Capabilities of the heterogeneous platform HybriLIT for quantum computing

D. Belyakov[1], A. Bogolubskaya[1], M.I. Zuev[1], Yu. Palii[1,2],
D. Podgainy[1], O. Streltsova[1], D. Yanovich[1]
[1] Meshcheryakov Laboratory of Information Technologies JINR, Dubna, Russia
[2] Institute of Applied Physics, Moldova State University, Chișinău, Republic of Moldova

These studies were supported by the Ministry of Science and Higher Education of the Russian Federation through Grant № 075-10-2020-117.

Mathematical Problems in Quantum Information Technologies
Dubna, Russia, May 27-28, 2024

Heterogeneous platform HybriLIT

System Level

Scientific Linux 7.9
(operating system)

xCAT
(OS deployment tool)

FreeIPA
(auth system)

SLURM
(workload manager)

NFS
(network file system)

Lustre
(parallel file system)

CernVM-FS
(software distribution service)

FlexLM/MathLM
(licence manager system)

Modules
(software environment tool)

Monitoring

- Home-HLIT
- Monitoring HLIT-VDI
- РСК БазИС
- Computing Resources' Statistics

Software Level

Parallel computing software

Open MPI, CUDA, Intel

Licensed software packages

Comsol Multiphysics, Maple, Wolfram Mathematica, Matlab

Application Packages

- GROMACS
- Java
- LAMMPS
- PandaRoot
- Python
- REDUCE
- ROOT
- Cmake
- FairRoot
- FairSoft
- FLAIR
- FLUKA
- GEANT4
- Quantum ESPRESSO

Information Level

HybriLIT web-site
http://hlit.jinr.ru/

Indico
https://indico.jinr.ru/

HybriLIT user support project
https://pm.jinr.ru/

HybriLIT user support telegram
https://web.telegram.org/k/#-1752786710

GitLab
https://gitlab-hybrilit.jinr.ru/

ML/DL/HPC ecosystem

Development component
https://studhub.jinr.ru
https://studhub2.jinr.ru

Component for carrying out resource-intensive calculations
https://jhub1.jinr.ru
https://jhub2.jinr.ru
https://jhub3.jinr.ru

Component for HPC on the HybriLIT platform nodes and data analysis
https://jlabhpc.jinr.ru/

HLIT-VDI
(Virtual Desktop Infrastructure)

Testbed for quantum computing
(quantum computing simulators)

Development of the heterogeneous platform HybriLIT

Cluster HybriLIT **2014**:
Peak performance:
50 TFLOPS double precision
140 TFLOPS single precision

#18 in Top50
Supercomputer "Govorun"
First stage 2018
Peak performance:
500 TFLOPS double precision
1 PFLOPS single precision

#10 in Top50
Supercomputer "Govorun"
Second stage 2019
Peak performance:
860 TFLOPS double precision
1.7 PFLOPF single precision
288 TB UDSS with I/O **>300 Gb/s**

Supercomputer “Govorun”

CPU component

- **21x Servers with Intel Xeon Phi**
  Intel Xeon Phi 7290 (72 cores @1.50 GHz), 96 GB RAM
- **76x Servers with Intel Xeon Scalable Gen2 (RSC Tornado TDN511)**
  2x Intel Xeon Platinum 8268 (24 Cores @2.90 GHz), 192 GB RAM
- **32x Servers with Intel Xeon Scalable Gen2 (RSC Tornado TDN511S)**
  2x Intel Xeon Platinum 8368Q (38 Cores @2.60 GHz), 2 TB RAM

GPU component

- **5x Servers NVIDIA V100**
  2x Intel Xeon E5-2698 v4 (20 cores @2.20 GHz), 8x NVIDIA V100 16 GB, 512 GB RAM
- **5x Servers with NVIDIA A100**
  2x AMD EPYC 7763 (64 Cores @2.45 GHz), 8x NVIDIA A100 80 GB, 2 TB RAM

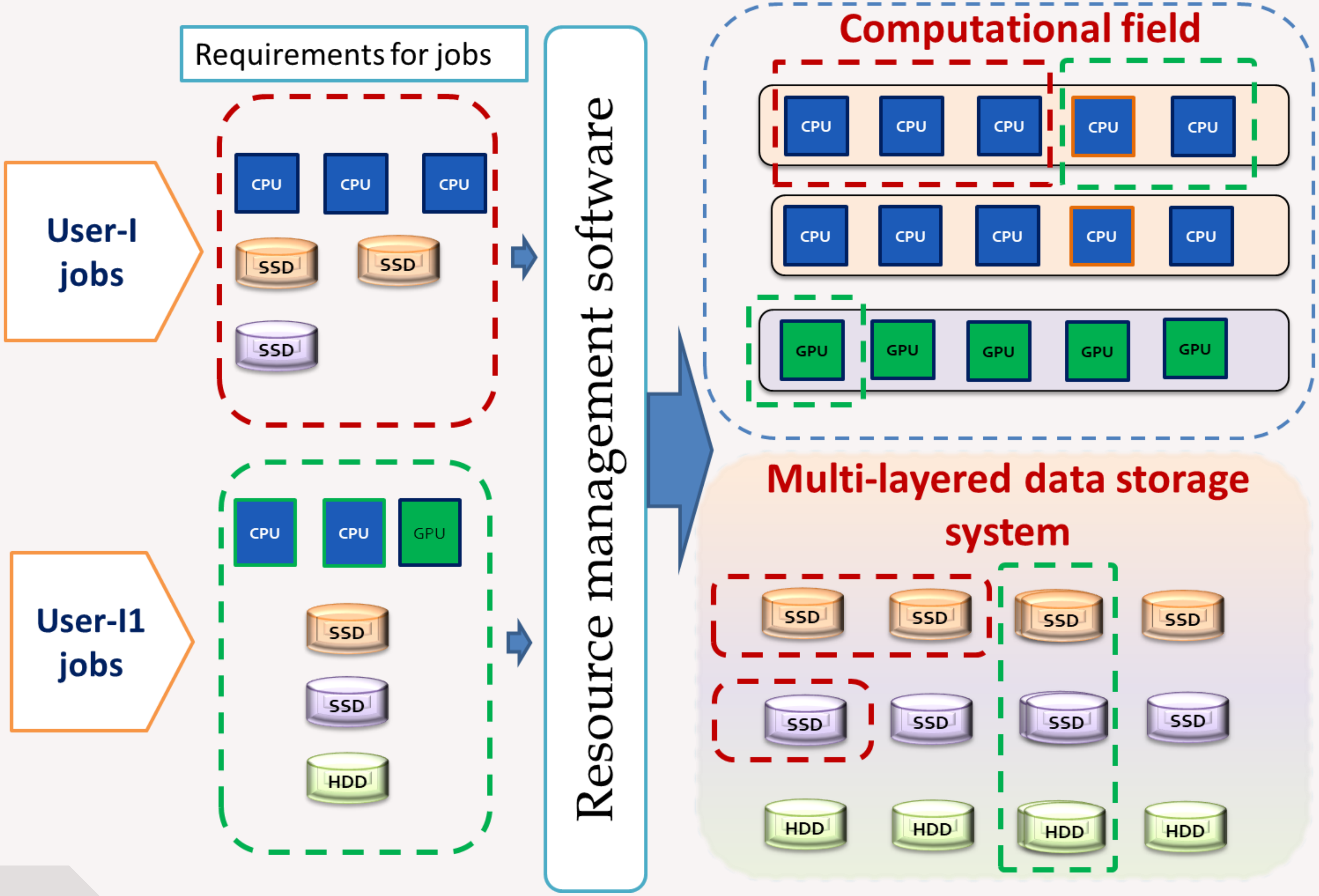
Data storage system: 8.6 PB

Total peak performance

1.7 PFLOPS double precision

3.4 PFLOPS single precision

Orchestration and hyperconvergence



The SC "Govorun" has unique properties for the flexibility of customizing the user's job.

For his job the user can allocate the required number and type of computing nodes and the required volume and type of data storage systems.

This property enables the effective solution of different tasks, which makes the SC "Govorun" a unique tool for research underway at JINR.

Ecosystem for tasks of machine learning, deep learning and data analysis

Component for HPC and data analysis

VM with JupyterHub and SLURM [https://jlabhpc.jinr.ru]

- ❑ Intel Xeon Gold 6126 (24 Cores @ 2.6 GHz)
- ❑ 32 GB RAM

Development component

JupyterLab Server [https://studhub.jinr.ru]
[https://studhub2.jinr.ru]

- ❑ 2x Intel Xeon Gold 6152 (22 Cores @ 2.1 GHz)
- ❑ 512 GB RAM

Component for carrying out resource-intensive calculations

Server with NVIDIA Volta [https://jhub1.jinr.ru]
[https://jhub2.jinr.ru]

- ❑ 2x Intel Xeon Gold 6148 (20 Cores @ 2.4 GHz)
- ❑ 4x NVIDIA Tesla V100 SXM2 32 GB HBM2
- ❑ 512 GB RAM

[https://jhub3.jinr.ru]

- ❑ 2x Intel Xeon E5 2698v4 (20 Cores @ 2.2 GHz)
- ❑ 8x NVIDIA Tesla V100 SXM2 16 GB HBM2
- ❑ 512 GB RAM

Quantum simulators

While quantum computers are not available for widespread use, various simulators of quantum computing on classical computers are being developed.
These are libraries on various programming languages or frameworks that allow to create, transform, optimize and effectively simulate quantum circuits. So, they allow user to completely control the behavior of a quantum system.

Work on the testing ground for quantum computing

Working through the Slurm workload manager

CVMFS

```
AMS/v2021.107_intel
AMS/v2021.107_openmpi
DIRAC/v19.0_intel2018
intel-qs/v20-07-14
intel-qs/v21-01-14
QuEST/v3.4.1
```

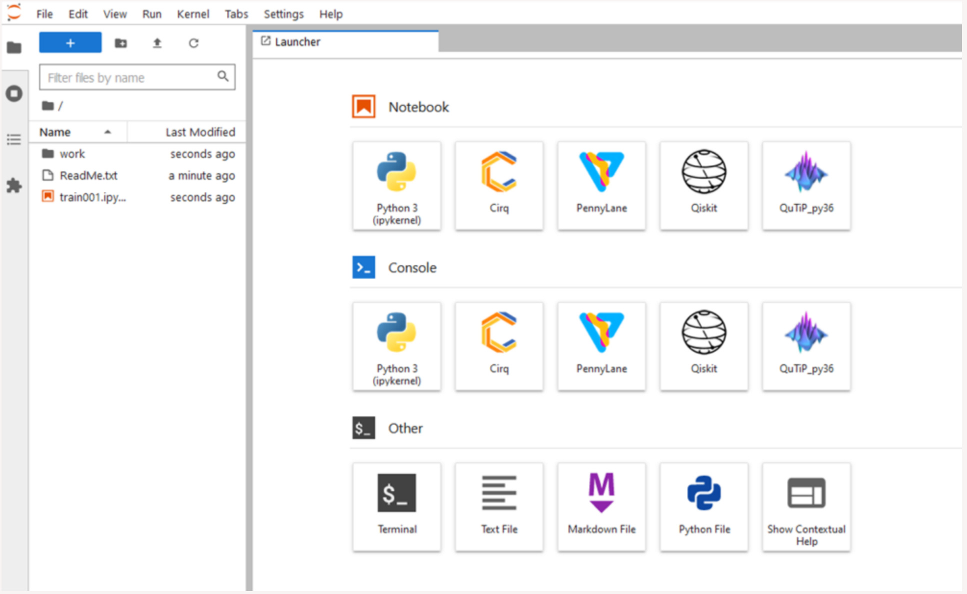
Modules

```
#include <QuEST.h>
#include <time.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

#define TQLENGTH 17

void oneQubitStep(Qureg* qRegister, int
qubitNumber, int* prevOp);
void algorithmStep(Qureg* qRegister, int
qubitNumber, int* prevOneQ, int step);
void twoQubitStep(Qureg* qRegister, int
qubitNumber, int* neighbour);
Vector w;
//ComplexMatrix4 fSim;
...
```

Working in interactive mode



Testbed for quantum computing. Working through the workload manager.

The main advantages:
- the ability to perform multi-node computations using MPI technology;
- the use of resources of the entire heterogeneous platform.

The order of the user's work:
- connection to the HybriLIT heterogeneous platform;
- setting the necessary environment variables for each individual quantum simulator in a work session using the Environment Modules. All available simulators are installed in the CVMFS network file system;
- preparation of a quantum algorithm;
- writing a script-file for run the task: the necessary computing resources (CPU, GPU, RAM), computation time;
- running a task through the SLURM;
- after the program is completed, the user can view the results of the computations and data about the process of the program in output files in his working directory.

Test task. QuEST simulator.

https://quest.qtechtheory.org

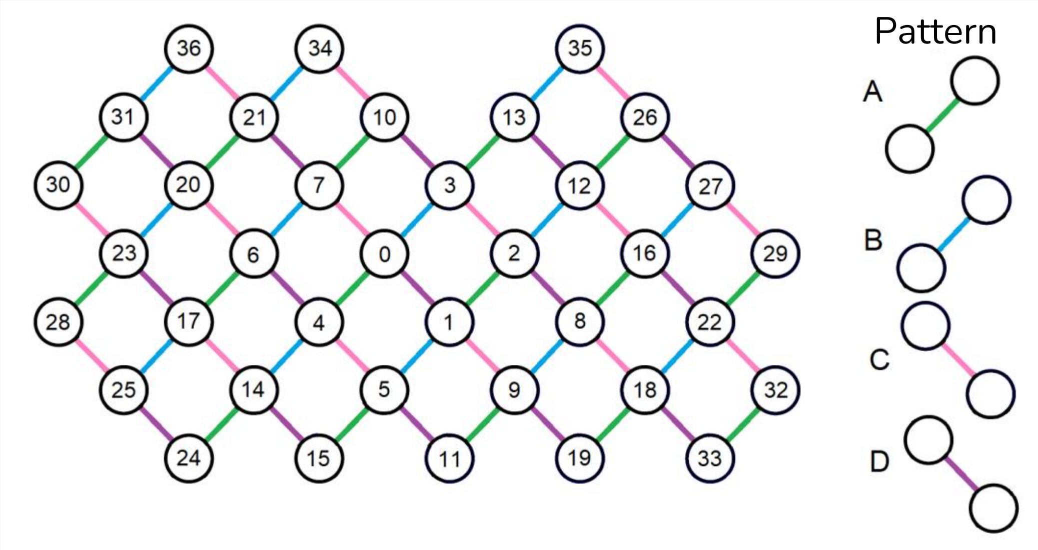
A task using the QuEST simulator together with Ekaterina Kotkova was prepared.

Quantum algorithm for creating a randomized quantum scheme

1. ***m*** *steps of the algorithm consisting of two parts:*
   a. *Applying one-qubit gates to all qubits, which are randomly selected from the set* $\{\sqrt{X}, \sqrt{Y}, \sqrt{W}\}$, *where* $W = (X+Y)/\sqrt{2}$.
   b. *Depending on the step number, applying two-qubit gates according to the pattern* ***ABCDCDAB****: at the first step, gates are applied between qubits with numbers corresponding to the pattern* ***A****, on the second, pattern* ***B****, etc.*
2. *Repeat of step 1.*
3. *Measurement of all qubits.*

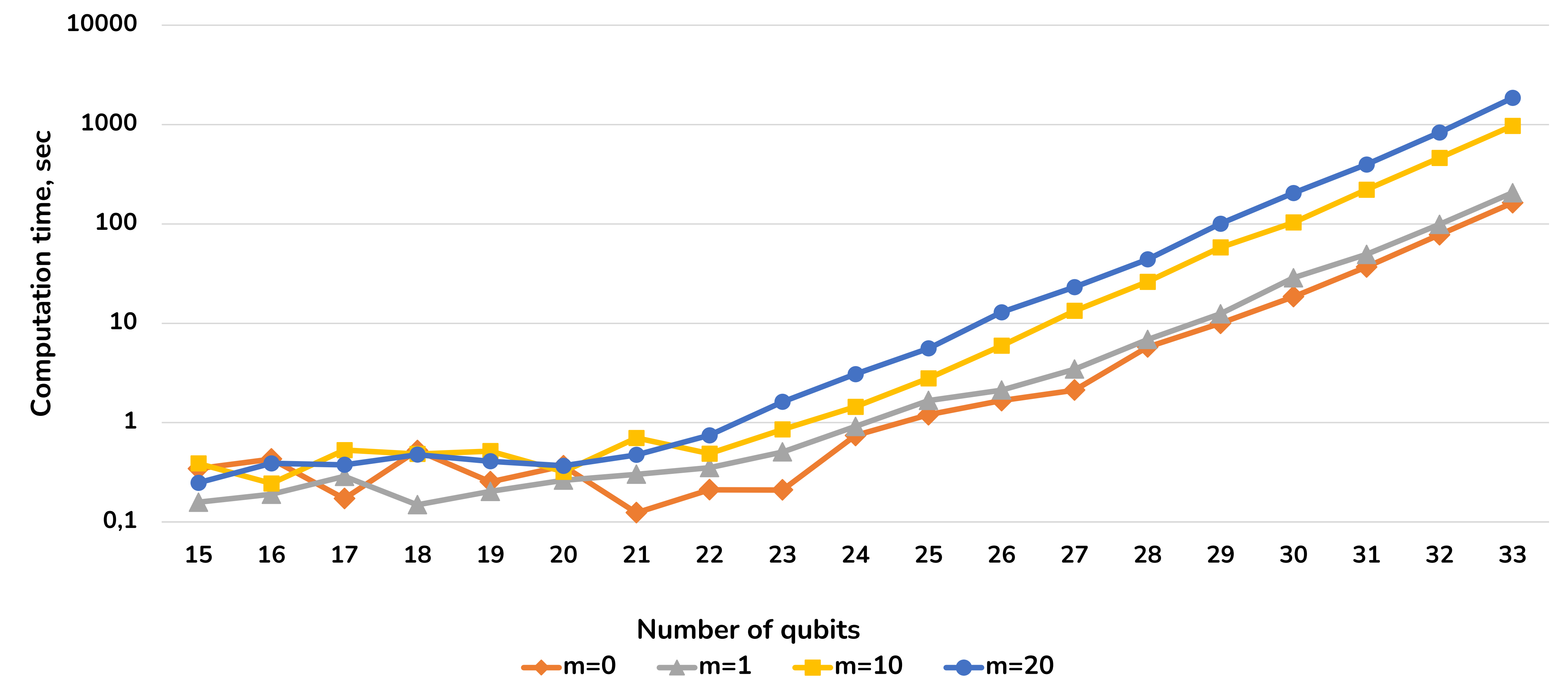
Two-qubit gate:

$$fSim\left(\frac{\pi}{2}, \frac{\pi}{6}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 0 & e^{-i\frac{\pi}{6}} \end{bmatrix}$$



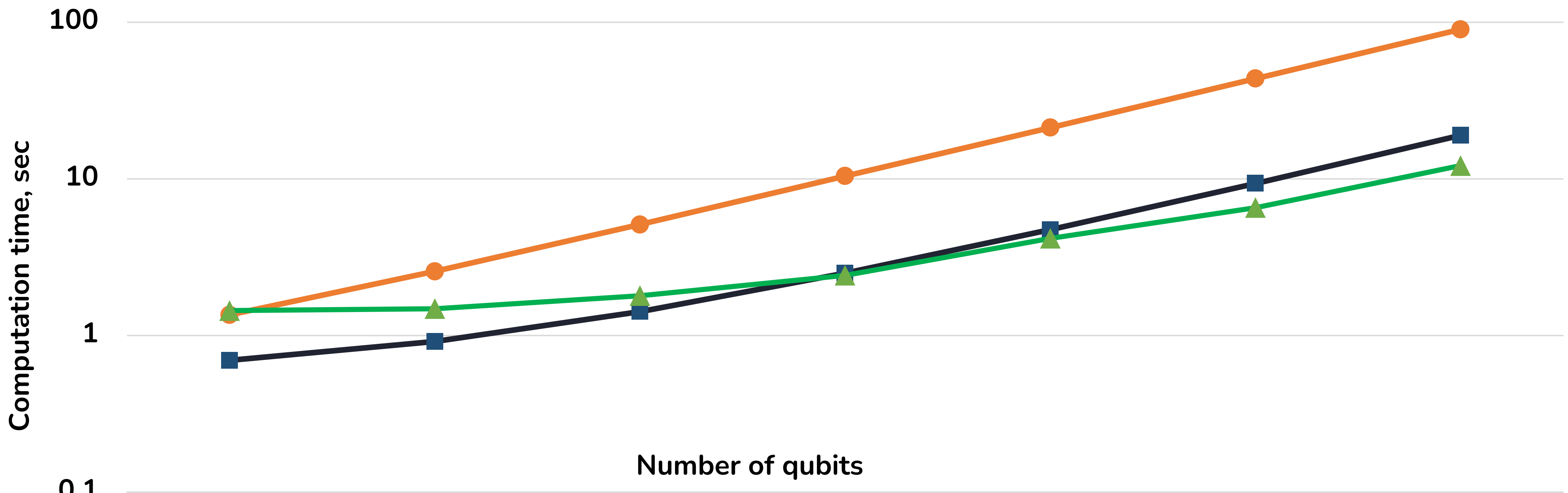
QuEST simulator. Computations on CPU. Execution time.

Server specification: 2x Intel Xeon Platinum 8368Q (38 cores @ 2.60 GHz), 2 TB RAM



Computation time of the task depending on the number of qubits for different numbers of threads ***m***

QuEST simulator. Computations on GPU. Execution time.



	23	24	25	26	27	28	29
Tesla K80	1,354	2,567	5,109	10,424	21,245	43,673	89,965
Tesla V100	0,694	0,916	1,422	2,501	4,735	9,343	18,991
Tesla A100	1,442	1,483	1,797	2,425	4,169	6,546	12,139

Computation time of the task depending on the number of qubits on different GPUs

QuEST simulator. Computations on GPU. Memory usage.

GPU: NVIDIA Tesla V100

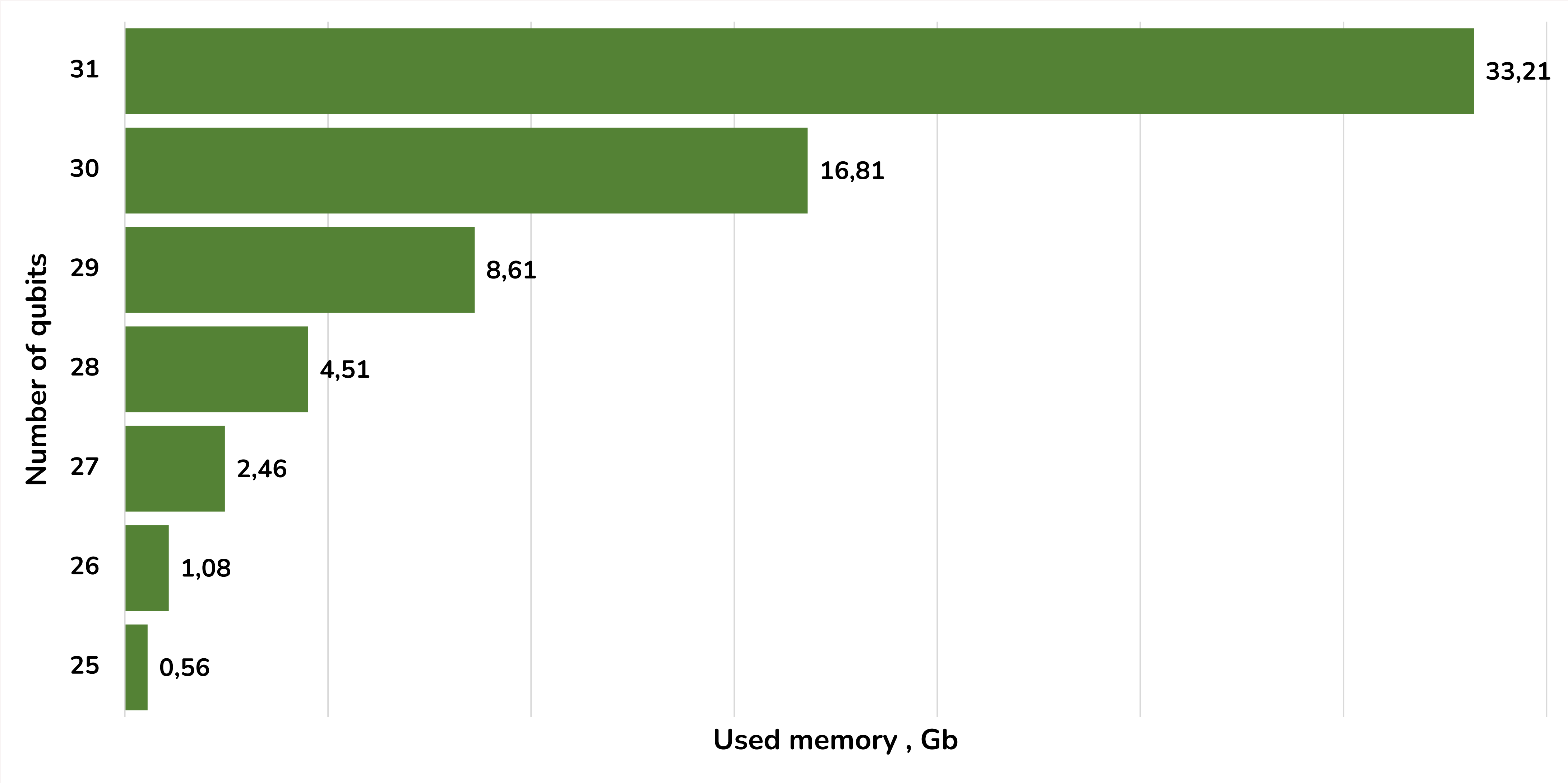
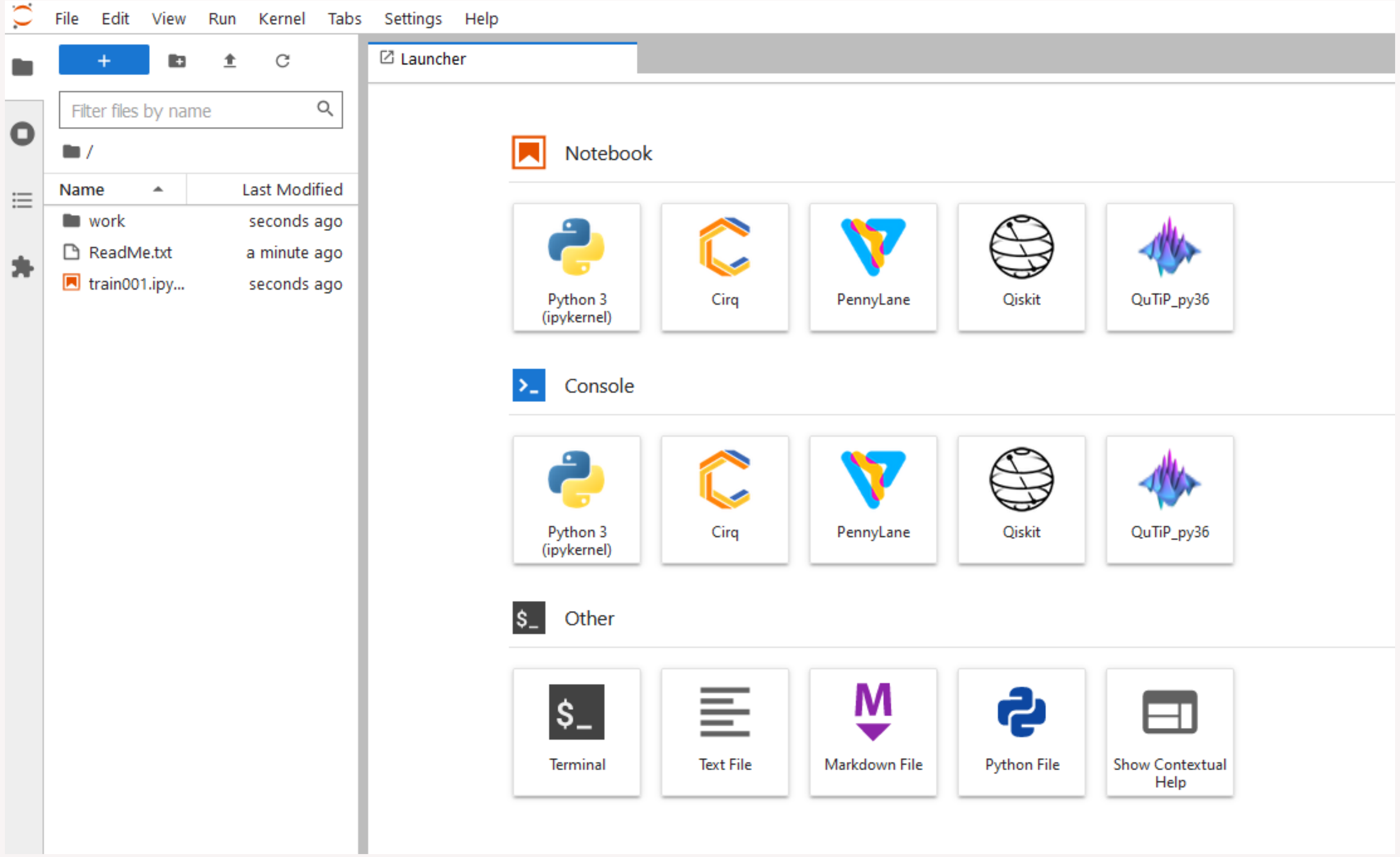


Diagram of dependency used GPU memory on the number of qubits

Testbed for quantum computing. Working in interactive mode.

The main advantages:
- the ability to visually develop algorithms, visualize quantum circuits;
- available Python language materials can significantly speed up research.

Technical implementation.
To work with quantum simulators, a separate server has been allocated, on which the Anaconda package is locally installed. Quantum simulators are installed in virtual environments. Due to this, it is possible to avoid conflicts between versions of libraries that are installed with simulators. Virtual environments are output to the JupyterLab interface by creating a computing core in an interactive ipython shell, which is installed in each environment separately.



Servers specification:
- 2x Intel Xeon E5-2698 (20 cores @ 2.2 GHz), 512 GB RAM, 8x NVIDIA Tesla V100 16 GB
- 2x AMD EPYC 7763 (64 cores @ 2.45 GHz), 2 TB RAM, 8x NVIDIA Tesla A100 80 GB

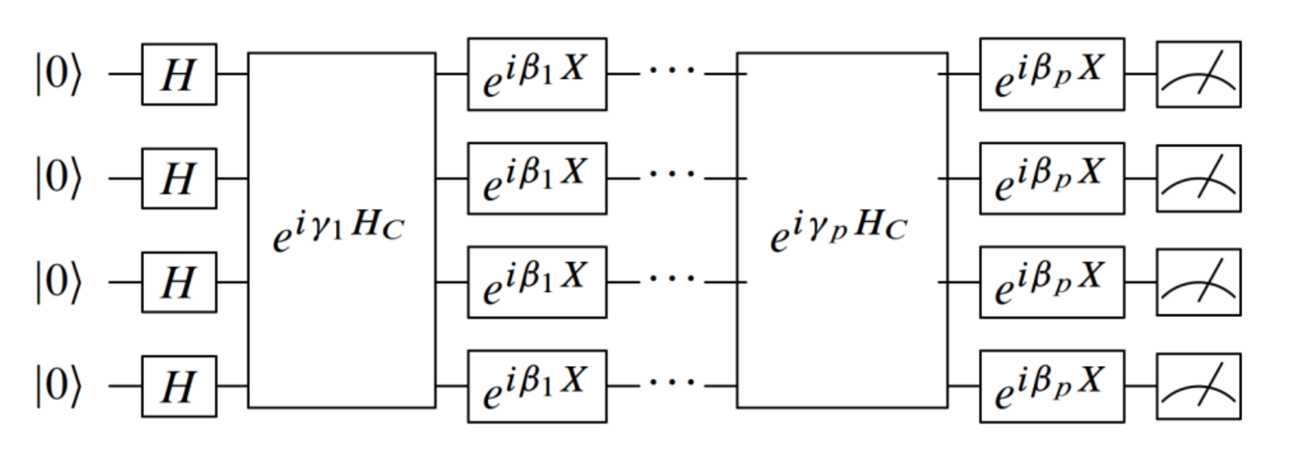
Task 2. Searching for the state with the lowest energy in the Ising model with a longitudinal magnetic field using the quantum approximation optimization algorithm (QAOA).

In QAOA, the function of a quantum computer is to construct a variational ansatz $|\psi(\beta, \gamma)\rangle$ of a wave function with parameters $\beta = (\beta_1,\dots, \beta_n)$ and $\gamma = (\gamma_1,\dots, \gamma_n)$ and measure the quantum energy averages $\mathcal{E}(\beta,\gamma)$ as the average for the Hamiltonian $\mathcal{H}$:

$$\mathcal{E}(\beta, \gamma) = \langle\, \psi(\beta, \gamma) \mid \mathcal{H} \mid \psi(\beta, \gamma)\rangle.$$

On a classic computer, the process of optimizing parameters takes place to reash a minimum value of the average $\mathcal{E}(\beta, \gamma)$.

The solution to the problem is to find a pair of parameters β, γ at which the energy value $\mathcal{E}(\beta, \gamma)$ will be minimal.



A quantum circuit to the variation ansatz of QAOA

$$|\psi(\gamma, \beta)\rangle = \underbrace{U(\beta_p, B)U(\gamma_p, \mathcal{H})}_{p} \dots \underbrace{U(\beta_1, B)U(\gamma_1, \mathcal{H})}_{1}\, H^{\otimes n}|0\rangle^{\otimes n}$$

The problem statement is presented in detail in the work *Yu. Palii, A. Bogolubskaya, D. Yanovich.* **Quantum approximation optimization algorithm for the Ising model in an external magnetic field**
https://indico.jinr.ru/event/3505/contributions/21552

Software implementation. The Cirq library.

https://quantumai.google/cirq

- The software implementation of the algorithm is performed using the Cirq library.
- The qsim optimized simulator is integrated into the Cirq library. It is written in C++ and uses SIMD instructions for vectorization, OpenMP for CPU computations and CUDA for GPU computations.

The use of various parallelization technologies is set by the command

`qsimcirq.QSimOptions()`

which sets the following basic parameters:

`cpu_threads: int = XXX` – number of OpenMP threads,
`use_gpu: bool = False/True` – use CPU or GPU in the simulation process,
`gpu_mode: int = 0/1` – use CUDA or **NVIDIA cuStateVec library**.

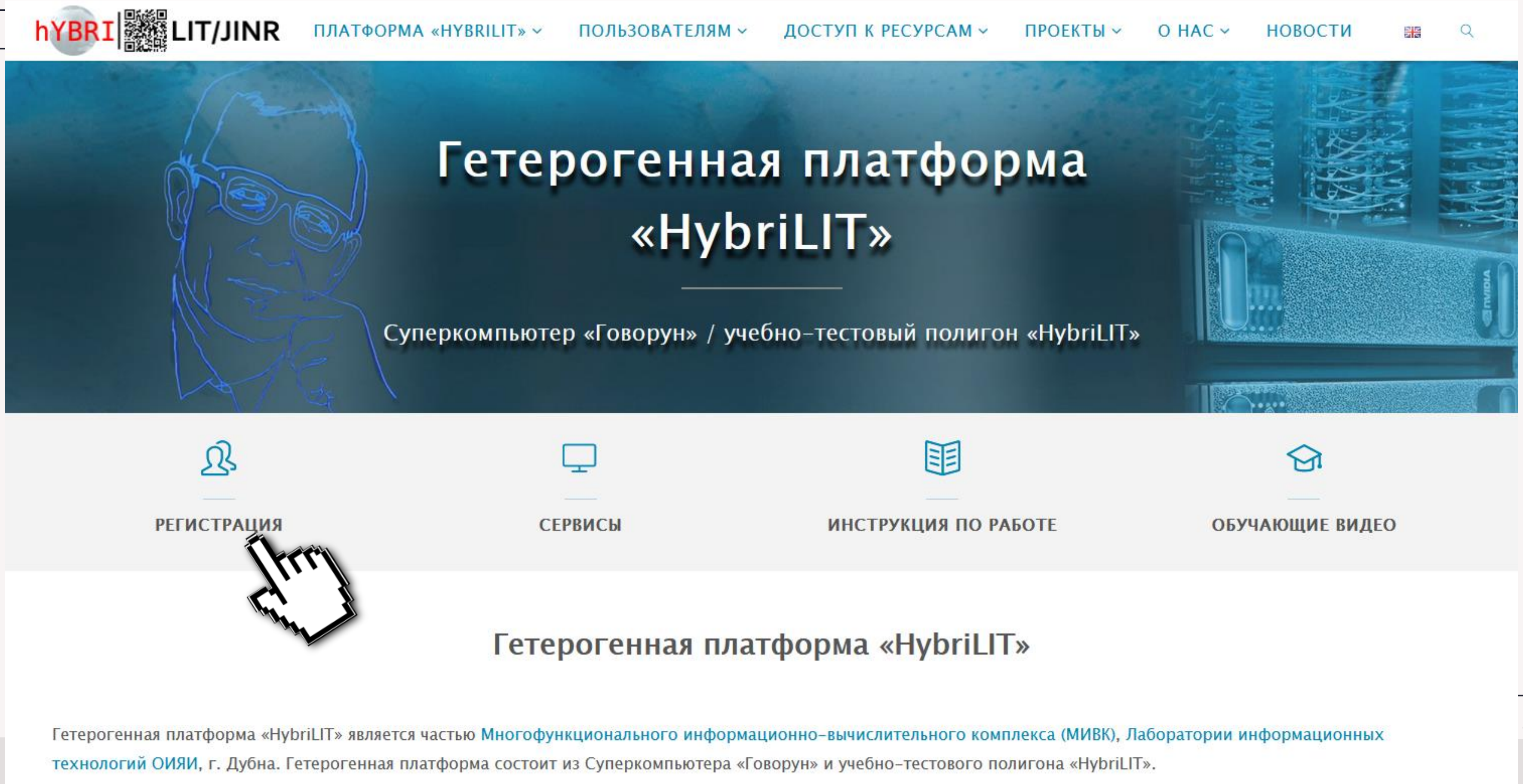
cuStateVec is a library from the NVIDIA cuQuantum SDK for modeling state vectors on the GPU.

Ising Model 3x3x3 lattice 27 qubits	**AMD EPYC 7763, 128 threads**	**Intel Xeon Platinum 8368Q, 128 threads**	**NVIDIA A100, cuStateVec**
Computation time	3 h 20 min	3 h 10 min	14 min 35 sec

Conclusion

- A quantum computing polygon has been organized on the resources of the heterogeneous platform HybriLIT. It is provides the opportunity to work with quantum simulators in two access modes: through the workload manager and in interactive mode.
- The heterogeneous structure of the platform allows to quickly change the characteristics of the testbed, adjusting it to the user tasks requirements, adding servers with the necessary computing components, both CPU and GPU.
- The ability to simulate systems with different numbers of qubits depends on the architecture of the simulator. On the considered tasks, 33 qubits were simulated on the QuEST simulator, and 27 qubits on the Cirq simulator.
- Computations for Ising Model (task 2) had been speeded up on the heterogeneous platform HybriLIT by more than 200 times (from 3 days to 14 minutes) by using the cuStateVec library on GPU A100.

hlit.jinr.ru


hYBRI LIT/JINR
ПЛАТФОРМА «HYBRILIT»
ПОЛЬЗОВАТЕЛЯМ
ДОСТУП К РЕСУРСАМ
ПРОЕКТЫ
О НАС
НОВОСТИ
Гетерогенная платформа «HybriLIT»
Суперкомпьютер «Говорун» / учебно-тестовый полигон «HybriLIT»
РЕГИСТРАЦИЯ
СЕРВИСЫ
ИНСТРУКЦИЯ ПО РАБОТЕ
ОБУЧАЮЩИЕ ВИДЕО
Гетерогенная платформа «HybriLIT»
Гетерогенная платформа «HybriLIT» является частью Многофункционального информационно-вычислительного комплекса (МИВК), Лаборатории информационных технологий ОИЯИ, г. Дубна. Гетерогенная платформа состоит из Суперкомпьютера «Говорун» и учебно-тестового полигона «HybriLIT».