUANTUM COMPUTERS ON GPUS

The quantum state

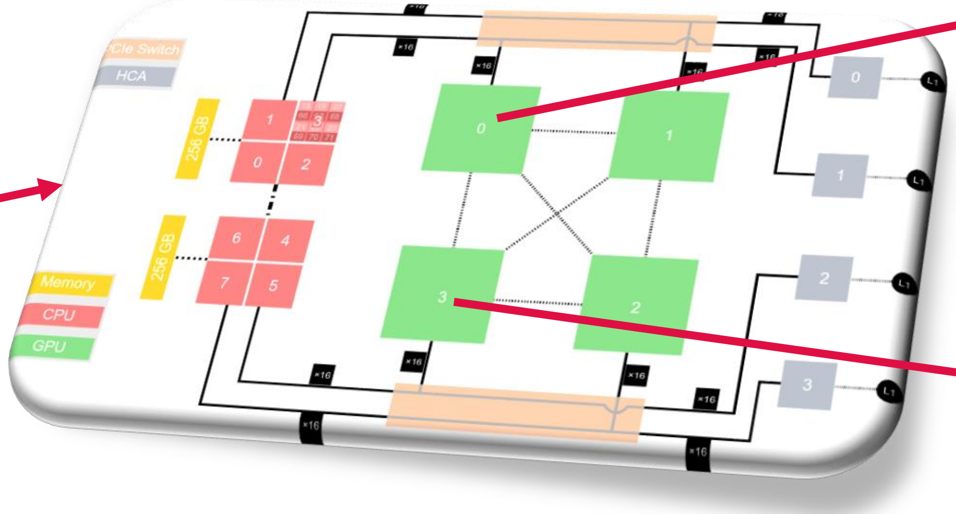
- Each simulation step transforms a quantum state $|\psi\rangle = (\psi_{\cdots q_3 q_2 q_1 q_0})$ or $\rho = (\rho_{k_0 m_0 k_1 m_1 \cdots})$
- Distribute all complex numbers on the GPUs (NVIDIA A100: ≤ 40GB per GPU)

For 40 qubits: 2^{40} $\psi'\mathrm{s}$ = 16 TiB complex numbers = 16 GiB per GPU with 4*256 GPUs

JUWELS Booster
936 compute nodes



4 A100 GPUs per node



GPU rank 0 = 0b0000000000:

$$\begin{pmatrix} \psi_{0000000000\,0\cdots0} \\ \vdots \\ \psi_{0000000000\,1\cdots1} \end{pmatrix}$$

GPU rank 3 = 0b0000000011:

$$\begin{pmatrix} \psi_{0000000011\,0\cdots0} \\ \vdots \\ \psi_{0000000011\,1\cdots1} \end{pmatrix}$$

JUQ***: SIMULATING QUANTUM COMPUTERS ON GPUS

Fundamental tensor operations

- The fundamental operations on the quantum state are given by

$$\psi_{\cdots q_3 q_2 q_1 q_0} \leftarrow \sum_{q_2'} H_{q_2 q_2'} \psi_{\cdots q_3 q_2' q_1 q_0}$$

$$\psi_{\cdots q_3 q_2 q_1 q_0} \leftarrow \sum_{q_2' q_0'} U_{q_2 q_2' q_0 q_0'} \psi_{\cdots q_3 q_2' q_1 q_0'}$$

$$\rho_{k_0 m_0 k_1 m_1 \cdots} \leftarrow \sum_{m_0'} V_{m_0 m_0'} \rho_{k_0 m_0' k_1 m_1 \cdots}$$

$$\rho_{k_0 m_0 k_1 m_1 \cdots} \leftarrow \sum_{k_0' k_1'} W_{k_0 k_0' k_1 k_1'} \rho_{k_0' m_0 k_1' m_1 \cdots}$$

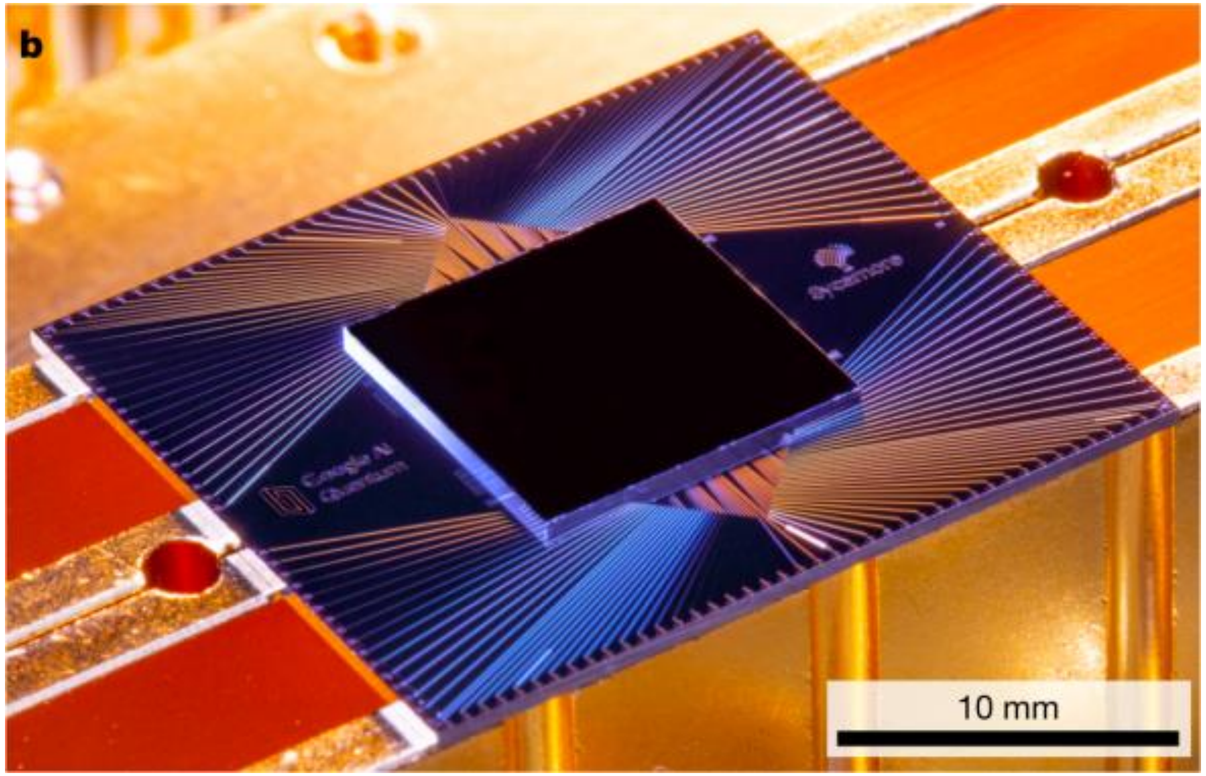
- In-place double complex SPMV (sparse matrix-vector) updates
- If numbers are on different GPUs → **MPI Communication**

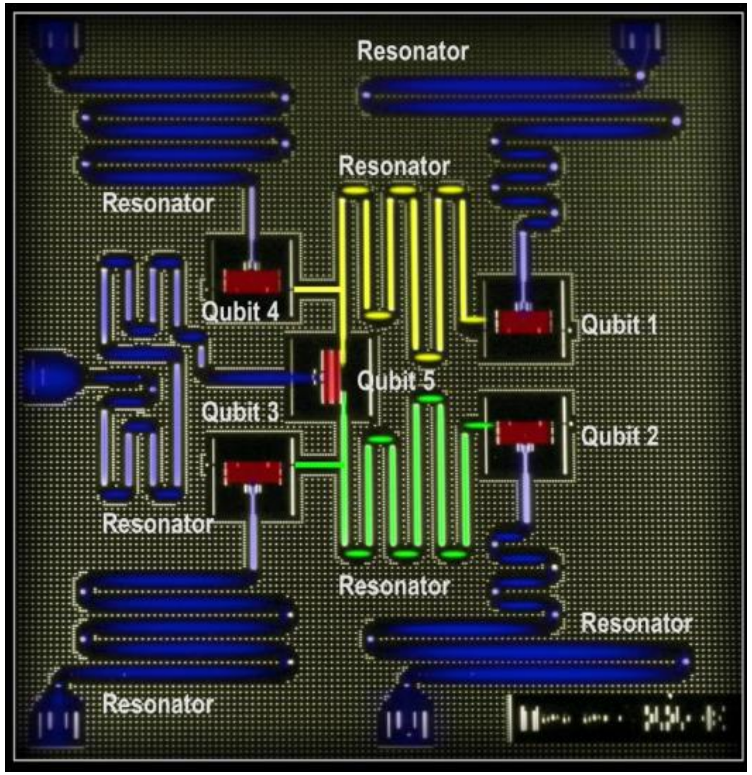
JUQMES: QUANTUM MASTER EQUATION SIMULATOR

Simulating physical realizations of quantum computers on GPUs

CUDA+OpenACC MPI C++

https://jugit.fz-juelich.de/qip/juqmes





JUQMES: QUANTUM MASTER EQUATION SIMULATOR

Simulating physical realizations of quantum computers on GPUs

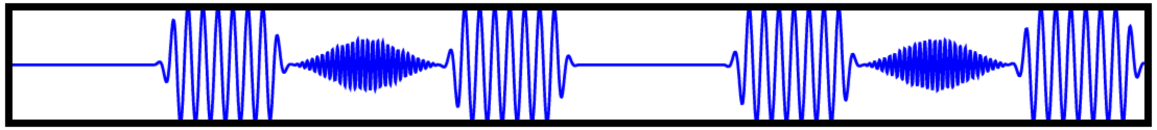
CUDA+OpenACC MPI C++

https://jugit.fz-juelich.de/qip/juqmes

- Physical realization: Solve Schrödinger / Master Equation

$$\frac{\partial}{\partial t}|\psi\rangle = -iH|\psi\rangle \quad \text{or} \quad \frac{\partial}{\partial t}\rho = -i[H,\rho] + \mathcal{D}[\rho] = \mathcal{L}[\rho]$$

Time-dependent pulse to implement quantum gates

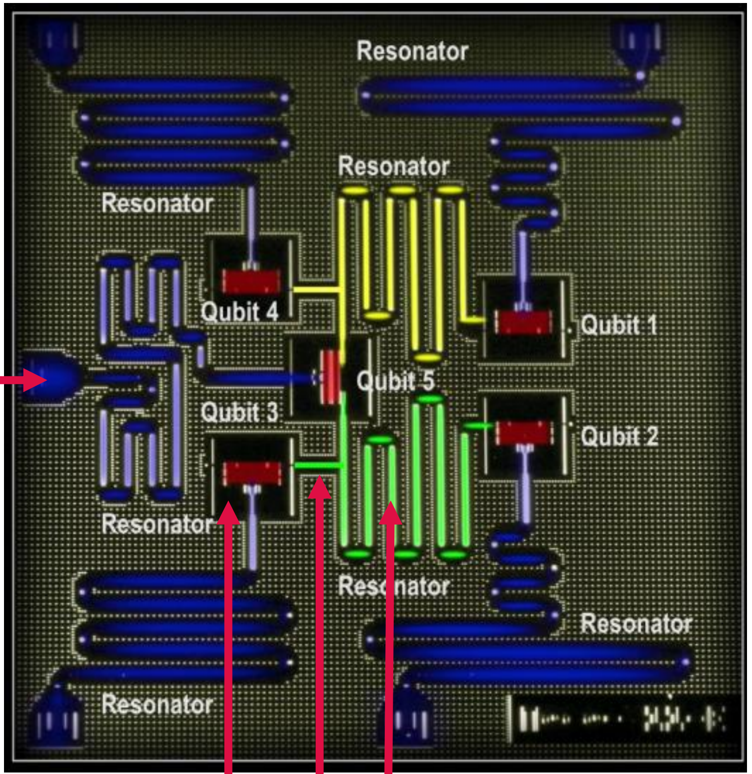


$$H = \boxed{\underbrace{4E_C(n - n_g(t))^2 - E_J \cos\varphi}_{\sum_{m\geq 0} E_m|m\rangle\langle m|}} + \boxed{\underbrace{\Omega a^\dagger a}_{\sum_{k\geq 0} k\Omega|k\rangle\langle k|}} + \boxed{\lambda G n(a + a^\dagger)}$$

Transmon

Resonator

Coupling



JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

JUQMES: QUANTUM MASTER EQUATION SIMULATOR

Simulating physical realizations of quantum computers on GPUs

CUDA+OpenACC MPI C++

https://jugit.fz-juelich.de/qip/juqmes

- Physical realization: Solve Schrödinger / Master Equation

$$\frac{\partial}{\partial t}|\psi\rangle = -iH|\psi\rangle \quad \text{or} \quad \frac{\partial}{\partial t}\rho = -i[H,\rho] + \mathcal{D}[\rho] = \mathcal{L}[\rho]$$

- Similar SPMV updates like JUQCS **but:**
  - Many updates per time step

    $$\rho(t+\tau) = e^{\mathcal{L}(t+\tau/2)}\rho(t)$$

  - More than 2 states per subsystem
    - JUQCS: $|\psi\rangle = (\psi_{q_3q_2q_1q_0})$
    - JUQMES: $\rho = (\rho_{k_0m_0k_1m_1})$
  - More complicated sparse matrices (sin, sinh, cos, cosh, exp, ...)
- Very computation-intensive (memory “only” 2 GiB)

$0 \leq k_0, k_1 < 4 \text{ to } 1000$

$0 \leq m_0, m_1 < 4 \text{ to } 12$

Before:

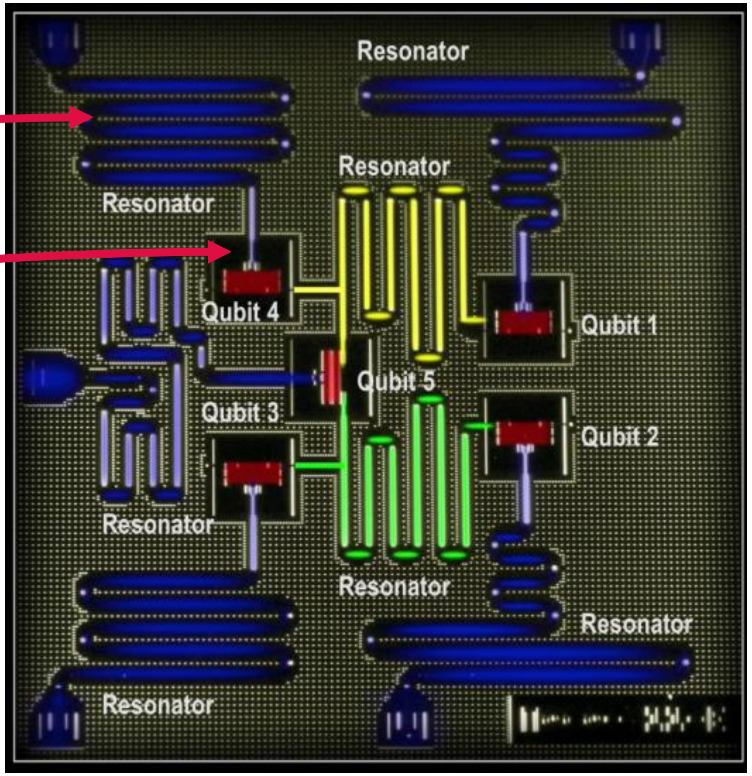
$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Now:

$$\begin{pmatrix} \tilde{c}_{k_0m_0} & -i\tilde{s}_{k_0m_0} \\ -i\tilde{s}_{k_0m_0} & \tilde{c}_{k_0m_0} \end{pmatrix}$$

$$\tilde{c}_{k_0m_0} = \cos(\tau\sqrt{k_0+1}(G\lambda_{m_0} + \varepsilon(\tilde{t})))$$

$$\tilde{s}_{k_0m_0} = \sin(\tau\sqrt{k_0+1}(G\lambda_{m_0} + \varepsilon(\tilde{t})))$$



JUQMES: QUANTUM MASTER EQUATION SIMULATOR

Simulating physical realizations of quantum computers on GPUs

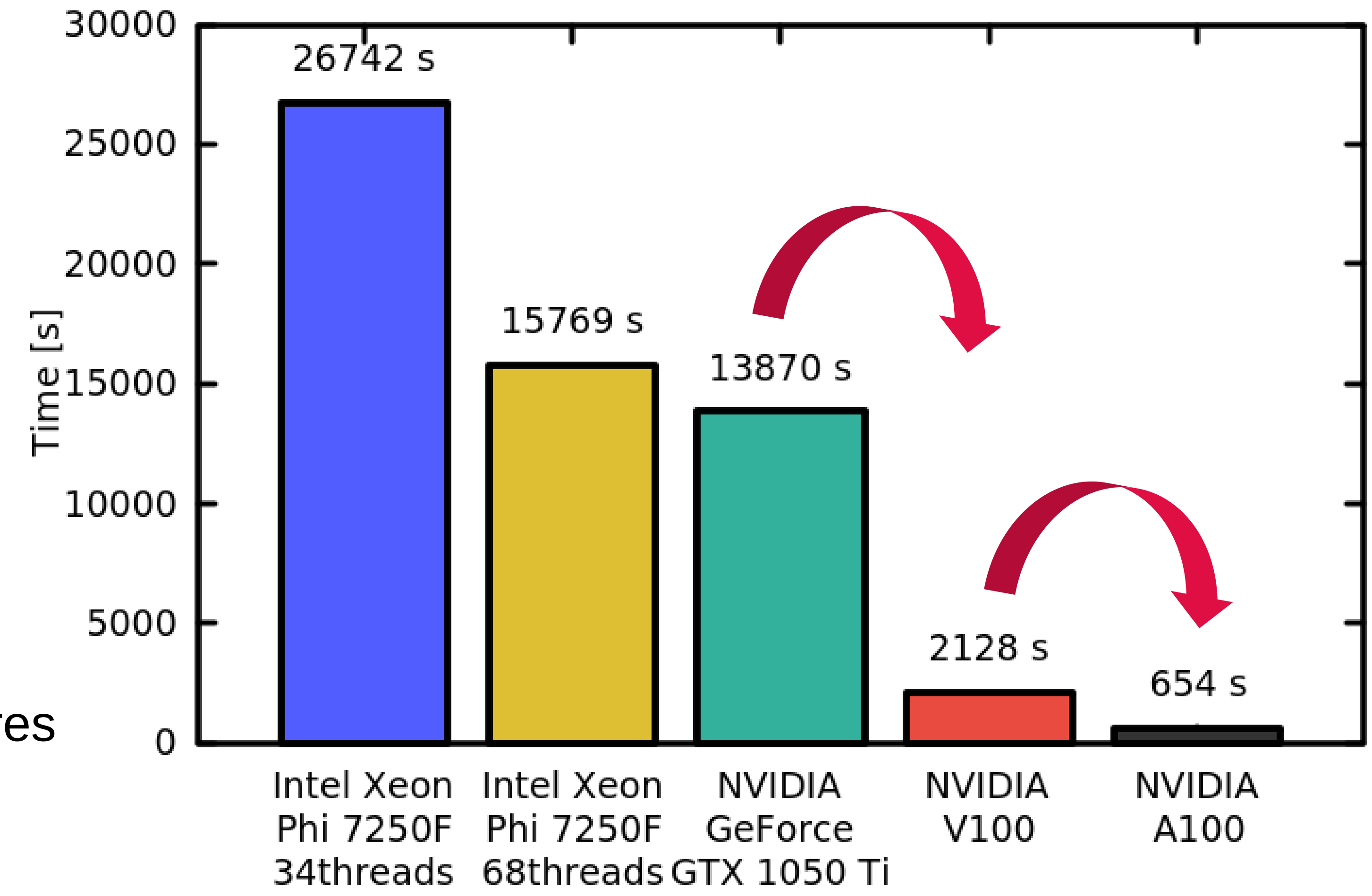
CUDA+OpenACC MPI C++

Realistic system with **M=12** and **K=400**

For each k_0, k_1, m_0

$$\rho_{k_0 m_0 k_1 m_1} \leftarrow \sum_{m_0'} V_{m_0 m_0'} \, \rho_{k_0 m_0' k_1 m_1}$$

12-component updates

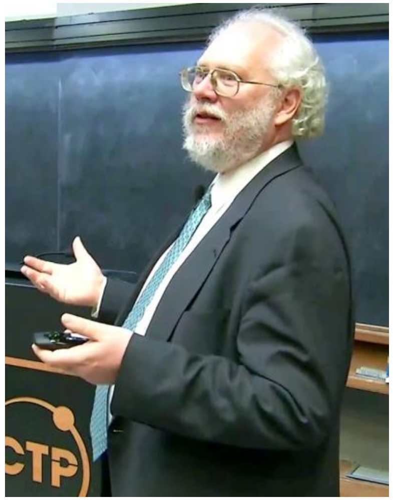


→ Significant speedup through tensor cores

→ Further speedup from **V100** to **A100**

JUQFAS: QUANTUM FACTORING ALGORITHM SIMULATOR

The factoring problem



- Given a composite integer N, find a nontrivial factor 1 < p < N
- For an L-bit **semiprime** N = p*q, find a prime factor p
  - We don’t know a classical algorithm that runs in polynomial time (in L)
  - A lot of **Internet security** is based on this “hardness”

    → Public key cryptosystems like RSA used in TLS, SSH, …
- **Shor’s algorithm** for a gate-based quantum computer can find p in polynomial time:
  - Randomly select a coprime to N
  - With high probability, Shor’s algorithm yields the **order** of a

    → the smallest r with $a^r \bmod N = 1$
  - With high probability, the **order** r yields a **factor** p of N

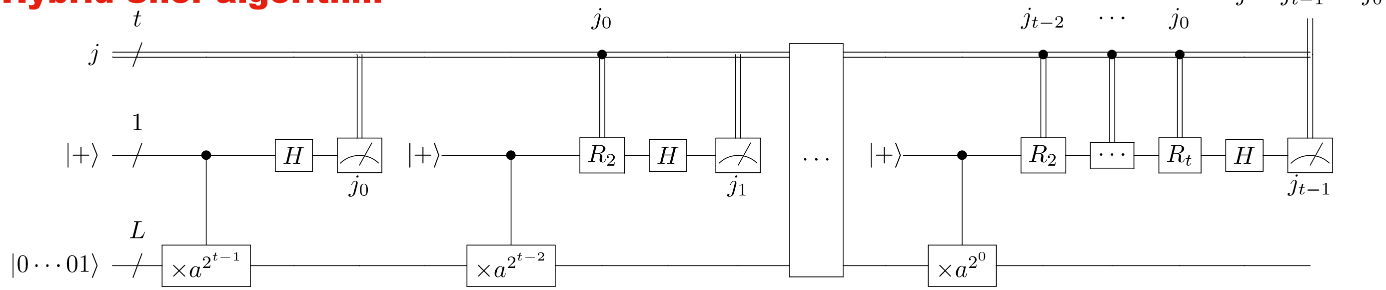
JUQFAS: QUANTUM FACTORING ALGORITHM SIMULATOR

qubits n	semiprime N	t
5	15	8
6	21	9
$\vdots$	$\vdots$	
40	549 755 813 699	78
40	549 755 813 701	78
40	549 755 813 713	78
$\vdots$	$\vdots$	

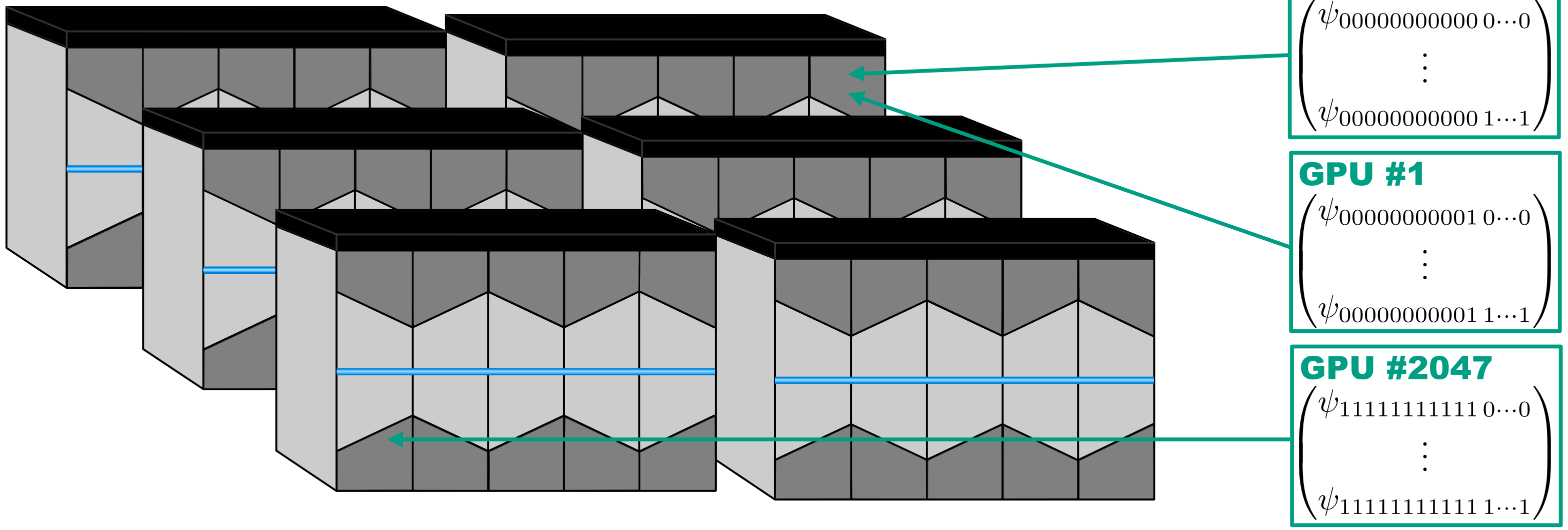
select problem

$$N = 549\,755\,813\,701$$
$$a = 439\,314\,602\,906$$

Hybrid Shor algorithm



JUQFAS simulating a 40-qubit quantum computer on 2048 GPUs



produces bitstring with t bits

$$j = 1111001101111101001111100011011000100000000111011111101110110110110011100010011_2$$
$$= 287\,448\,630\,931\,475\,209\,611\,027$$

extract order

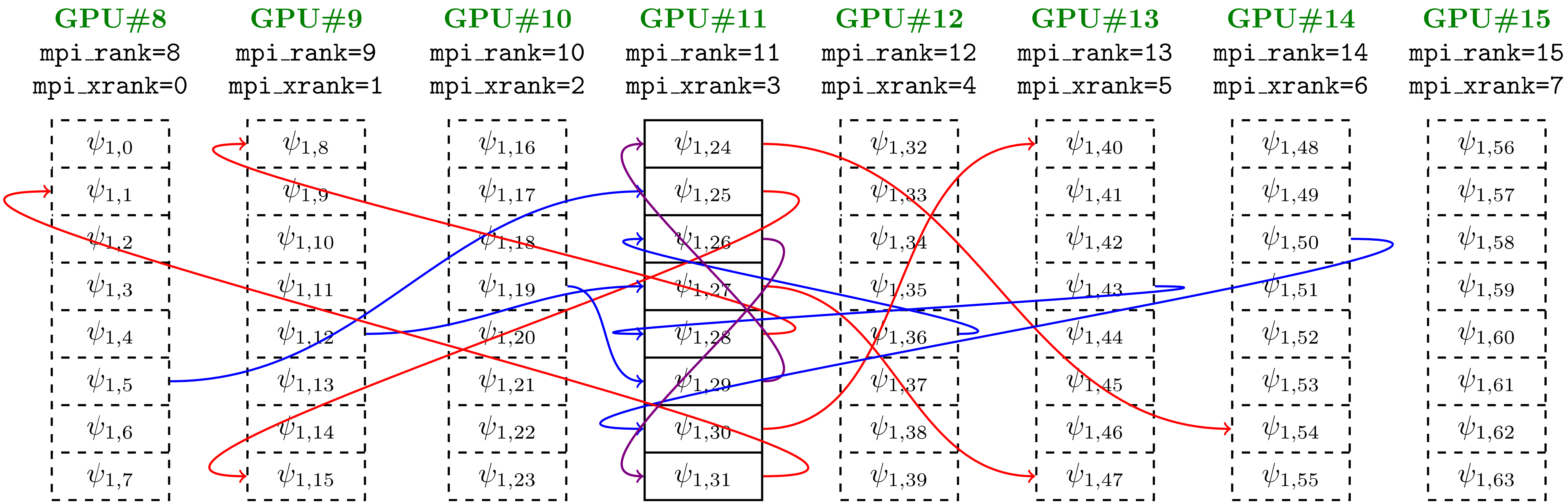
$$s/r \approx j/2^t$$
$$r = 4\,581\,286\,080$$

find factors

$$549\,755\,813\,701 = \underset{\gcd(a^{\lfloor r/2\rfloor}+1,N)}{712\,321} \times \underset{\gcd(a^{\lfloor r/2\rfloor}-1,N)}{771\,781}$$

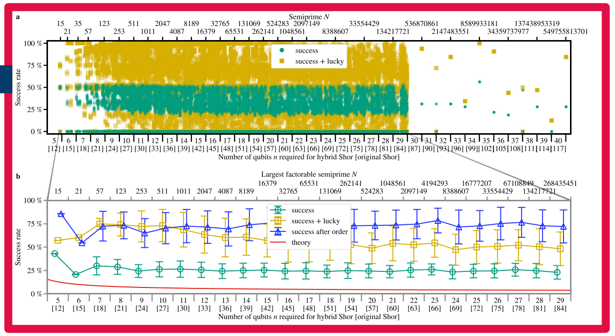
JUQFAS: QUANTUM FACTORING ALGORITHM SIMULATOR

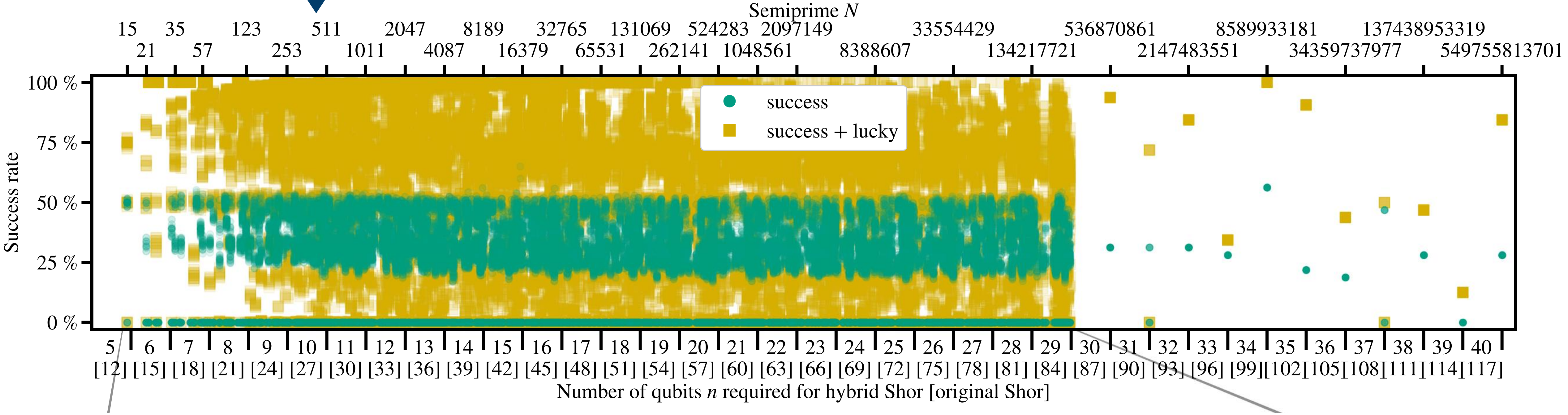
Complicated MPI Communication Scheme



JUQFAS: QUANTUM FACTORING ALGORITHM SIMULATOR

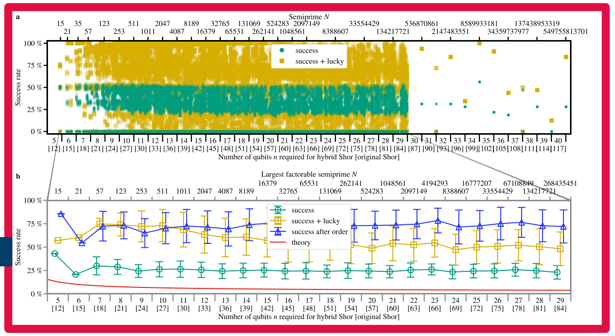
Results

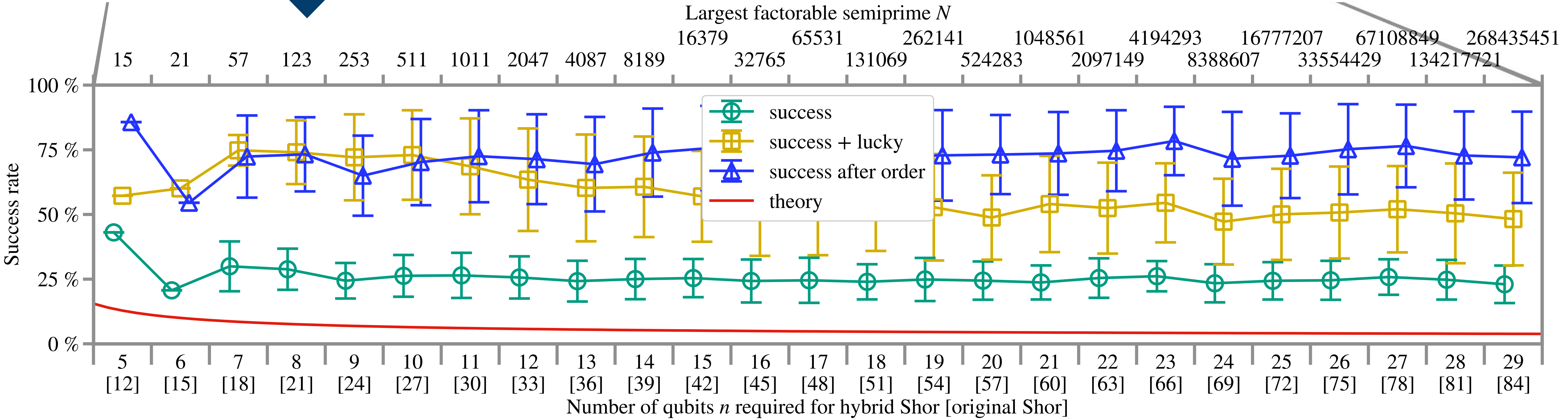




JUQFAS: QUANTUM FACTORING ALGORITHM SIMULATOR

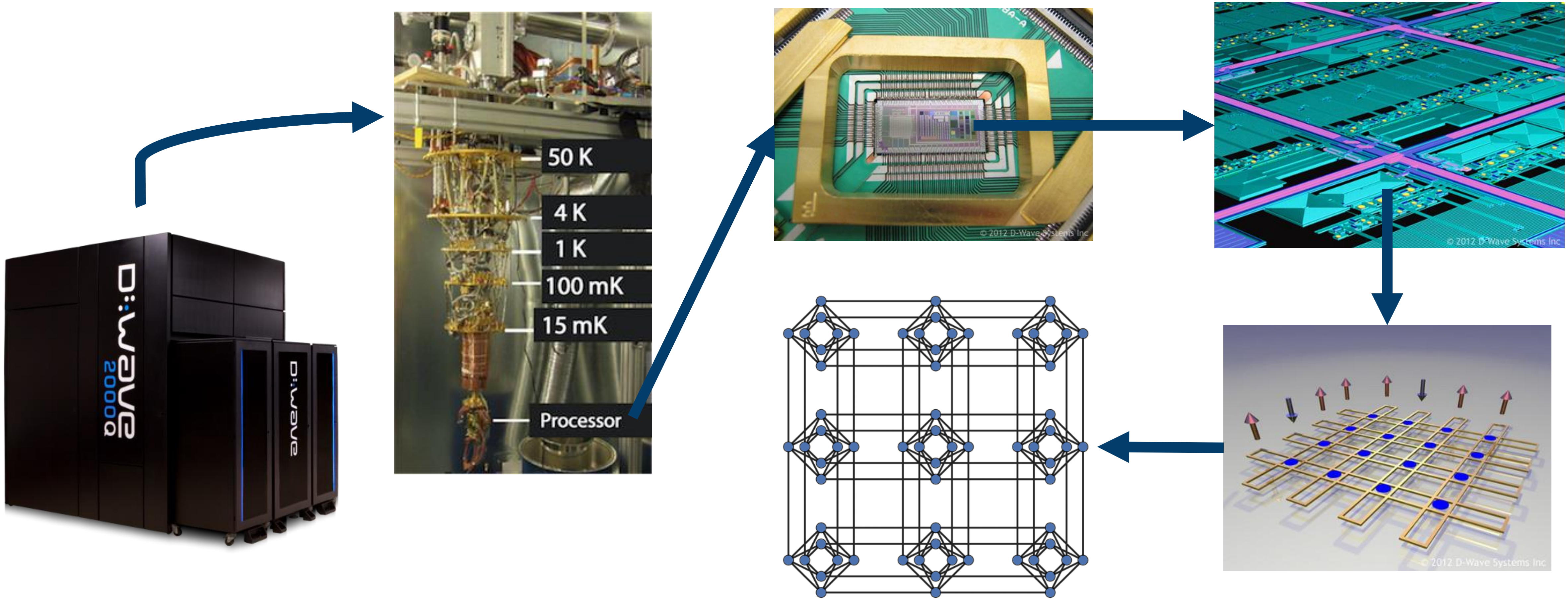
Results





JUQAS: QUANTUM ANNEALING SIMULATOR

What do quantum annealers look like?



JUQAS: QUANTUM ANNEALING SIMULATOR

What do quantum annealers do?

Quadratic Unconstrained Binary Optimization

"bias" (real number)

$$\textbf{QUBO:} \quad \min_{x_i=0,1} \left(\sum_i a_i x_i + \sum_{i<j} b_{ij} x_i x_j \right)$$

problem variables (binary)

"coupler" (real number)

Why might this be interesting?
→ discrete optimization is **hard** (NP-hard)
→ produces **many** solutions
→ very **low** energy consumption

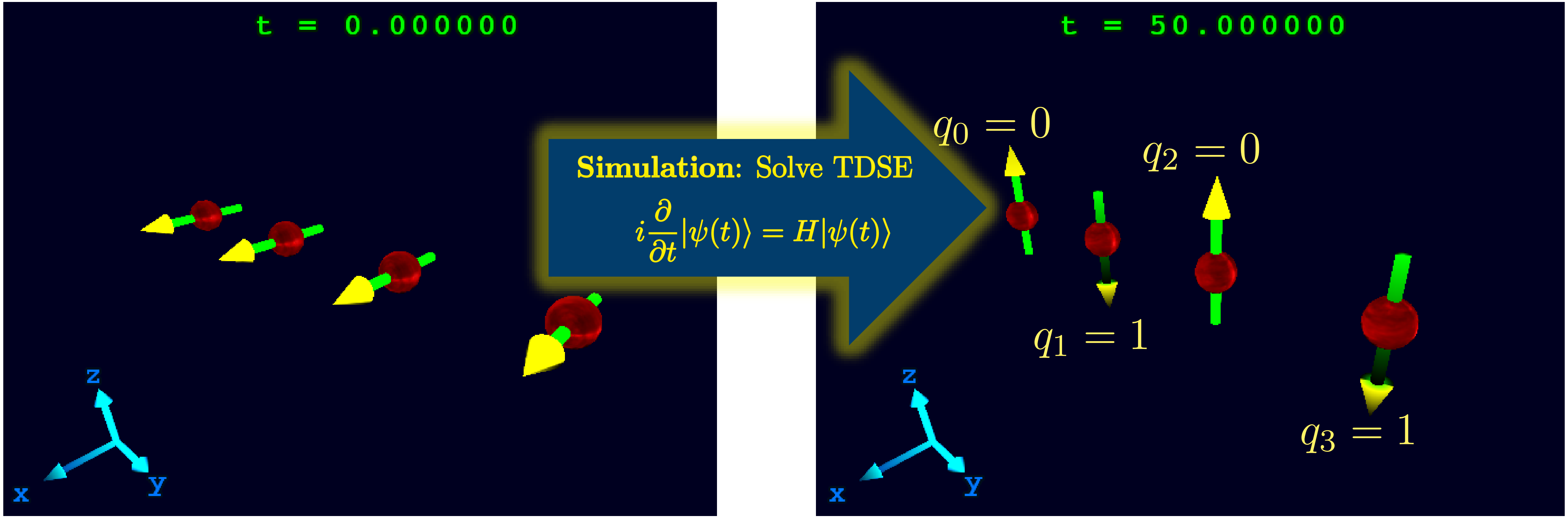
JUQAS

How does quantum annealing work?

$$\textbf{QUBO}: \quad \min_{q_0 q_1 q_2 q_3} \left(\sum_i a_i q_i + \sum_{i<j} b_{ij} q_i q_j \right) = \;????$$

Before the annealing process

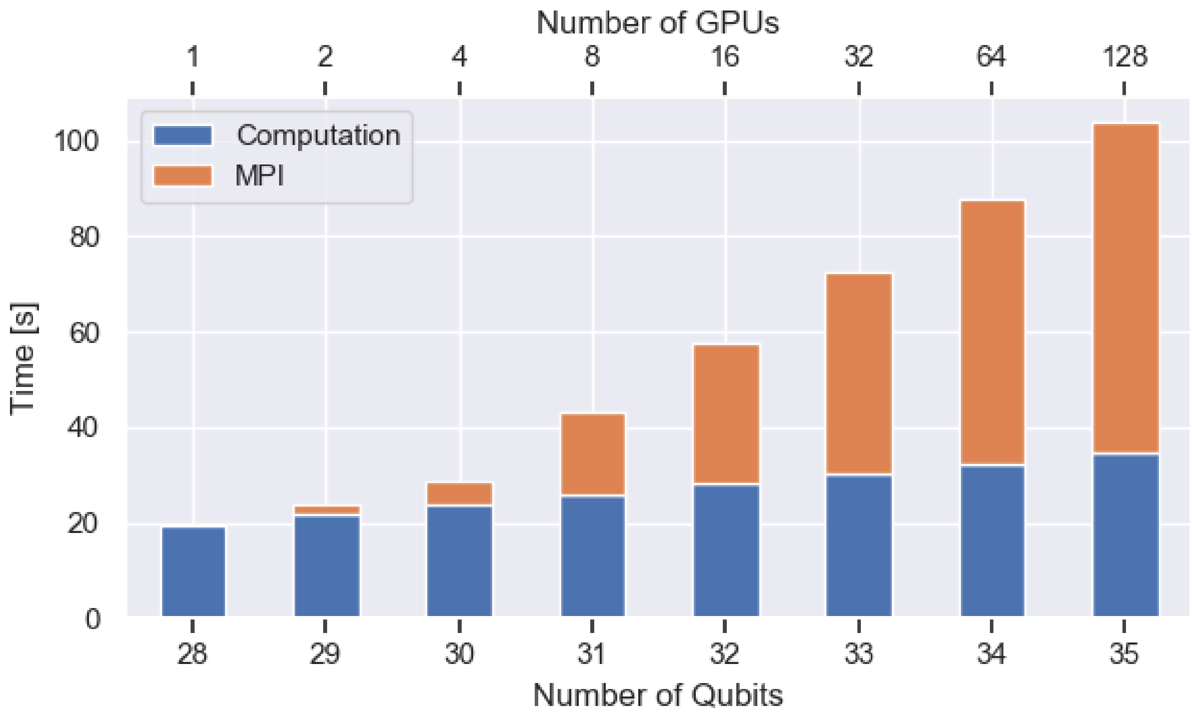
After the annealing process

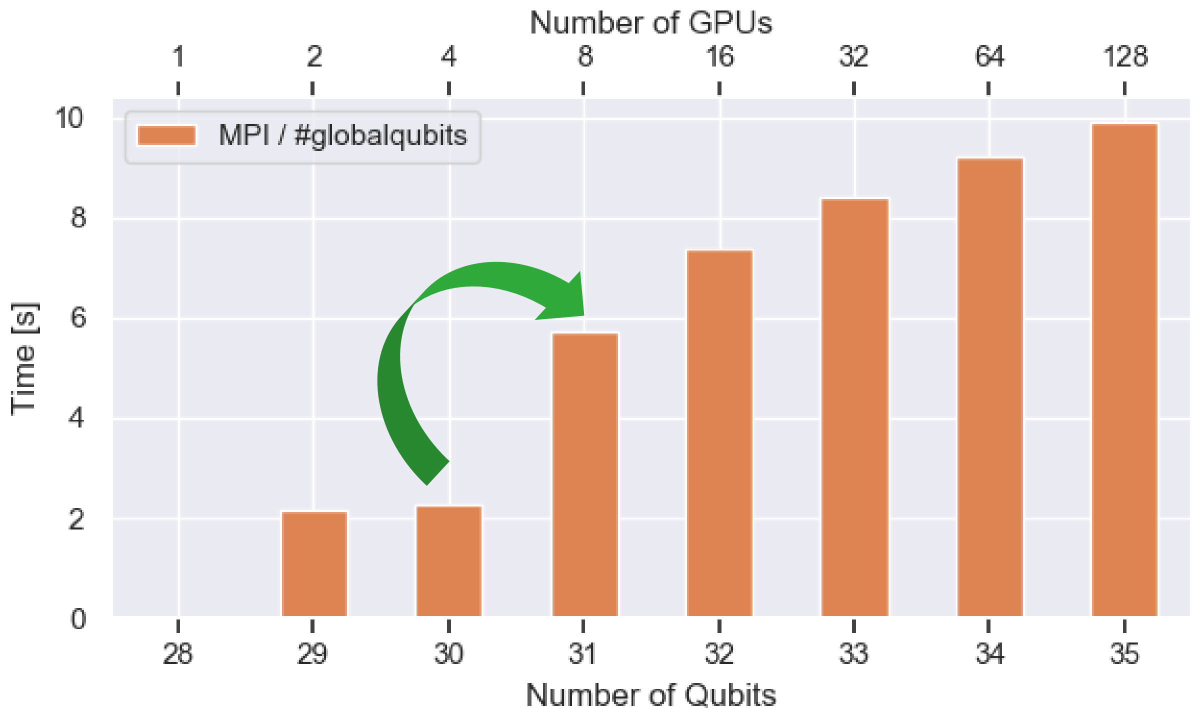


Quantum Annealing

JUQAS: QUANTUM ANNEALING SIMULATOR

Results: “Weak” Scaling





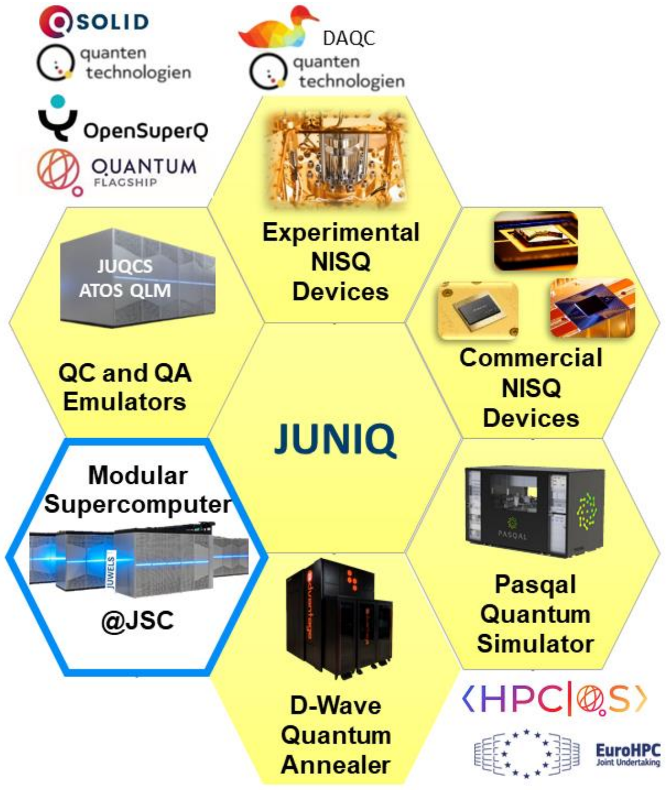
- Computation time almost constant
- MPI time almost linear (≠ exponential!)
- Normalized MPI time reveals **JUMP** from intra-node to inter-node communication

SUMMARY

- Besides **JUQCS**, there are specific QC simulators such as **JUQMES**, **JUQFAS**, and **JUQAS**
- All QC simulators compute a large, complex quantum state

$$|\psi\rangle = (\psi_{\dots q_3 q_2 q_1 q_0}) \text{ or } \rho = (\rho_{k_0 m_0 k_1 m_1 \dots})$$

- Simulating QCs is memory-, network-, and computation-intensive
- QC simulators can profit massively from GPU-based programming



THANK YOU FOR YOUR ATTENTION

- More information and references:
  - **MPI communication scheme:** De Raedt et al., Comput. Phys. Commun. 176, 121 (2007)
  - **Benchmarking gate-based quantum computers:** Michielsen et al., Comput. Phys. Commun. 220, 44 (2017)
  - **JUQCS:** De Raedt et al., Comput. Phys. Commun. 237, 41 (2019)
  - **Quantum supremacy with JUQCS:** Arute et al., Nature 574, 505 (2019)
  - **Benchmarking supercomputers with JUQCS:** Willsch et al., NIC Series 50, 255 (2020)
  - **Airplane scheduling problems on D-Wave quantum annealers:** Willsch et al., Quantum Inf. Process. 21, 141 (2022)
  - **GPU-accelerated simulations of QA and the QAOA:** Willsch et al., Comput. Phys. Commun. 278, 108411 (2022)
  - **Programming Quantum Computers (Lecture Notes):** Willsch et al., https://doi.org/10.48550/arXiv.2201.02051 (2022)
  - **JUQMES:** https://jugit.fz-juelich.de/qip/juqmes