Atos Quantum Learning Machine

Technical training

Ivano Pullano
Global Product Manager - Quantum computing solutions

01. Quantum computing basics

Digital bits and qubits

UX/UI

Software code

Libraries

Algorithms

Instruction sets

Logic gates

Quantum gates

Bits

Qubits

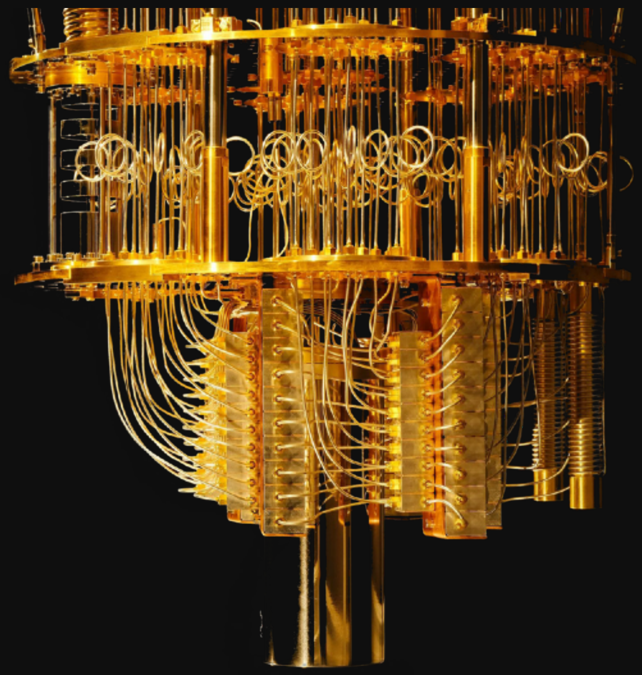
Transistors

Quantum junctions

Quantum Processing Units

Superconducting QPUs

- Made of electronic-grade silicon
- Qubits are spin states of electron currents in superconducting regime
- Pattern of conducting lanes and control systems
- Refrigerated to few thousandth of a degree Kelvin
- Up to 127 qubits
- Very large choice of quantum gates
- Quick operation time
- Very cumbersome machinery
- Limited connectivity between qubits

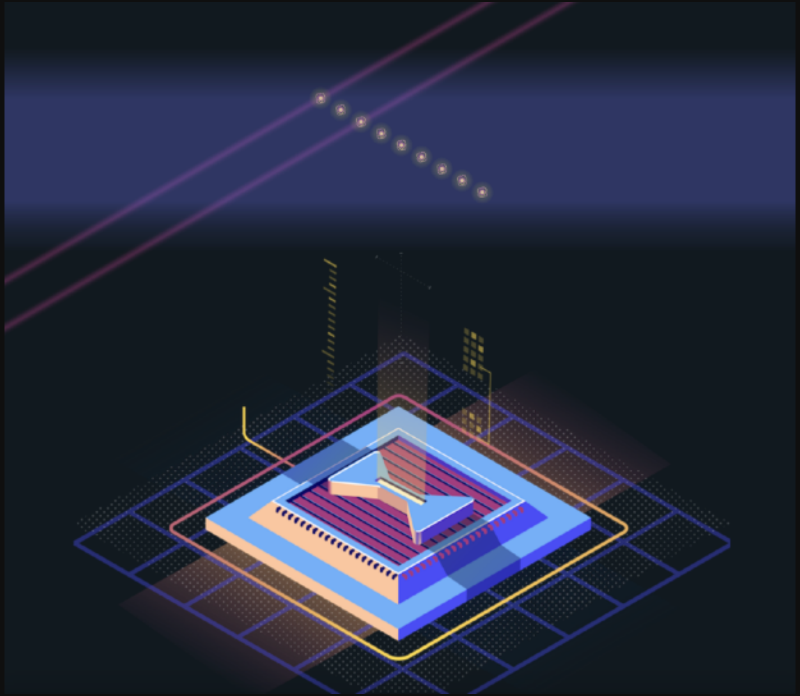


Atos

Quantum Processing Units

Trapped ions

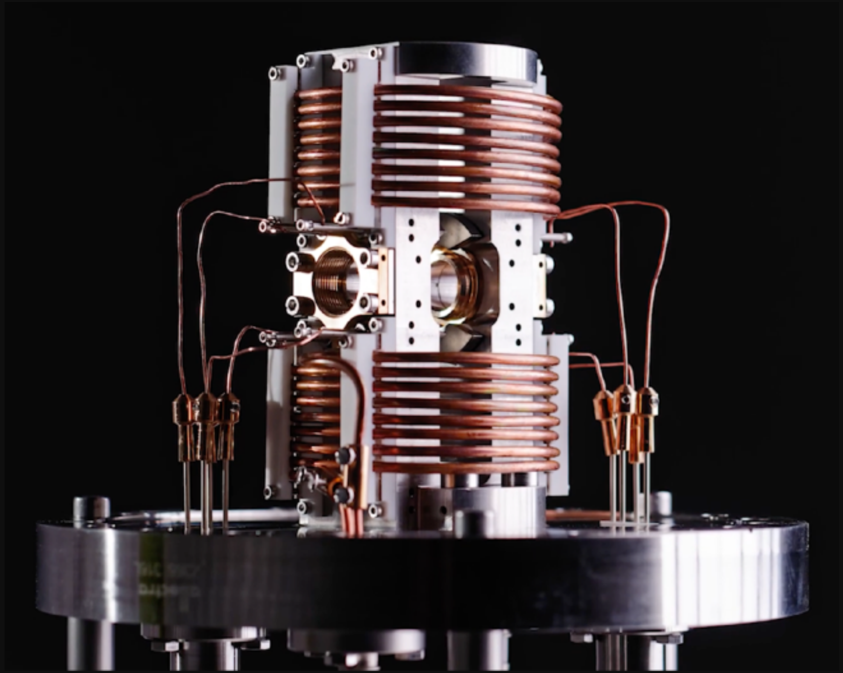
- Made by dispersing certain rare-earth atoms into a matrix
- Qubits are spin states of electrons in ionised atoms
- Ions are in a linear chain
- Pattern of ionising electrodes and control systems
- Refrigerated to few thousandth of a degree Kelvin
- Very high vacuum
- Up to 100 qubits
- Limited choice of quantum gates
- Very stable
- Full connectivity between qubits



Quantum Processing Units

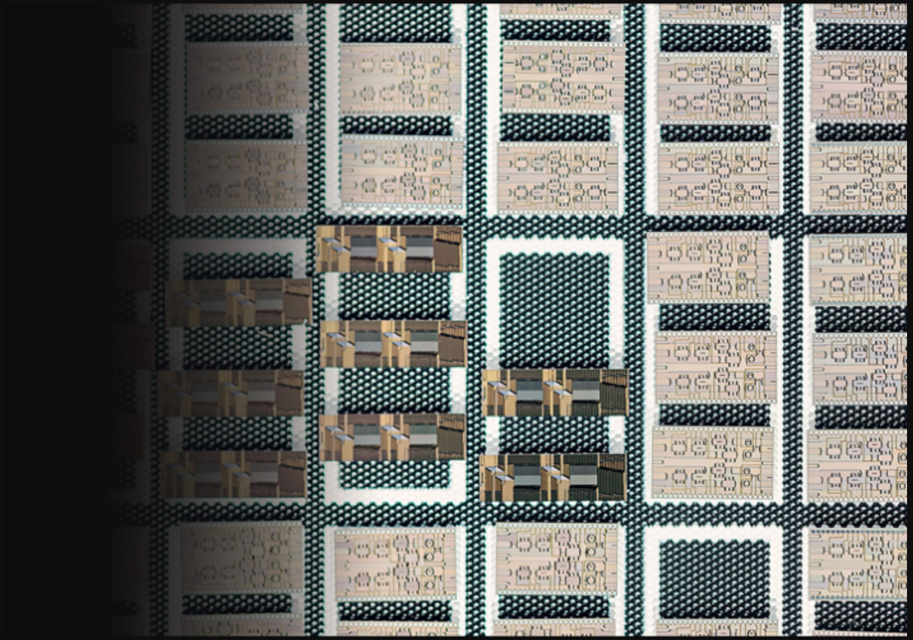
Neutral atoms

- Made by aligning certain rare-earth atoms into traps
- Qubits are excited states of electrons within atoms
- Ions are in a 1D, 2D or 3D array
- Pattern of lasers and control systems
- Refrigerated to few thousandth of a degree Kelvin
- Up to tens of qubits
- Analog and digital computing paradigm (see later)



Quantum Processing Units
Photonic QPUs

- Made of electronic-grade silicon
- Qubits are polarisation states of guided light beams
- Pattern of wave guides and control systems
- Operates at room temperature
- Up to ~30 qubits
- Limited choice of quantum gates
- Slow operation time
- All-to-all connectivity between qubits



Atos

Programming paradigms

Digital computing (gate-based computing)

- Programming flow
    1. Decompose the algorithm into a qubit register and quantum gates
    2. Instruct the qubit register as quantum states of the qubits
    3. Apply the sequence of gates
    4. Readout the final state of the qubits
    5. Build the qubit register upon the qubit states
- Advantages
    - Physics grants that any algorithm can be written as a sequence of quantum gates (under certain conditions)
    - Clear, reusable and maintainable programming code
- Disadvantages
    - Building the gate sequence can be extremely complicated
    - Low scalability to high qubit count

Programming paradigms

Analog computing

- Programming flow
    1. Encode your model into a quantum Hamiltonian
    2. Encode such Hamiltonian into the quantum processing unit
    3. Prepare the initial state of the qubits
    4. Readout the final state of the qubits
    5. Deduce the answer to your problem from the answer you got
- Advantages
    - If you have to understand how a quantum system reacts in a certain environment, this is very easy
    - Reliable result
- Disadvantages
    - Use-case specific

Programming paradigms

Quantum annealing

- Programming flow
    1. Convert your optimisation problem into a specific quantum Hamiltonian
    2. Take a quantum annealer and let it find the minimum energy of an easy Hamiltonian
    3. Slowly change your Hamiltonian to make it look like the one you want to minimise
    4. Readout the final state of the qubits, it will be the optimisation solution
- Advantages
    - Quantum speedup is a tangible reality
    - Automated tools to convert real problems into quantum Hamiltonians
- Disadvantages
    - It works only on numerical optimisation problems

Quantum speedup and quantum supremacy

Introduction

- What do quantum computer do very well
  - Mathematical operations with vectors and matrices
  - Discrete Fourier Transform
  - Modelling quantum physics
  - Amplifying amplitudes of quantum states
- Examples of speedup
  - Research in unstructured database - quadratic speedup
  - Integer number factoring - exponential speedup
  - Quantum Fourier Transform - exponential speedup
  - Quantum generative neural networks - exponential speedup
  - Selected natural language processing tasks - exponential speedup

Quantum speedup and quantum supremacy

Limitations

Data loading

State preparation

Interconnect and data transport

02. Technology landscape

Hardware technologies

Quantum annealing	Super conducting	Trapped ions	Topological qubits	Photons	Silicon qubits	Other technologies
D-Wave The Quantum Computing Company™	IBM, intel, Google, UCSB, rigetti, IQM, QuTech, OQC	IONQ, AQT, universität innsbruck, Honeywell	Microsoft, NOKIA, QuTech, Technische Universiteit Delft	Hewlett Packard Enterprise, XANADU, University of Oxford, ORCA Computing	intel, cea, NOKIA, NTT	QDTI Quantum Diamond Technologies, PASQAL

Atos Quantum computing emulation

Atos

Supeconductive QPUs

- IBM
  - Up to 127 qubits
  - 100 nanoseconds gate application time
  - 5% error rate in gate application
- Rigetti
  - Up to 80 qubits
  - 150 nanoseconds gate application time
  - 4% error in gate application
- Google
  - Up to 54 qubits
  - 35 nanoseconds ion gate application time
  - 3% error in gate application

Quantum annealers

- D-Wave quantum annealer
  - Up to 5,000 qubits
  - 15 nearest-neighbours connectivity
  - Up to 1,000,000 variables problems
  - Up to 100,000 constrains problems

Trapped-ion technology

- IonQ
  - Up to 79 qubits for single-qubit operation
  - Up to 32 qubits for multiple-qubit operation
  - <2 % gate application error rate
  - Full topology
- Alpine Quantum Technology
  - Up to 20 qubits for multiple-gate operation
  - <1% gate application error rate
  - Rack-mount QPU
- Quantinuum
  - Up to 12 qubits for multiple-gate operation
  - <0.5% gate application error rate
  - Full topology

Photonic QPUs

- Xanadu
  - Up to 24 qubits
  - Full connectivity
  - 1 % error rate
- PsiQuantum
  - Currently under construction
  - Up to 1,000,000 qubits

Neutral atoms

- Pasqal
  - 196 neutral atoms array
  - 14 operational qubits
  - 200 atoms simulators
- ColdQuanta
  - 100 neutral atoms array
  - <1% multiple-qubit gate application error rate
  - Large readout error

03. Atos Software

Atos

Atos Quantum solutions

Desktop solution

myQLM

- Freeware
- Entry-level simulation
- Open-source plugins
- Scalability: ~20 qubits

On-premise solution

Atos QLM

- Advanced simulation
  - Noise modelling
  - Optimization
  - Quantum annealing
- Multi-tenancy
- Scalability: 40 qubits
- Optional GPU acceleration

Any quantum computing hardware

Connection to proprietary frameworks via Interop

Product trainings
Atos Quantum Academy
Atos CEPP

Atos

What is Atos Quantum Learning Machine?

A complete programming environment and a quantum processor emulator

Programming

AQASM
Assembly language to build quantum circuits

pyAQASM
Python extension to AQASM

CIRC
Binary format of quantum circuits

QLIB
AQASM & pyAQASM libraries

CO problems class
Describe any Combinatorial Optimization problem

INTEROP (open-source)
Connectors with other frameworks

ProjectQ

Qiskit

Optimization

PBO
Pattern based optimizer

NNIZER
Topology constraint solver

Circuit Optimizer
Generic circuit optimizer

Simulation

Simulators
Digital QC Simulators
Quantum-Inspired Simulators

Physics
Physical Noise models

myQLM
Included functionalities

Programming

AQASM
Assembly language to build quantum circuits

pyAQASM
Python extension to AQASM

CIRC
Binary format of quantum circuits

QLIB
AQASM & pyAQASM libraries

CO problems class
Describe any Combinatorial Optimization problem

INTEROP (open-source)
Connectors with other frameworks

ProjectQ

Qiskit

Simulation

Noiseless
Open-source simulation engine, fully customizable

Atos

myQLM Power Access

Design principles

Simplify the connection between user and remote QPUs

Asynchronous job execution

Better integration with HPC resources with Advanced Edition

One platform for local and remote program runs

myQLM Power Access

User workflow

Without myQLM Power Access

Write and test your code on a local machine

Open a network session with a remote QLM

Write the code adding advanced QLM features

Transfer the code sheet to the QLM

Run the code and download the output

With myQLM Power Access

Write and test your code on a local machine

Add the required advanced features in the QLM

Run the code and get the output on the local machine

myQLM Power Access

System administrator workflow

Without myQLM Power Access

Configure the system to be made available in the customer's network

Manually add users and SSH keys

Hardware usage and job reporting via external tools

With myQLM Power Access

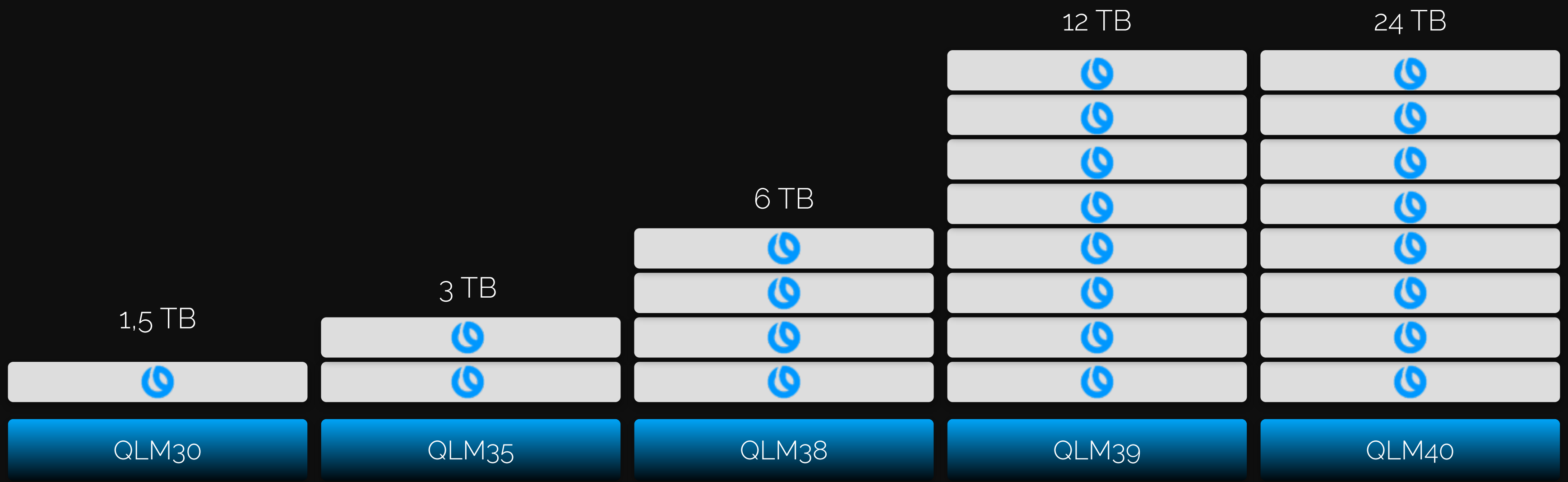
Configure the system to be made available in the customer's network

Add users manually or automatically via Django portal

Monitor and manage the machine using the same tools as the rest of the datacenter

04. Atos Hardware

Atos QLM
Hardware range

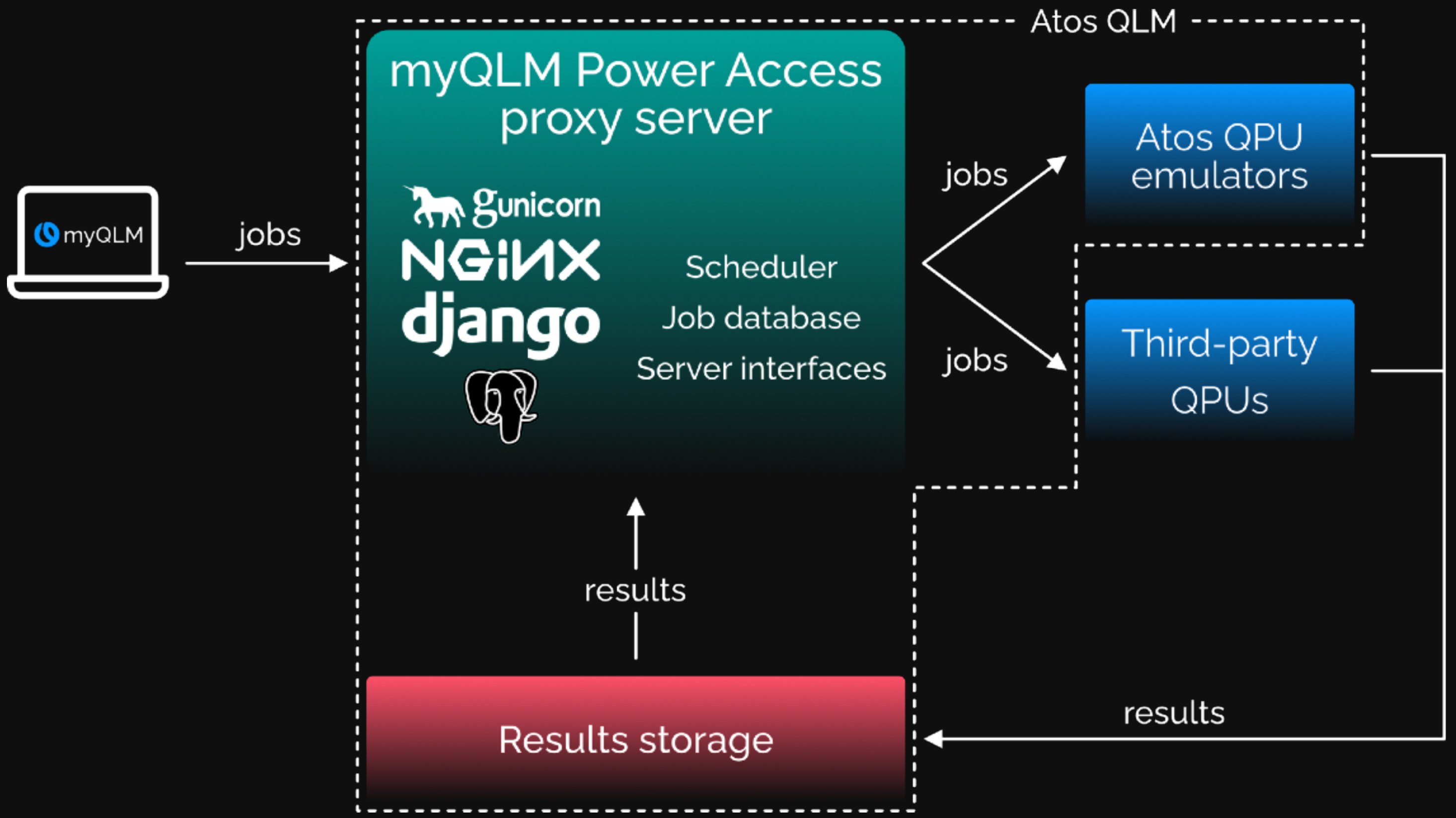


Atos QLM hardware
Configuration details

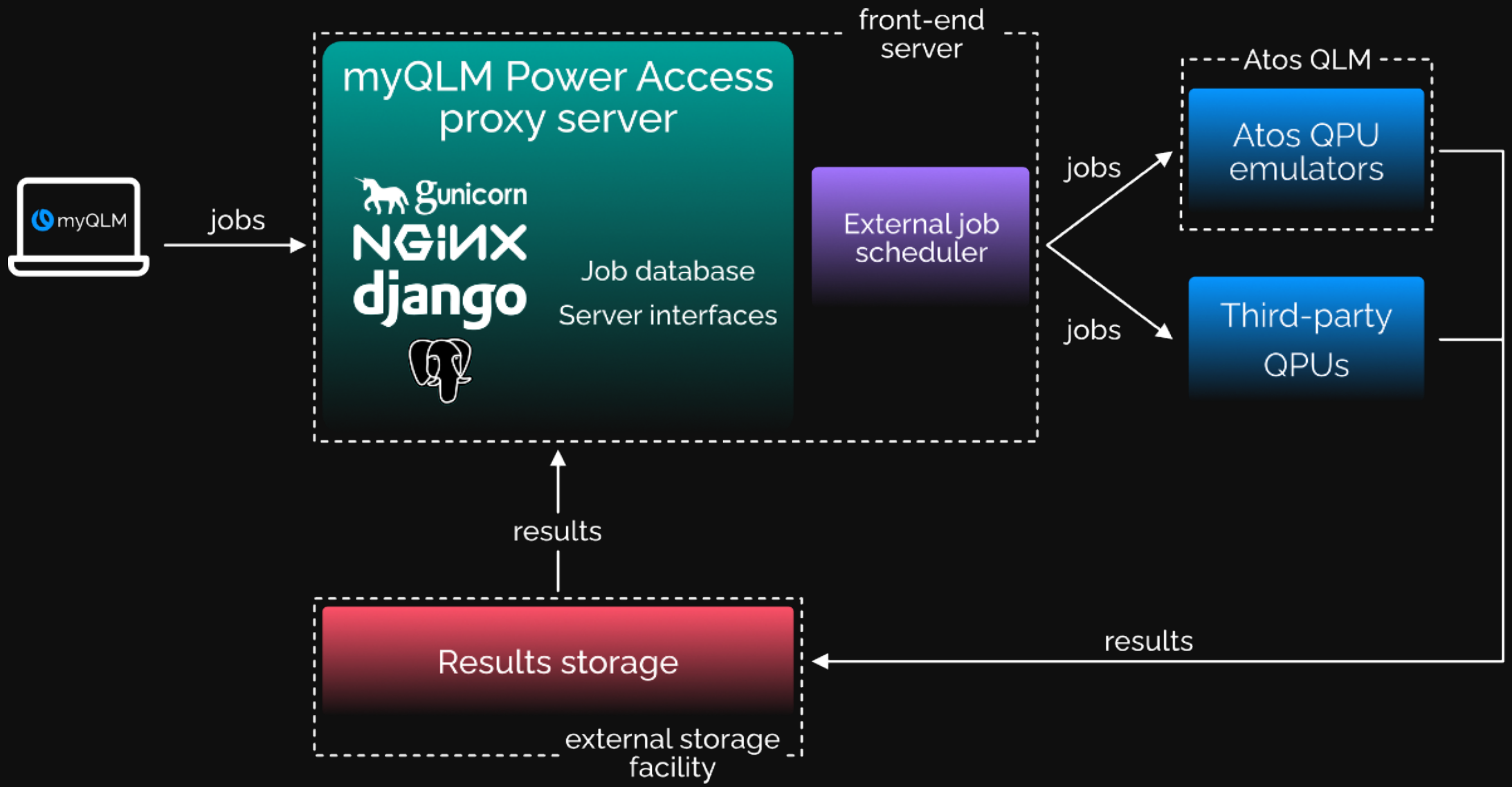
- 19" rackmount server - from 2 to 16 rack units
- Shared-memory machines based on custom NUMA switch implementation
- Intel Cascade Lake CPU - from 2 to 16
- DDR4 RAM - from 1.5 to 24 TB (from 64GB to 128GB DIMMs)
- 10Gb/s dual-port board SFP+
- 1 Gb/s dual-port board RJ45
- SSDs and HDDs for OS boot and data storage

myQLM Power Access

Free edition



myQLM Power Access
Power edition



Questions?

Thank you

For further information please contact
Ivano Pullano
ivano.pullano@atos.net