QUANDELA

Quantum Computing with single photons

Perceval, an open-source framework for photonic quantum computing

PERCEVAL

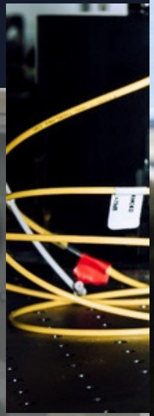
Jean Senellart, 29/05/2024

QUANDELA

- About Quandela
- 101 Photonic Quantum Computing : how to build your own Quantum Computer!
- The Software Stack – exQalibur, MosaiqOS, **Perceval** and the Quantum Toolbox

QUANDELA

- About Quandela
- 101 Photonic Quantum Computing : how to build your own Quantum Computer!
- The Software Stack – exQalibur, MosaiqOS, **Perceval** and the Quantum Toolbox



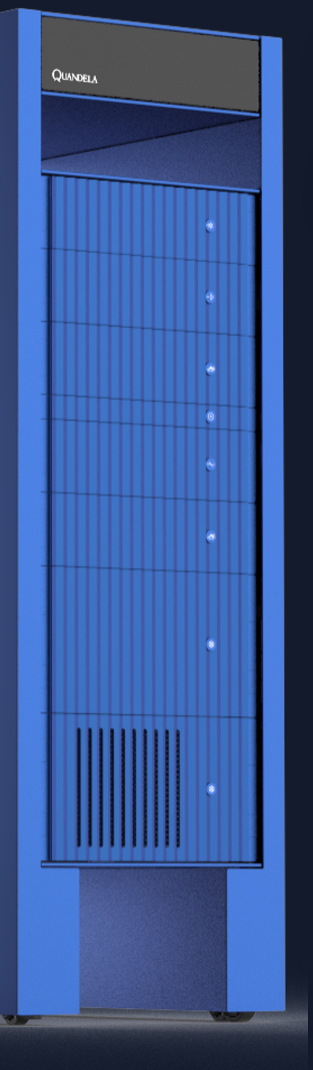
OVH cloud datacenter, Croix – France, 1st delivered QC system, November 2023.

- About Quandela

➢ **101 Photonic Quantum Computing : how to build your own Quantum Computer!**

- The Software Stack – exQalibur, MosaiqOS, **Perceval** and the Quantum Toolbox

Quandela Today: A full stack Quantum Computing company



MOSAIQ

Software & cloud

Algorithms & Protocols

Compilation

Optics / PICs / Electronics

Semiconductor

The offer

Quantum as a Service
1 QPU live since Nov 2022
GPU-boosted simulator
600+ Active users on Quandela Cloud

Research Community
Open Source framework
#2000s users
4 Hackathons

Algorithm toolbox
#8 projects today + *12* in the pipe
Consulting Service

MOSAIQ on-premise
2023: 1 QC delivered
2024: 1 QC delivered + 3 in preparation

Expertise across all sectors

Software dev. & Computer science
#10 experts + internal cloud dev. in 2023

Quantum computing
#20 scientists + #6 patents

Systems integration / engineering
Production facility + QC cloud farm

Semiconductor physics
Proprietary cleanroom

Photonics, an **optimal** platform for universal QC

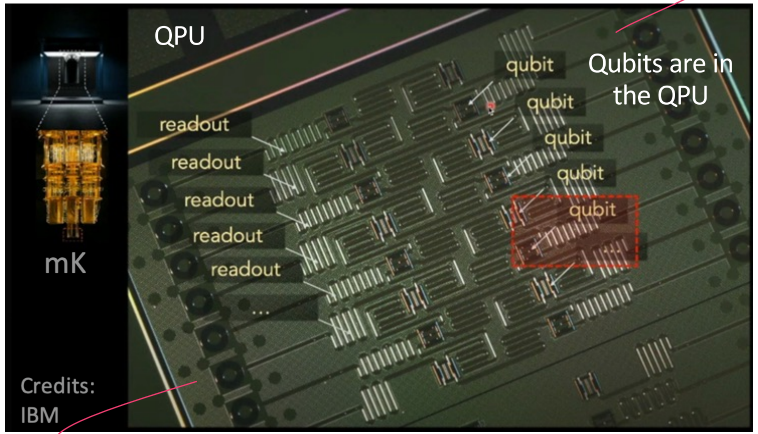
Manufacturing	Capability to build and deliver several industry grade QCs / year
Adaptability	Low cooling constraints, ease-of-integration in current datacenters, energy efficient
Hybridization	Ability to be connected to classical CPU, GPUs, and integrated into current infrastructures (computing, networks and communication)
Algorithms	Ready-to-use optimized primitives
Scalability	Single Photon is the only particle able to transport quantum state between multiple computers

Digital Quantum Computing Approaches

1 Matter Qubits: Ions, Superconductors, Cold Atoms…

Physical QuBits

Control Signals IN

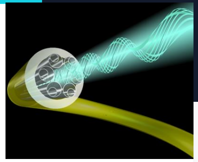


Read-OUT Signals

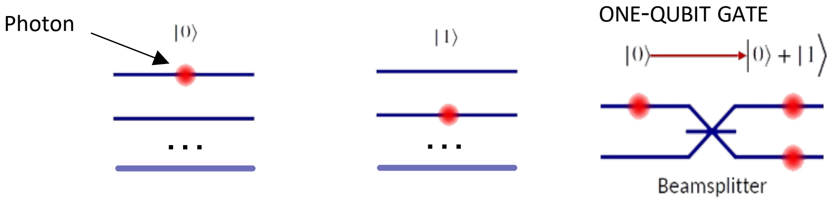
+	Highly EFFICIENT 2-qubit GATES (deterministic)
BUT	Qubits have a physical size → Manufacturability roadblocks
BUT	Each qubit undergoes DECOHERENCE → errors with #qubits

2 Photonic Qubits

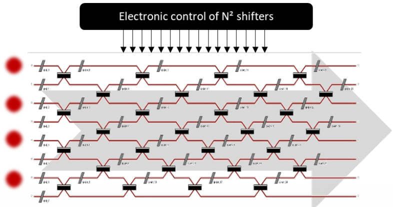
Flying Qubits



1 PHOTON IN 2 SPATIAL MODES (DUAL RAIL ENCODING)



SEQUENCE OF N-QUBIT GATES

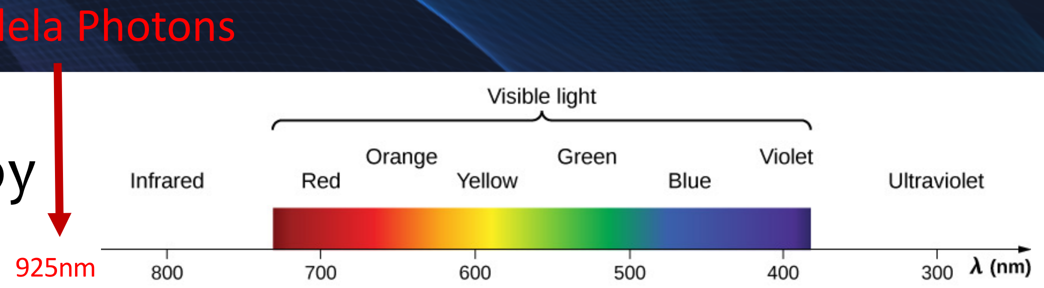


- + Increase Qubit# with fix hardware size
- + ROBUST qubits (no decoherence)
- + MANIPULATION at room temperature with standard optics

1 & 2 Quandela exploits the efficient manipulation of optical qubits but tackle probabilistic nature of gates with a matter-based qubit generator

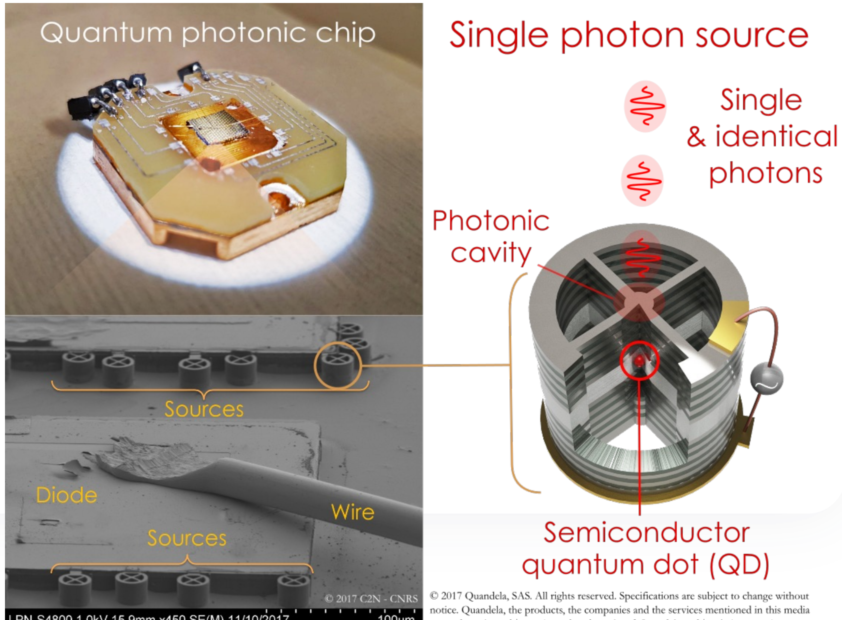
Computing with light

The Physics – 101



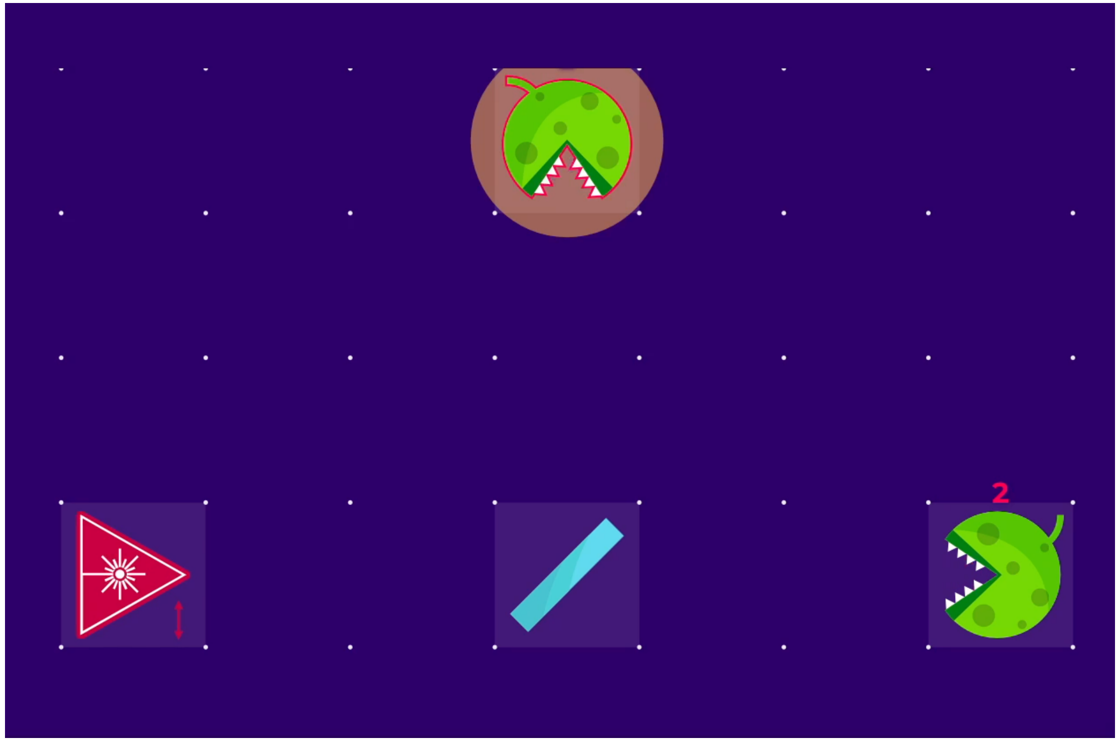
- Light is an **electromagnetic wave** described by Maxwell's equations
- **A single photon is the quantum of light.** It is an indivisible particle-like entity behaving as **both** a particle and a wave.
- The energy of a single photon is: $E = \frac{hc}{\lambda}$. For visible light: $\boldsymbol{E < 1aJ\ (10^{-18}J)}$
- A single photon will **only** interfere with another single photon at the same position, wavelength, polarization.

  → Such photons are called **indistinguishable**
- Single Photons can be deterministically generated using **quantum dots**

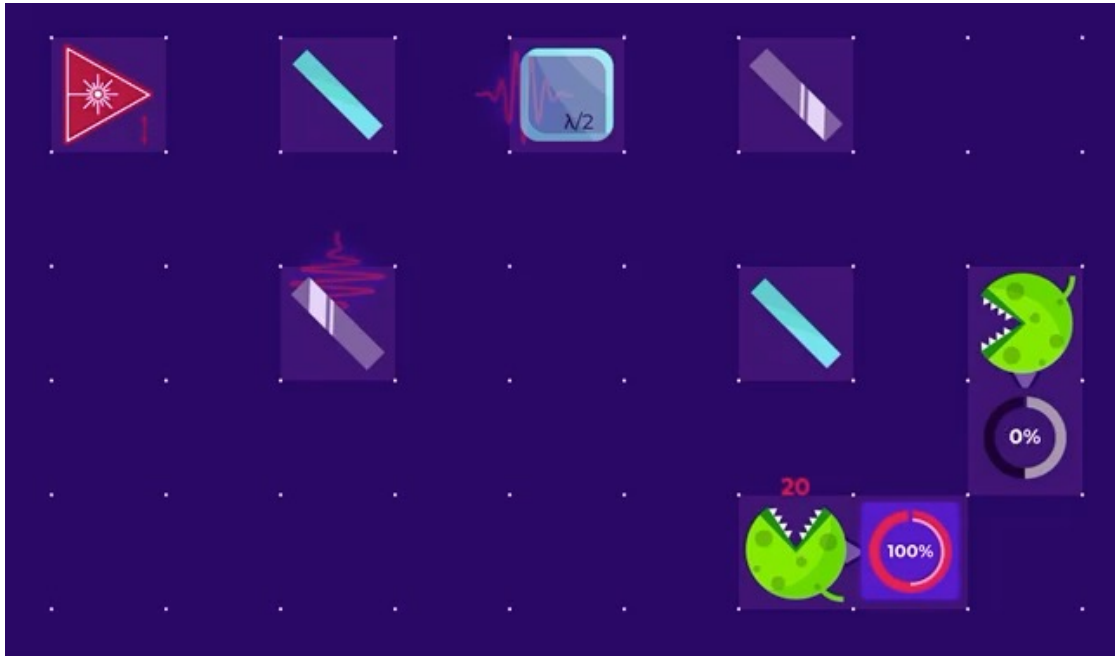


Quantum Computing with Light

Superposition



Interference – Mach-Zehnder interferometer



Computing with light

The Optics - 101

- As a particle of light, a photon will have the different properties:
  - Speed of a single photon depends on the refractive index of a material – **change of speed can be used to change phase of a single photon** (practically we can use *change of temperature*)
  - When a single photon hits a new material boundary, it can be **refracted** or **reflected**
  - A single photon can be **absorbed** by almost any material on its way (**photon loss**)
- Common passive optical elements are: beamsplitters, polarizers, mirrors, etc... which follows linear equations (**linear optics**)
- Single photons can be detected with high accuracy with Superconducting Nanowire Single-Photon Detector (**SNSPD**)

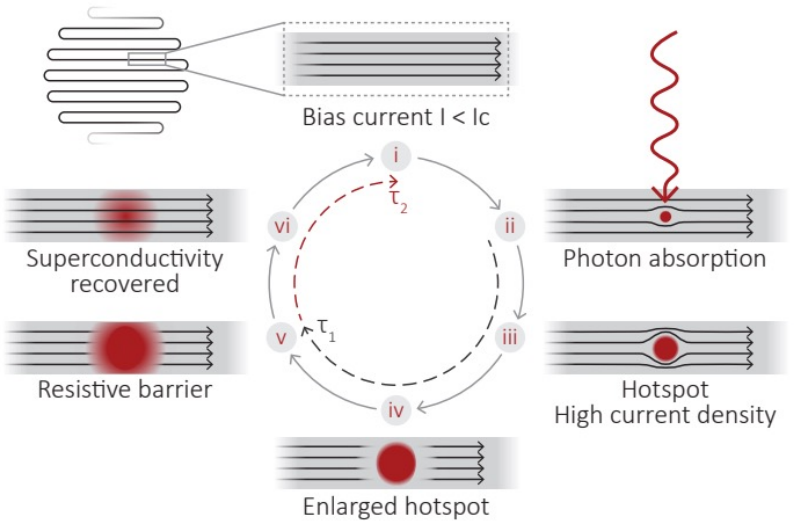
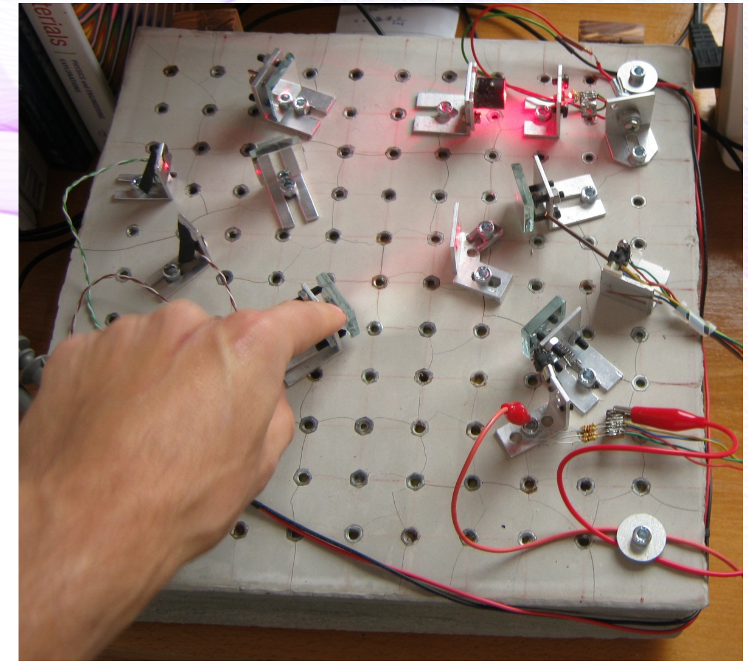


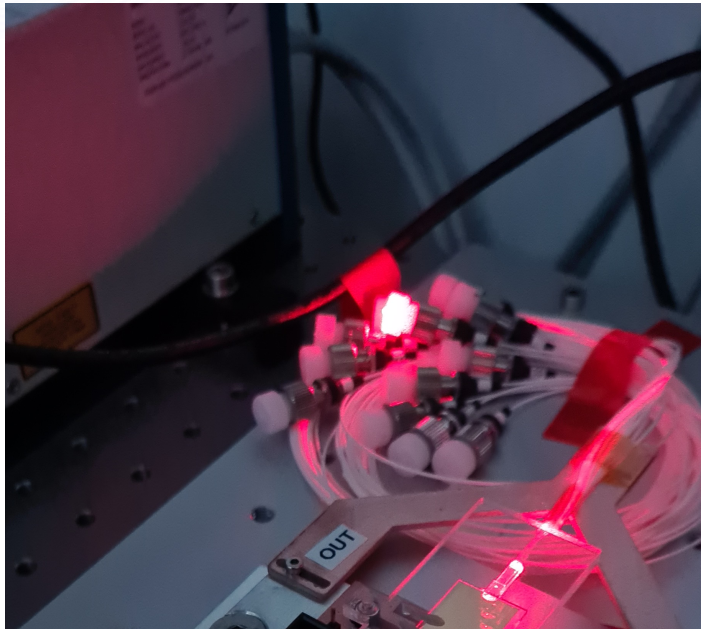
Figure adapted from: Natarajan et al. Supercond. Sci. Technol. 25, 063001 (2012).

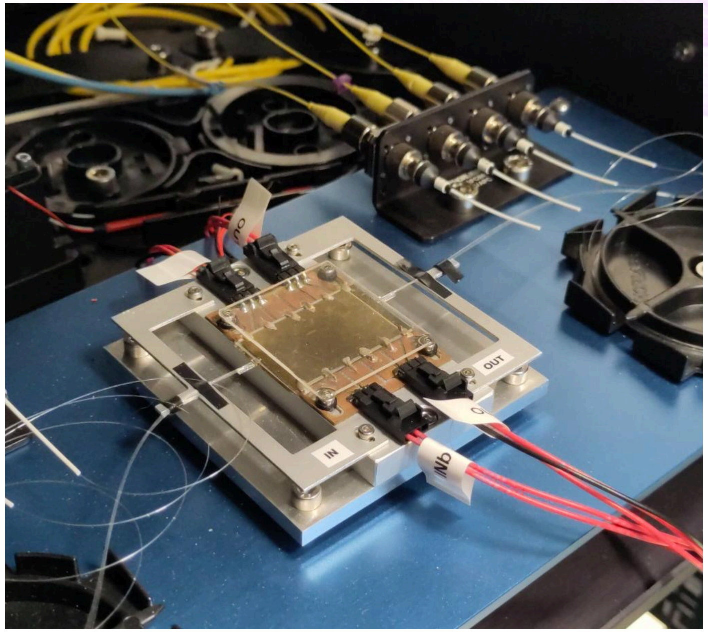
Computing with light
With optical components

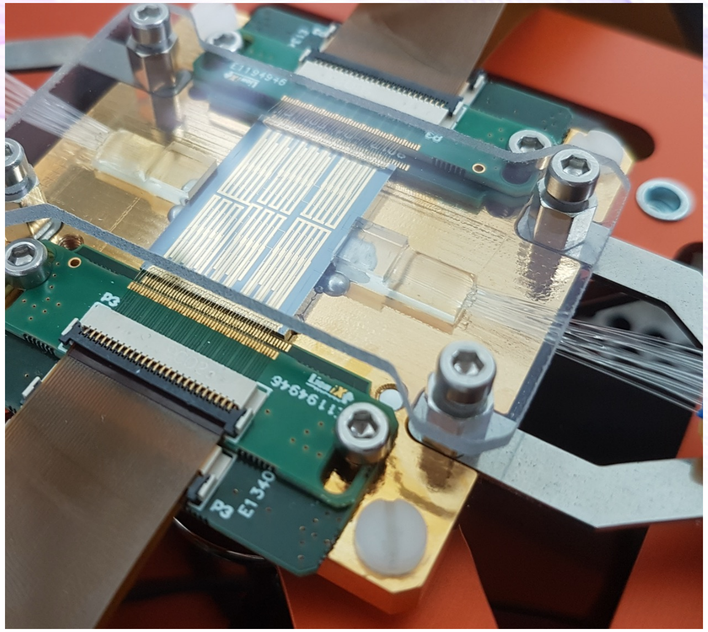


Computing with light

On photonic chip





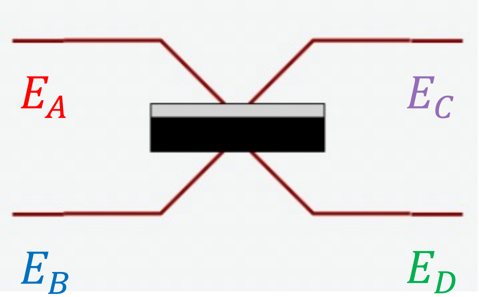


Computing with light

Linear Optics - 101

- **Linear optics** transformations can be represented by unitary matrices

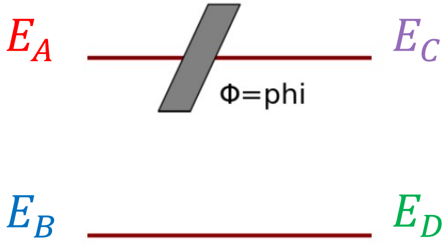
The beam-splitter:



$$\begin{bmatrix} E_C \\ E_D \end{bmatrix} = \begin{bmatrix} Re^{i\phi_{ac}} & Te^{i\phi_{bc}} \\ Te^{i\phi_{ad}} & Re^{i\phi_{bd}} \end{bmatrix} \begin{bmatrix} E_A \\ E_B \end{bmatrix} \quad R^2 + T^2 = 1$$

(more generally any 2-mode unitary is a beam-splitter)

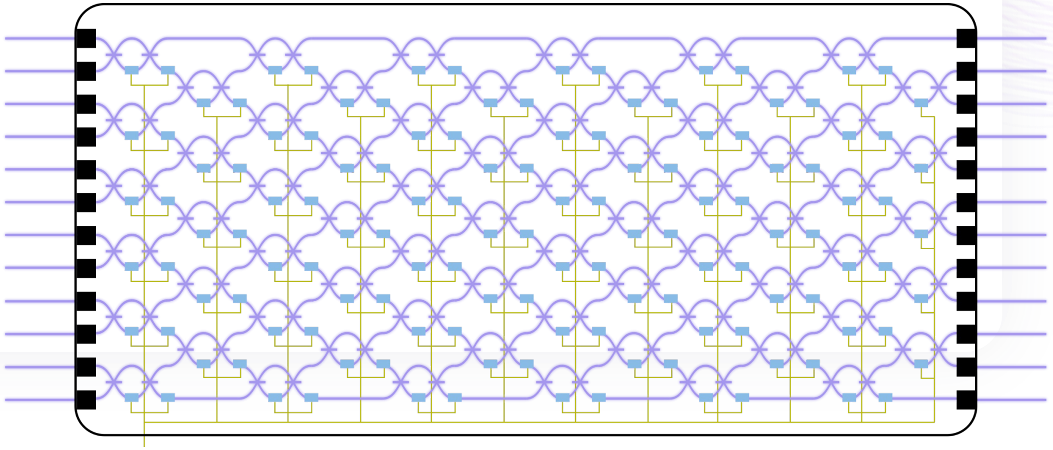
The phase-shifter:



$$\begin{bmatrix} E_C \\ E_D \end{bmatrix} = \begin{bmatrix} e^{i\varphi} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} E_A \\ E_B \end{bmatrix}$$

(more generally any 1-mode unitary is a phase-shifter)

- A **universal interferometer** can implement any unitary transformation

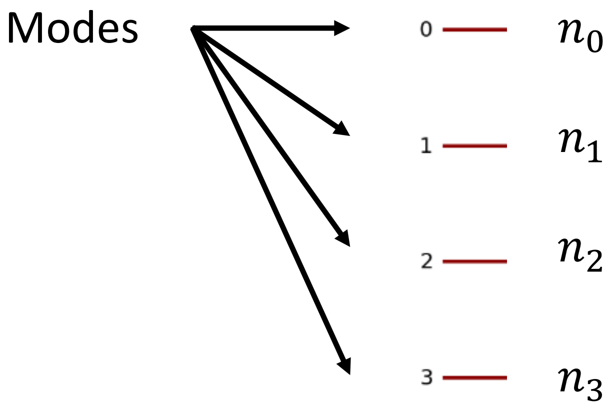


Computing with light

The Mathematics – 101

- Input/Output State is represented by a Fock state

$$|s\rangle = |n_0, n_1, n_2, n_3\rangle$$



- Quantum State is given by:

  $s = \sum_i \alpha_i |s_i\rangle$ where α_i is the **probability amplitude** of $|s_i\rangle$

- Fock space size is $M_n = \binom{n+m-1}{m-1} = \frac{(n+m-1)!}{n!(m-1)!} \sim 4^n$ (for m=2n)
- Probability amplitude of a specific circuit output given a specific unitary is given by **<u>Permanent</u>** of the scattering matrix:

$$\langle t|U|s\rangle = \frac{perm\left(U_{|s\rangle,|t\rangle}\right)}{\sqrt{s_1!, \dots s_m!\, t_1!, \dots t_m!}}$$

Heurtel, Nicolas, et al. "Strong simulation of linear optical processes." Computer Physics Communications 291 (2023): 108848.

Computing with light

Calculating the Permanent

What does a Photonic QPU can do

- A QPU performs sampling tasks
- **Sampling distribution is driven by a mathematical operation on a matrix called "Permanent"**

$$perm(U) = \sum_{\sigma \in S(n)} \prod_{i=1}^{n} u_{i,\sigma(i)}$$

- **Calculating the Permanent is #P-hard !**

The Permanent

1	2	3
a	b	c
d	**e**	f
g	h	**i**

1	3	2
a	c	b
d	**f**	e
g	i	**h**

2	1	3
b	a	c
e	**d**	f
h	g	**i**

2	3	1
b	c	a
e	**f**	d
h	i	**g**

3	1	2
c	a	b
f	**d**	e
i	g	**h**

3	2	1
c	b	a
f	**e**	d
i	h	**g**

$$aei + afh + bdi + bfg + cdh + ceg =$$

(17 operations)

With some optimizations:

$$a(ei + fh) + b(di + fg) + c(dh + eg)$$

(14 operations)

Best known algorithm requires $\boldsymbol{O(n.2^n)}$ operations !

Computing with light

Calculating the Permanent - How big is $n.2^n$?

Time necessary to collect/simulate 1000 samples on n photons:

n	Number of operations per sample	High Performance Laptop	Nvidia H100	Jean Zay HPC #274 worldwide	1GHz QPU with 80% transmission	1GHz QPU with 90% transmission	Ideal QPU
4	64	milliseconds	milliseconds	milliseconds	milliseconds	milliseconds	milliseconds
10	10240	seconds	milliseconds	milliseconds	milliseconds	milliseconds	milliseconds
20	21M	minutes	seconds	milliseconds	milliseconds	milliseconds	milliseconds
30	32B	days	hours	1s	milliseconds	milliseconds	milliseconds
48	3.10^{15}	months	weeks	100s	milliseconds	milliseconds	milliseconds
80	10^{26}	-	-	95 years	1 hour	1 second	milliseconds

What about qubits and GBQC?

- LOQC computing space is Fock space which includes the Hilbert space - the gate-based quantum computing model
- Size of the Fock space depends on the number of modes and the number of photons
- For dual-rail encoding size of the space grows as 4^n
- No decoherence – “noise” is at the source
- Simulation less demanding on memory but far more demanding on computing

Putting everything together

OVH cloud datacenter, Croix – France, 1st delivered QC system, November 2023.

QUANDELA

- About Quandela
- 101 Photonic Quantum Computing : how to build your own Quantum Computer!
- The Software Stack – exQalibur, MosaiqOS, **Perceval** and the Quantum Toolbox

The Software Stack

Developing solutions for real-life problems

OVHcloud

Quandela Cloud

Quantum as a Service

Run your Quantum algorithms on powerful and elastic hardware

Scaleway

Quandela Jousting Arena

Quandela Quantum Toolbox

EVIDEN myQLM

Qiskit

Graphix

…

Connectors

Perceval

exQalibur

MosaiqOS

Optimized Simulation and Compilation

Hardware Control and Transpilation

The Software Stack

Developing solutions for real-life problems

Quandela Cloud

Quandela Quantum Toolbox

Perceval

Why and What is Perceval?

A French legend…

Perceval, the story of the Grail (1180)

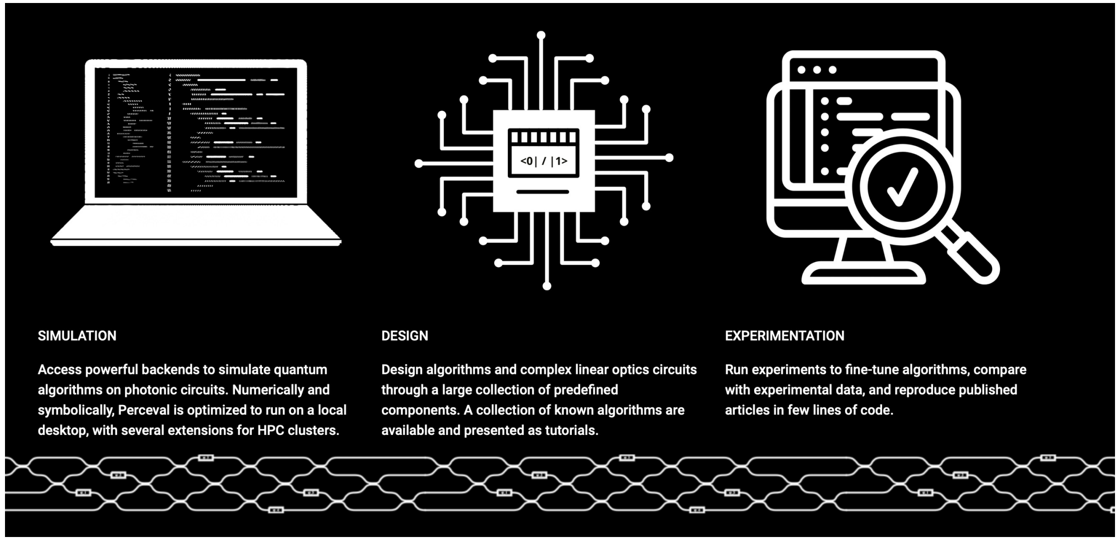
Perceval is the youngest of the king Arthur knights and is the only knight to have encountered the “Grail” described as a **super bright source of light.**

Chrétien de Troyes (1180)

Une si granz clartez i vint
Qu’ausi perdirent les chandoiles
Lor clarté come les estoiles
Quant li solauz lieve ou la lune.

Perceval (2022)

A Software Platform for **Discrete Variable** Photonic Quantum Computing



Perceval entry points

Getting started with Perceval

Detailed walkthrough

ADVANCED TUTORIALS

Error-tolerant BS-based circuit

LOv rewriting rules

Simulation of non-unitary components

Decomposing gate-based circuits: qiskit and myQLM

Remote computing on Quandela Cloud

Graph States generation and display

Tomography of a CNOT Gate

BOSON SAMPLING

Boson Sampling

MPS techniques for Boson Sampling

STANDARD QUANTUM ALGORITHMS

Shor's algorithm implementation

2-mode Grover's search algorithm

VARIATIONAL QUANTUM ALGORITHMS

Differential equation resolution

Variational Quantum Eigensolver

Reinforcement learning

The shortest path problem using QUBO

QUANTUM WALK

Two-particle bosonic-fermionic quantum walk

Building an array of beam splitters

Single photon quantum walk

Two photons quantum walk

OTHERS

Gedik's algorithm with qudit encoding

Two-particle bosonic-fermionic quantum walk   View page source

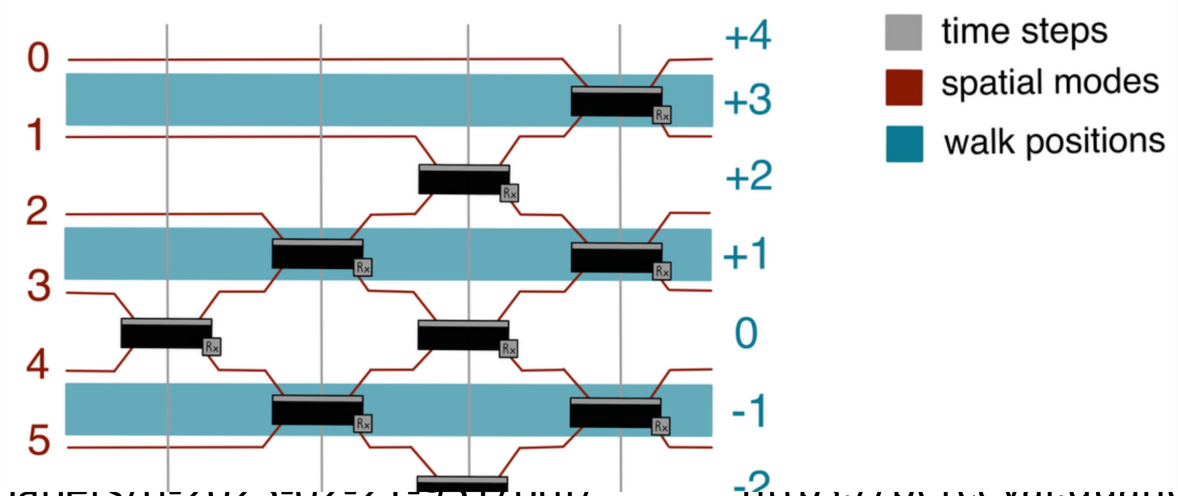
Two-particle bosonic-fermionic quantum walk

We provide an implementation of the two-particle quantum walk. The aim is to reproduce the results of "Two-particle bosonic-fermionic quantum walk via integrated photonics" by L. Sansoni et al. [1] with Perceval.

[5]:
```
# imports
import matplotlib.pyplot as plt
import matplotlib as mpl

import numpy as np

import perceval as pcvl
from perceval.components.unitary_components import BS
from perceval.backends import NaiveBackend
from perceval.simulators import Simulator
from perceval.components import Source

## Use the symbolic skin for display
from perceval.rendering.circuit import DisplayConfig, SymbSkin
DisplayConfig.select_skin(SymbSkin)
```

Building an array of beam splitters

The dynamics of a quantum walk can be achieved by an array of beam splitters (BSs) as in figure. Here we reproduce a four steps quantum walk, we highlight the difference between the optical spatial modes (in red) and the walk positions (in blue).



https://

Quandela / Perceval

Code Issues 8 Pull reque

Perceval Public

main 6 branc

ericbrts PCVL-397 Fix

.github
benchmark
data
docs
perceval
scripts
tests
.gitignore
.pre-commit-config.ya
CODE_OF_CONDUCT.
CONTRIBUTING.md
LICENSE

Perce
Phot

Nicolas
Sébastie
Pierre-E
Benoît V

https://quantum-journal.org/papers/q-2023-02-21-931/pdf/

PERCEVAL

Edit on GitHub

Une si granz clartez i vint
Qu'ausi perdirent les chandoiles
Lor clarté come les estoiles
Quant li solauz lieve ou la lune.
Perceval, the Story of the Grail – Chrétien de Troyes (circa 1180)

both the ideal and realistic behaviours,

ks or local development environments.

s, and practitioners of quantum

New Topic

Activity

13d

16d

Apr 26

Apr 2

Mar 4

Feb 26

Feb 23

https://perceval.quandela.net/forum/

Main concepts in Perceval

Hello World!

```
[1]: import perceval as pcvl
```

[pcvl] is a good short phonetic Perceval namespace shortcut

```
[2]: help(pcvl)
```

```
Help on package perceval:

NAME
    perceval

DESCRIPTION
    Through a simple object-oriented python API, Perceval provides tools for building a circuit with linear optics
    components, defining single-photon sources and their error model, manipulating Fock states, running simulations,
    reproducing published experimental papers results and experimenting a new generation of quantum algorithms.

    It is interfaced with the available QPUs on https://cloud.quandela.com, so it is possible to run computations on an
    actual photonic computer.

    Perceval aims to be a companion tool for developing discrete-variable photonic circuits
      - while simulating their design, modeling their ideal and real-life behaviour;
      - and proposing a normalized interface to control photonic quantum computers;
      - while using powerful simulation backends to get state-of-the-art simulation;
      - and also allowing direct access to the QPUs of Quandela.

    See also:
      - Perceval user documentation: https://perceval.quandela.net/docs/
      - Quandela cloud documentation: https://cloud.quandela.com/webide/documentation (requires a free account to access)
```

Read the doc !

Main concepts in Perceval

Hello World!

```
PACKAGE CONTENTS
    algorithm (package)
    backends (package)
    components (package)
    converters (package)
    rendering (package)
    runtime (package)
    serialization (package)
    simulators (package)
    utils (package)

[…]

VERSION
    0.9.1.post33+2023.8.22

FILE
    /Users/senellart/DEV/PercevalJean/perceval/__init__.py
```

No need to know about these packages, most of the useful objects are available in main namespace!

Check the latest stable version, see: https://github.com/Quandela/Perceval/releases

Main Concepts in Perceval (1)

From BasicStates to StateVectors

`StateVector`

Complex normalized linear combination of `BasicState`

`sqrt(2)/2*|0,1>+sqrt(2)/2*|1,0>`

$$\frac{\sqrt{2}}{2}(|0{,}1\rangle + |1{,}0\rangle)$$

`BasicState`

Annotated Fock state

`|0,1,2,3>`

`|{P:H},{P:V}>`

Annotations are used for “distinguishable” features – for instance polarization

You do need to know objects in **bold**, other objects are generally outputs of operators

`BSSamples`

Container that stores samples (unannotated `BasicState`*) in a time ordered way*

```
[|0,1>,
 |1,0>,
 |1,0>,
 |0,1>]
```

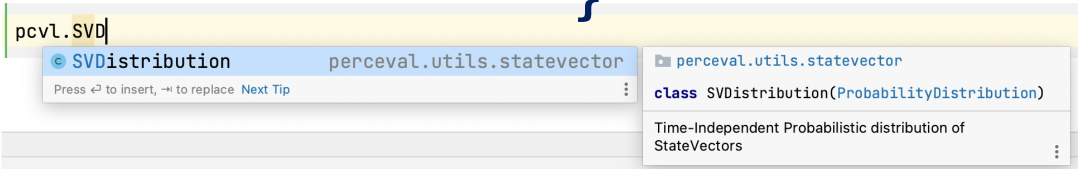
`SVDistribution`

`BSDistribution`

*Probabilistic distribution of BasicState/StateVector (**Mixed States**)*

```
{
  |0,1>: 0.5,
  |1,0>: 0.5
}
```

Check online help in your IDE!

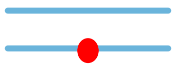


Main Concepts of Perceval (1)

From BasicStates to StateVectors

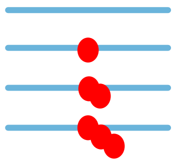
perceval.utils.BasicState

```
>>> s=pcvl.BasicState("|0,1>")
>>> print(s[0], s[1])
0 1
```

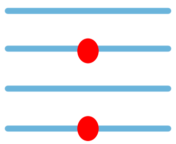


1 photon
2 modes

```
>>> print(s.n, s.m, s, list(s))
1 2 |0,1> [0,1]
>>> print(pcvl.BasicState([0,1])*
          pcvl.BasicState([2,3]))
|0,1,2,3>
```

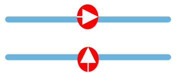


```
>>> pcvl.BasicState([0,1])**2
|0,1,0,1>
```



 # Using Annotations

```
>>> a_bs = pcvl.BasicState("|{P:H},{P:V}>")
```



```
>>> print(a_bs[0],a_bs[1], a_bs.clear())
1 1 |1,1>
```

perceval.utils.StateVector

```
>>> print(pcvl.BasicState([0,1])+
          pcvl.BasicState([1,0]))
sqrt(2)/2*|0,1>+sqrt(2)/2*|1,0>
```

Self-normalization!

```
>>> print(pcvl.StateVector([0,1])-
          2*pcvl.StateVector([2,3]))
sqrt(5)/5*|0,1>-2*sqrt(5)/5*|1,0>
```

$$= \frac{\sqrt{5}}{5}(|0,1\rangle - 2|1,0\rangle)$$

 # Sampling from StateVector:

```
>>> st = pcvl.StateVector([0,1]) +
         pcvl.StateVector([1,0])
>>> c = Counter()
>>> for s in st.samples(10):
...     c[s] += 1
>>> print(", ".join(["%s: %d" % (str(k), v)
                     for k,v in c.items()]))
|0,1>: 3, |1,0>: 7
```

Main Concepts of Perceval (2)

Measurement and Mixed State

Measuring a StateVector

```
>>> sv = pcvl.StateVector("|0,1,1>")
        +pcvl.StateVector("|1,1,0>")

>>> map_measure_sv = sv.measure(1)
>>> for s, (p, sv) in map_measure_sv.items():
...                 print(s, p, sv)
|1> 0.9999999999999998
sqrt(2)/2*|0,1>+sqrt(2)/2*|1,0>

>>> map_measure_sv = sv.measure(2)
>>> for s, (p, sv) in map_measure_sv.items():
...                 print(s, p, sv)
|0> 0.5000000000000001 |1>
|1> 0.5000000000000001 |0>
```

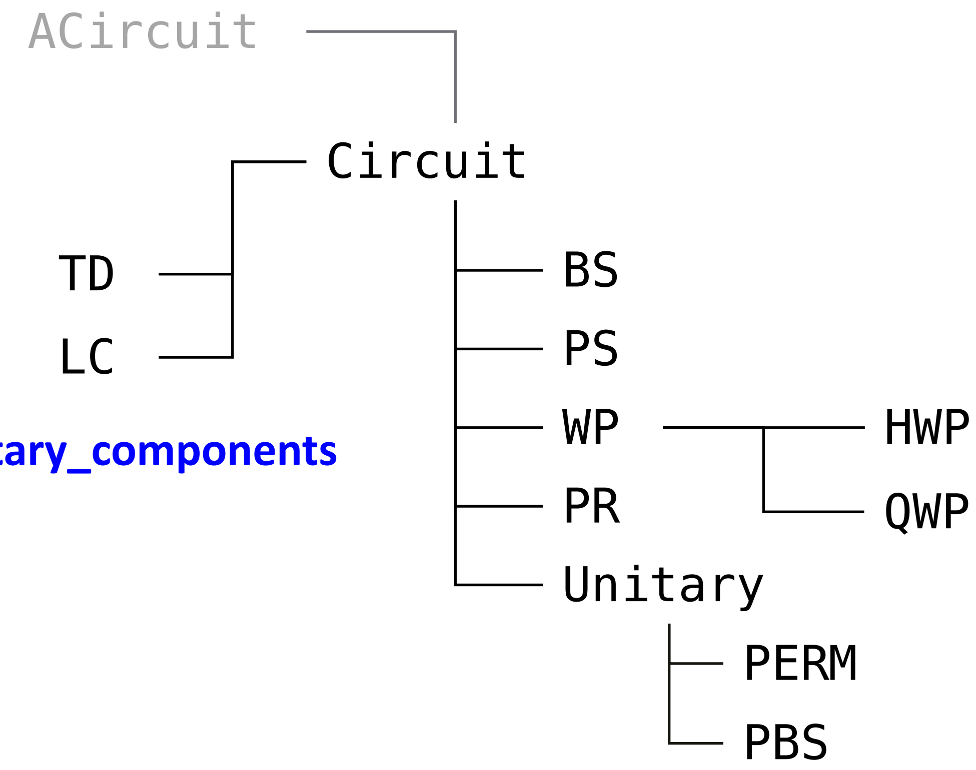
perceval.utils.SVDistribution

Used to generate mixed state (probabilistic combination of different StateVector)

state	probability
|0,1>	1/2
1/sqrt(2)*|1,0>+1/sqrt(2)*|0,1>	1/4
|1,0>	1/4

Main Concepts of Perceval (3)

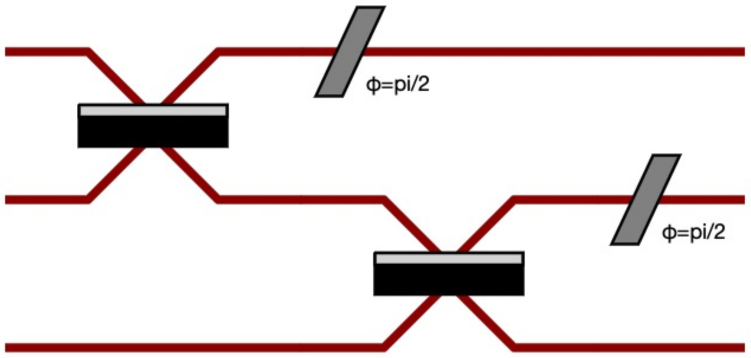
Components and Circuits



.non_unitary_components

pcvl.components.unitary_components

```
>>> bsps = comp.BS()
              .add(0, comp.PS(np.pi/2)
>>> c = pcvl.Circuit(3)
             .add(0, bsps).add(1, bsps)
>>> pcvl.pdisplay(c)
```



$$\begin{bmatrix} \mathrm{sqrt}(2)/2 & \mathrm{sqrt}(2)*I/2 & 0 \\ I/2 & 1/2 & \mathrm{sqrt}(2)*I/2 \\ -1/2 & I/2 & \mathrm{sqrt}(2)/2 \end{bmatrix}$$

Main Concepts of Perceval (4)

Backends and Processors

Strong Simulator – provide probability amplitude

Approximate Simulation

Weak Simulator – provide Sampling

Features Name	CliffordClifford2017	SLOS	Naive	Stepper	MPS
Sampling Efficiency	$O(n2^n + poly(m,n))$	$O(mC_n^{n+m-1})$	N/A [1]	N/A [1]	N/A [1]
Single output Efficiency	N/A	N/A	$O(n2^n)$	$o(N_c C_n^{n+m-1})$	$o(N_c C_n^{n+m-1})$
Full Distribution Efficiency	N/A	$O(nC_n^{n+m-1})$	$O(n2^n C_n^{n+m-1})$	$o(N_c C_n^{n+m-1})$	$o(N_c C_n^{n+m-1})$
Probability Amplitude	No	Yes	Yes	Yes	Yes
Support Symbolic Computation	No	Yes	No	Yes	No
Support of Time-Circuit	No	No	No	Yes	No
Practical Limits	$n \approx 30$	$n, m < 20$	$n \approx 30$		

Main Concepts of Perceval (4)

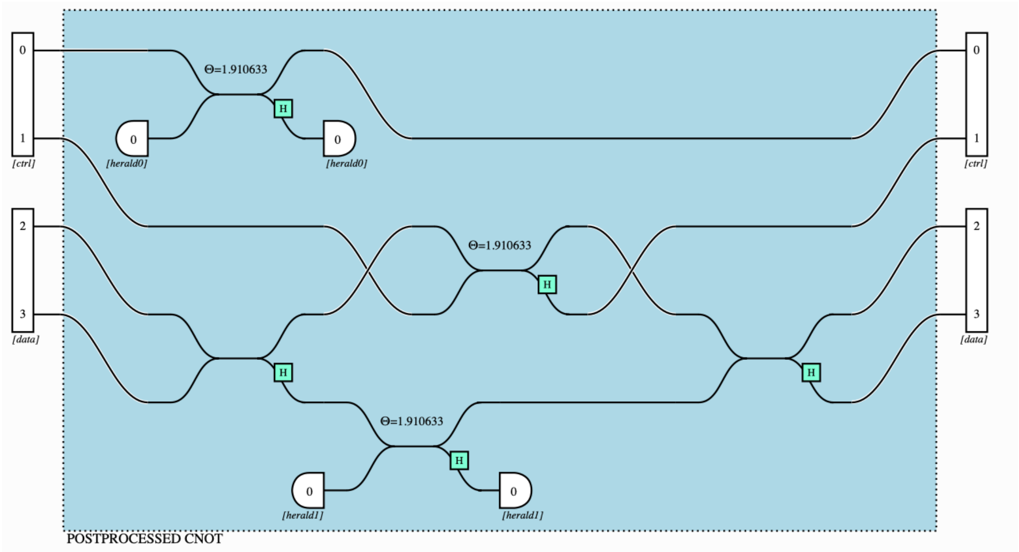
Backends and Processors

Definition

- A Backend is either a hardware or a specific simulation algorithm
- A Processor is an end-to-end access to a virtual or real QPU
  - Processor models
    - source
    - Backend
    - encoding logic (through ports)
    - optional postprocessing logic (heralds)
  - Processor can be either local (local simulation) or Remote

Example

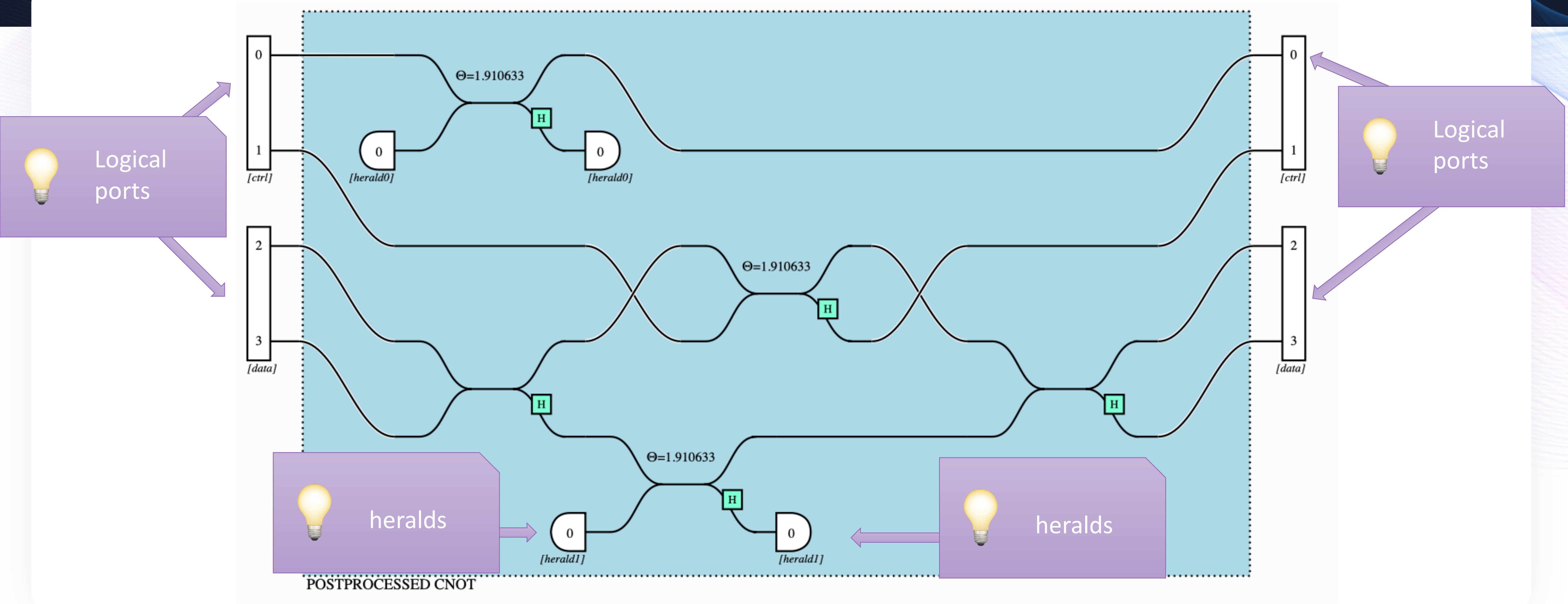
```
>>> cnot = catalog['postprocessed cnot']
                              .build_processor()
>>> pcvl.pdisplay(cnot, recursive=True)
```




```
>>> cnot.with_input(pcvl.LogicalState([1, 0]))
```

Main Concepts of Perceval (4)

Processor example



Quantum Toolbox

Let us forget about code !

Toolboxes

VQE chemistry	$H = \sum h_\alpha P_\alpha$ VQE custom	CVaR-VQE	Graph isomorphism	Dense Subgraph identification
Compute **ground state** of BeH, LiH, H2O, H2 on **QPU**	Compute ground state of **user-chosen Hamiltonian**	VQE variant to solve **combinatorial optimisation problem**	**Boson sampling** based algorithm to **check whether two graphs are isomorphic**	**Boson sampling** based algorithm to **identify dense subgraphs**

Quantum Toolbox going live

5 easy to use algorithms to unlock tens of use-cases

Toolboxes

VQE chemistry	VQE custom ($H = \sum h_\alpha P_\alpha$)	CVaR-VQE	Graph isomorphism	Dense Subgraph identification
Compute **ground state** of BeH2, LiH, H2O, H2 on **QPU**	Compute ground state of **user-chosen Hamiltonian**	VQE variant to solve **combinatorial optimisation problem**	**Boson sampling based** algorithm to **check whether two graphs are isomorphic**	**Boson sampling** based algorithm to **identify dense subgraphs**

Use-cases

VQE chemistry	VQE custom	CVaR-VQE	Graph isomorphism	Dense Subgraph identification
Battery design : - Compute **force field on molecules** - Compute **binding energies**	Try to compute **new molecules ground states** on a photonic QPU Solve **differential equation** using energetic formulation	**Multi-agent path finding:** For logistics issues in warehouses **Train-unit assignment problem:** to minimize number of carriages	**Fault detection in chip design :** defect-free chip should be isomorphic to graph of perfect chip **Cross-checking databases in cheminformatics**	**Drug design :** Docking problem i.e. finding optimal configuration for ligand-receptor couple

Five toolboxes unlocking tens of use cases

Ground state energies of large molecules

Toolbox

VQE chemistry

Compute **ground state** of BeH2, LiH, H2O, H2 on **QPU**

Use case

Problem : Computing ground state of large molecules becomes quickly infeasible

Solution : Active space method such as **DMET-VQE** enables to separate a molecule in fragments and compute its ground state energy

Benefits : it will enable **more accurate and faster predictions** of ground state energies of molecules, which in turn enables to compute force field, binding energies **for material design**

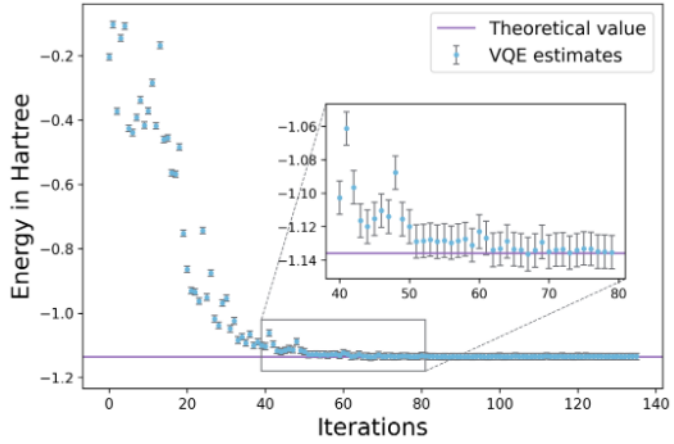


Fig. Energy vs iterations for H2 molecules with given bond length

Five toolboxes unlocking tens of use cases

Predicting behaviour of mechanical structure (dams, nuclear pipes)

Toolbox

$H = \sum h_\alpha P_\alpha$ VQE custom

Compute ground state of **user-chosen Hamiltonian**

Use case

Problem : solving PDEs **represents >50% of HPC usage of EDF**, crucial for mechanical structure, electricity bill is consequential

Solution : EDF and Quandela co-developed a variational algorithm based on an energetic formulation

Benefits : quantum algorithm scales better (poly-logarithmic) than classical state-of-the-art. It will **reduce energy consumption** when solving PDEs.

eDF

Five toolboxes unlocking tens of use cases

Directing multiple robots in a warehouse without collisions

Toolbox

CVaR-VQE

VQE variant to solve **combinatorial optimisation problem**

Use case

Problem : multi-agent path finding notorious **NP-hard problem**. Useful in **logistics**, drones traffic management, etc. Only heuristics to solve it and limited to few agents on medium-size graphs.

Solution : Quandela co-developed with a client the energetic formulation of that problem to solve it using CVaR-VQE.

Benefits : as QPU size grows, this algorithm should be able to tackle instances where **number of agents, locations and constraints are greater** compared to **classical solvers**

Five toolboxes unlocking tens of use cases

Assigning minimal amount of train units to trips

Toolbox

CVaR-VQE

VQE variant to solve **combinatorial optimisation problem**

Use case

Problem : train-unit assignment problem is a notorious NP-hard problem. Useful in rolling stock circulation phase. Crucial economically no minimize the number of train-units used as one train-unit costs millions of euros.

Solution : Quandela co-developed with a partner the energetic formulation of that problem to solve it using CVaR-VQE.

Benefits : solving large instances in short time of TUAP is beyond the reach of classical devices. This approach provides a **quantum heuristics sample-efficient and resilient to noise**.

Five toolboxes unlocking tens of use cases

Cross-checking chemical description

Toolbox

Graph isomorphism

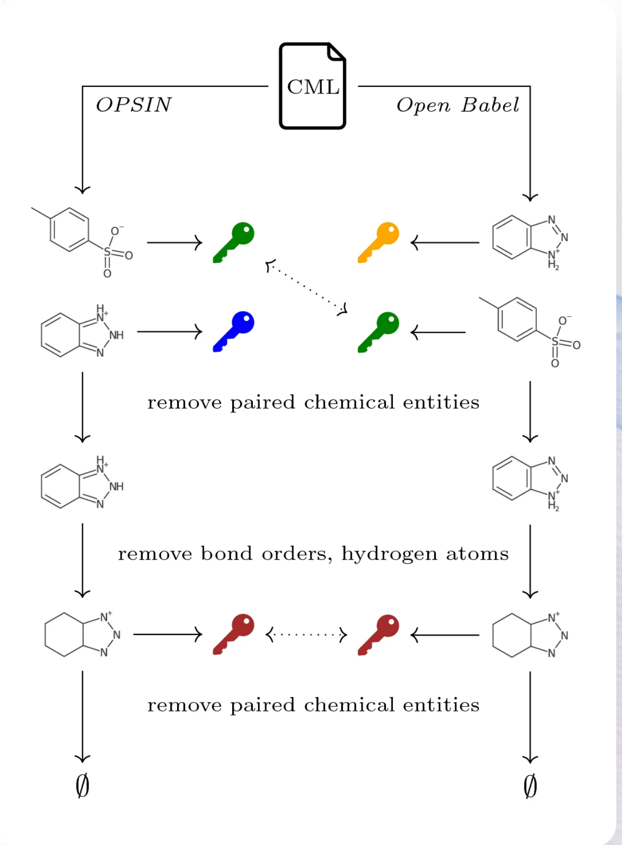
Boson sampling based algorithm to **check whether two graphs are isomorphic**

Illustrative Use case

Problem : various machine-readable entries for chemical molecules . How to compare those since different basis of representation were used ?

Solution : represent each **machine-readable entry with keys**, then we compare keys, if they **don't correspond make simplification** on original data to see **if keys now match**. If so, we detect **differences in notations** or retrieve **chemical experimental information** about a dataset.

Benefits : graph isomorphism problem is a really hard problem (NP), with no efficient classical solution. Some heuristics are fast but not exact. Our proposed approach provides a **quantum heuristics exploiting hardness of boson sampling that should provide a speed-up when scaling up.**



Ref : Merkys, A., Vaitkus, A., Grybauskas, A. *et al.* Graph isomorphism-based algorithm for cross-checking chemical and crystallographic descriptions. *J Cheminform* **15**, 25 (2023). https://doi.org/10.1186/s13321-023-00692-1

Five toolboxes unlocking tens of use cases

Molecular docking

Toolbox

Illustrative Use case

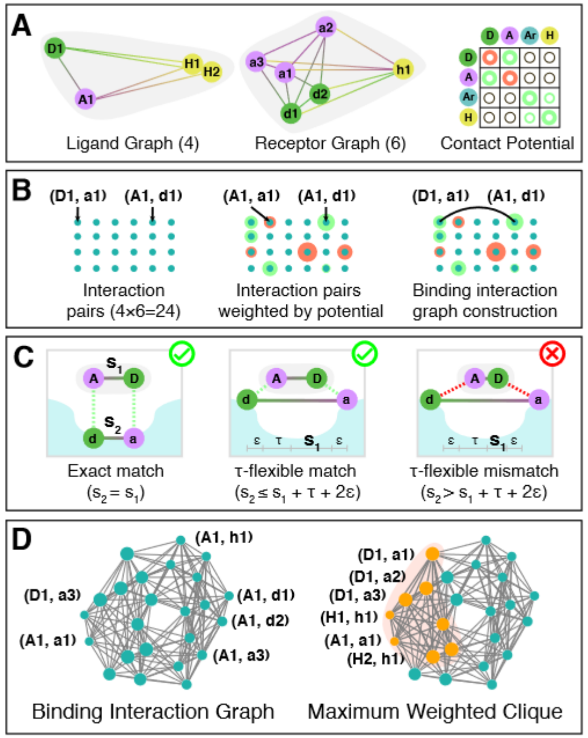
Dense Subgraph identification

Boson sampling based algorithm to **identify dense subgraphs**

Problem : Molecular docking is **central in structural molecular biology and computer-assisted drug design**. The goal is to predict **how a ligand** (small molecule) **will dock to a protein** (large molecule). There is a vast number of ways to do so, which makes is a **hard problem for classical computers.**

Solution : map the ways a ligand docks to a protein (step A on the right panel) to a graph called a binding interaction graph (step D on the right panel). Then using Quandela quantum computer one can find the maximum weighted clique, i.e. the optimal solution for a ligand to dock with a protein.

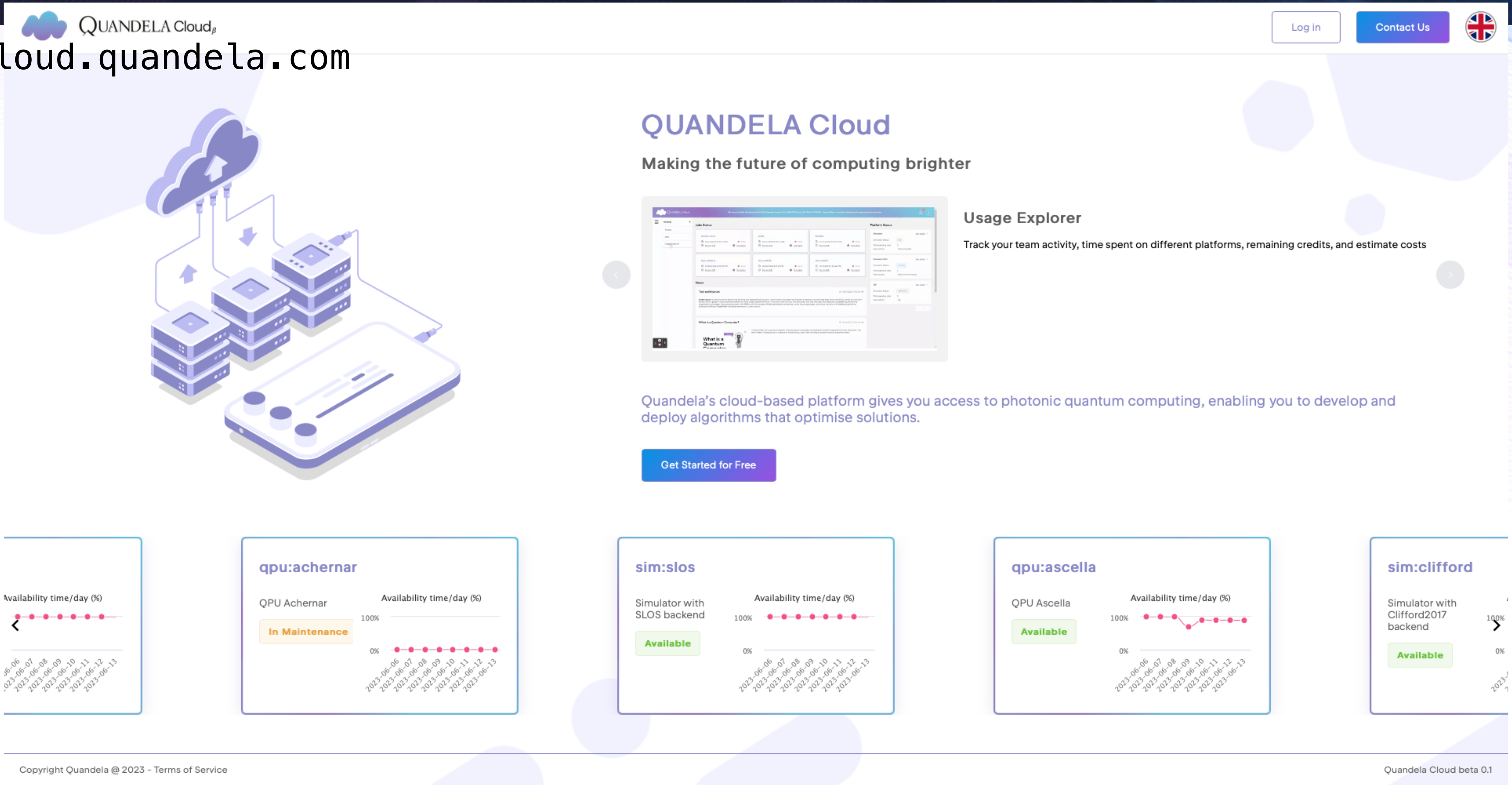
Benefits : molecular docking has no efficient classical solution. Our proposed approach provides a **quantum heuristics exploiting hardness of boson sampling that should provide a speed-up when scaling up.**



•Ref : Leonardo Banchi *et al.*, Molecular docking with Gaussian Boson Sampling.*Sci. Adv.***6**,eaax1950(2020).DOI:10.1126/sciadv.aax1950

Quandela Cloud

https://cloud.quandela.com



QUANDELA Cloud NEW RELEASE 2.0

6 j

OVERVIEW

- Dashboard

APPLICATIONS

- Notebook
- Tokens
- Jobs
- Platforms
- Quantum Toolbox

BILLING

- Usage Explorer

USER PROFILE

- Information
- Password
- Setting

USER GUIDE

- Cloud Documentation

14

Platforms

View all ↗

sim:aws:batch_spot

System type
simulator
Pending jobs 0
Commands
sample_count

sim:aws:batch

System type
simulator
Pending jobs 0
Commands
sample_count

sim:gpu:medium

System type
simulator
Pending jobs 0
Commands
samples
sample_count

sim:gpu:small

System type
simulator
Pending jobs 0
Commands
samples
sample_count

Recent Jobs

View all ↗

0 pending **0** running **9 312** completed

Job Name	Platform	Status	Created time	Duration
sample_count	sim:sampling:v100	Completed	24/04/2024 10:43:32	00:39
sample_count	sim:sampling:v100	Completed	24/04/2024 10:43:31	00:39
sample_count	sim:sampling:v100	Completed	23/04/2024 21:02:28	00:40
sample_count	sim:sampling:v100	Completed	23/04/2024 21:02:27	00:39

What's new

View all ↗

Announcement

Replay the Highlights: Quandela Cloud 2.0 Meetup Recap!

17/05/2024 · Read more ↗

Announcement

Introducing Quandela Cloud 2.0

06/05/2024 · Read more ↗

Announcement

Scheduled Maintenance on Ascella - Important Announcement

11/12/2023 · Read more ↗

Announcement

Replay LOQCathon 2.0 Highlights

29/11/2023 · Read more ↗

Announcement

The return of the LOQCathon. The 2.0 edition is here to unloqc even more challenges. What you need to know ?

10/10/2023 · Read more ↗

QUANDELA Cloud NEW RELEASE 2.0

6 ? j

OVERVIEW

Dashboard

APPLICATIONS

Notebook

Tokens

Jobs

Platforms

Quantum Toolbox

BILLING

Usage Explorer

USER PROFILE

Information

Password

Setting

USER GUIDE

Cloud Documentation

Jobs

Manage and view the status and results of all of the jobs you have run on Quandela's platform.

Create New | Filter | Columns

Search jobs by job name, token label | Exclude

	Action	Process ID	Job Name	Token	Priority	Platform	Command	Status
		00764310-4966-43b5-bb4f-a168f46ca879	sample_count	quarmen	Low	sim:sampling:v100	sample_count	Cor
		00764310-4966-43b5-bb4f-a168f46ca879	sample_count main	quarmen	Low	sim:sampling:v100	sample_count	Cor
		e33399b4-e601-440e-94b8-5aa88cc3dc19	sample_count	quarmen	Low	sim:sampling:v100	sample_count	Cor
		e33399b4-e601-440e-94b8-5aa88cc3dc19	sample_count main	quarmen	Low	sim:sampling:v100	sample_count	Cor
		5510cb53-8213-440c-a0ff-fd7b5feedc83	QPU sampling	class_10_token	Normal	qpu:ascella	sample_count	Cor
		5510cb53-8213-440c-a0ff-fd7b5feedc83	My sampling job main	class_10_token	Normal	sim:ascella	sample_count	Cor
		38cb04b2-e382-4619-8f1d-02c67526	QPU sampling	class_10_token	Normal	qpu:ascella	sample_count	Cor

Quandela Web Api

Overview

ENDPOINTS

Get Swagger Json Docs GET
Auth
Job
Auth App
QuantumToolbox
Chemistry VQE POST
Chemistry VQE Results GET
Custom VQE POST
Custom VQE Results GET
CVar VQE POST
CVar VQE Results GET
Graph DSI POST
Graph DSI Results GET
Graph Isomorphism POST
Graph Isomorphism Results GET
VQE POST
VQE Results GET

SCHEMAS

ApiKey
Atom
Atom1
AuthAppCreate

CVar VQE

POST `https://api.cloud.quandela.com/qt/cvarvqe`

CVar VQE input API

Request

> Security: Bearer Auth

Body

application/json

`max_iterations` integer or null
>= 1
Default: null

`platform_name` string — required
Example: sim:ascella

`qubo_matrix` array[array] — required
<= 15 items
Example: [[31,-500],[-500,32]]

- number

Responses

200 401 422

Successful response

Auth

Token : 123

Body

```
{
  "max_iterations": null,
  "platform_name": "sim:ascella",
  "qubo_matrix": [
    [
      31,
      -500
    ],
    [
      -500,
```

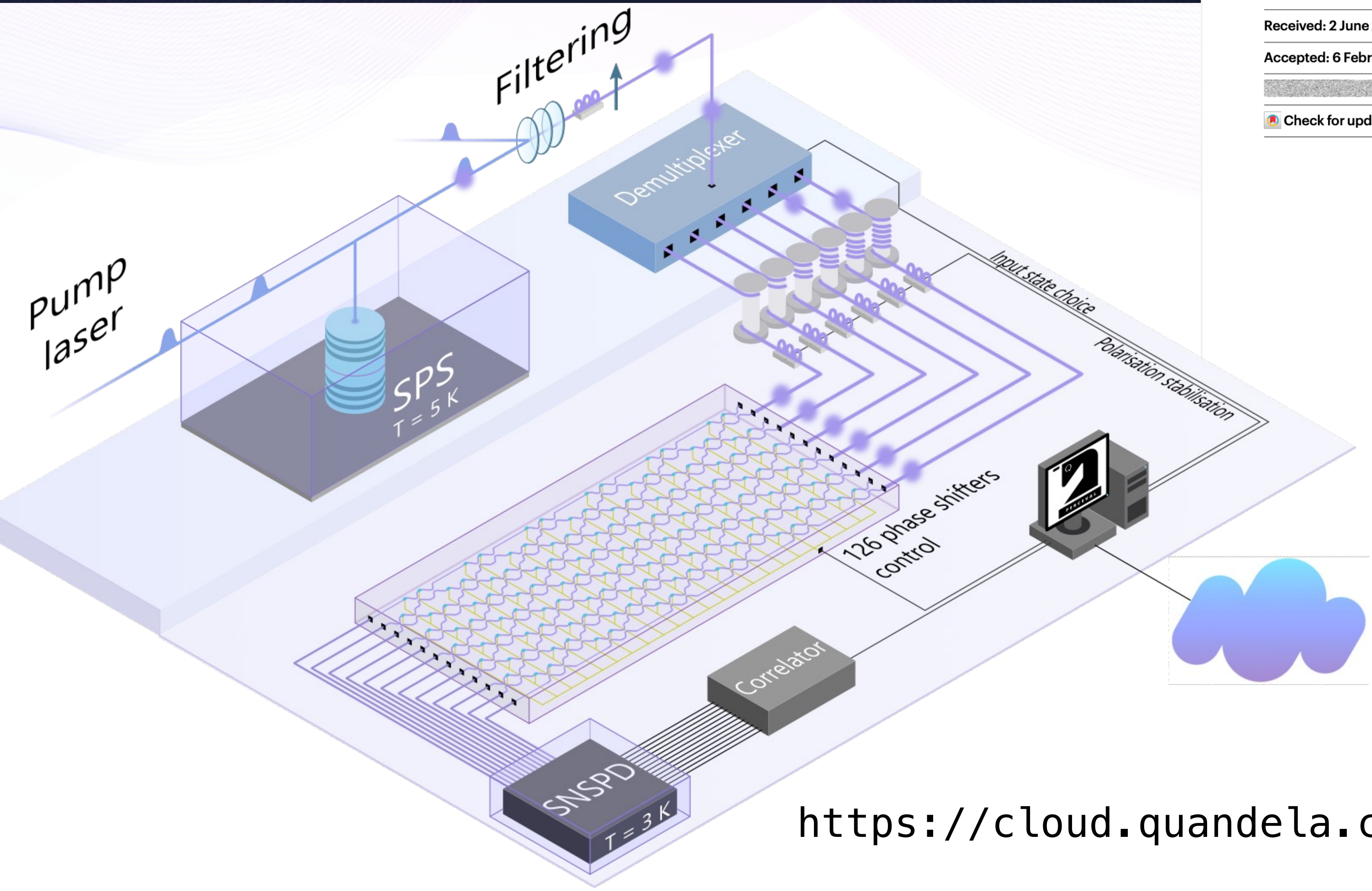
Send API Request

Request Sample: Shell / cURL

```
curl --request POST \
  --url https://api.cloud.quandela.com/qt/cvarvqe \
  --header 'Accept: application/json' \
  --header 'Authorization: Bearer 123' \
  --header 'Content-Type: application/json' \
  --data '{
  "max_iterations": null,
  "platform_name": "sim:ascella",
  "qubo_matrix": [
    [
      31,
```

Today at Quandela

From Ascella to Diadem



https://cloud.quandela.com

nature photonics

Article

https://doi.org/10.1038/s41566-024-01403-4

A versatile single-photon-based quantum computing platform

Received: 2 June 2023

Accepted: 6 February 2024

Check for updates

Nicolas Maring[1], Andreas Fyrillas[1,3], Mathias Pont[1,2,3], Edouard Ivanov[1,3], Petr Stepanov[1], Nico Margaria[1], William Hease[1], Anton Pishchagin[1], Aristide Lemaître[2], Isabelle Sagnes[2], Thi Huong Au[1], Sébastien Boissier[1], Eric Bertasi[1], Aurélien Baert[1], Mario Valdivia[1], Marie Billard[1], Ozan Acar[1], Alexandre Brieussel[1], Rawad Mezher[1], Stephen C. Wein[1], Alexia Salavrakos[1], Patrick Sinnott[1], Dario A. Fioretto[2], Pierre-Emmanuel Emeriau[1], Nadia Belabas[2], Shane Mansfield[1], Pascale Senellart[2], Jean Senellart[1] & Niccolo Somaschi[1]

Quantum computing aims at exploiting quantum phenomena to efficiently perform computations that are unfeasible even for the most powerful classical supercomputers. Among the promising technological approaches, photonic quantum computing offers the advantages of low decoherence, information processing with modest cryogenic requirements, and native integration with classical and quantum networks. So far, quantum computing demonstrations with light have implemented specific tasks with specialized hardware, notably Gaussian boson sampling, which permits the quantum computational advantage to be realized. Here we report a cloud-accessible versatile quantum computing prototype based on single photons. The device comprises a high-efficiency quantum-dot single-photon source feeding a universal linear optical network on a reconfigurable chip for which hardware errors are

- 2022
  - Ascella
  - 6 qubits
- 2023
  - Altair – error mitigation
  - 8 qubits
- 2024
  - Bélénos – utility
  - 12 qubits
- 2025
  - Canopus - logical qubits
  - 24 qubits – w/ cluster states
- 2026
  - Deneb - Adaptive circuits

HPC-scale simulator?

Come and see first-ever demo tomorrow at 11am on AWS booth !

QUANDELA aqora

A quantum strategy for more robust networks

Optimising for the Champions!

INSEP eDF

QuantX