goto; copenhagen

GOTO Copenhagen 2019

Conference Nov. 18 - 20

Click ‘Rate Session’ to rate session and ask questions.

Follow us @gotocph

Fueling the Quantum Application Era with the Cloud

Murray Thom, VP Software and Cloud Services

GOTO Berlin, Oct 2019

D-Wave
The Quantum Computing Company™

Overview

- You have access to quantum computers today
- A simple example
- Other example applications
- Hybrid quantum programming
- Look inside a quantum computer – why is it different
- Leap – quantum cloud services

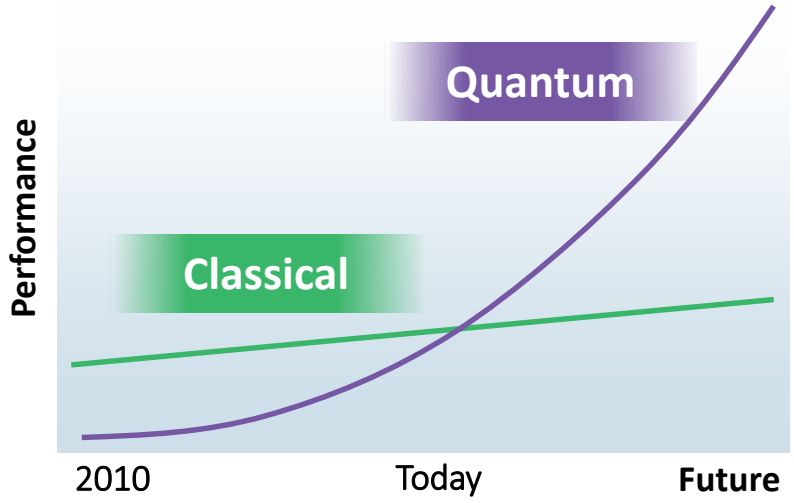
Why Quantum Computing?

Meeting the challenge of complex problems

Performance beyond the reach of Moore's Law

Transformative reduction in power consumption

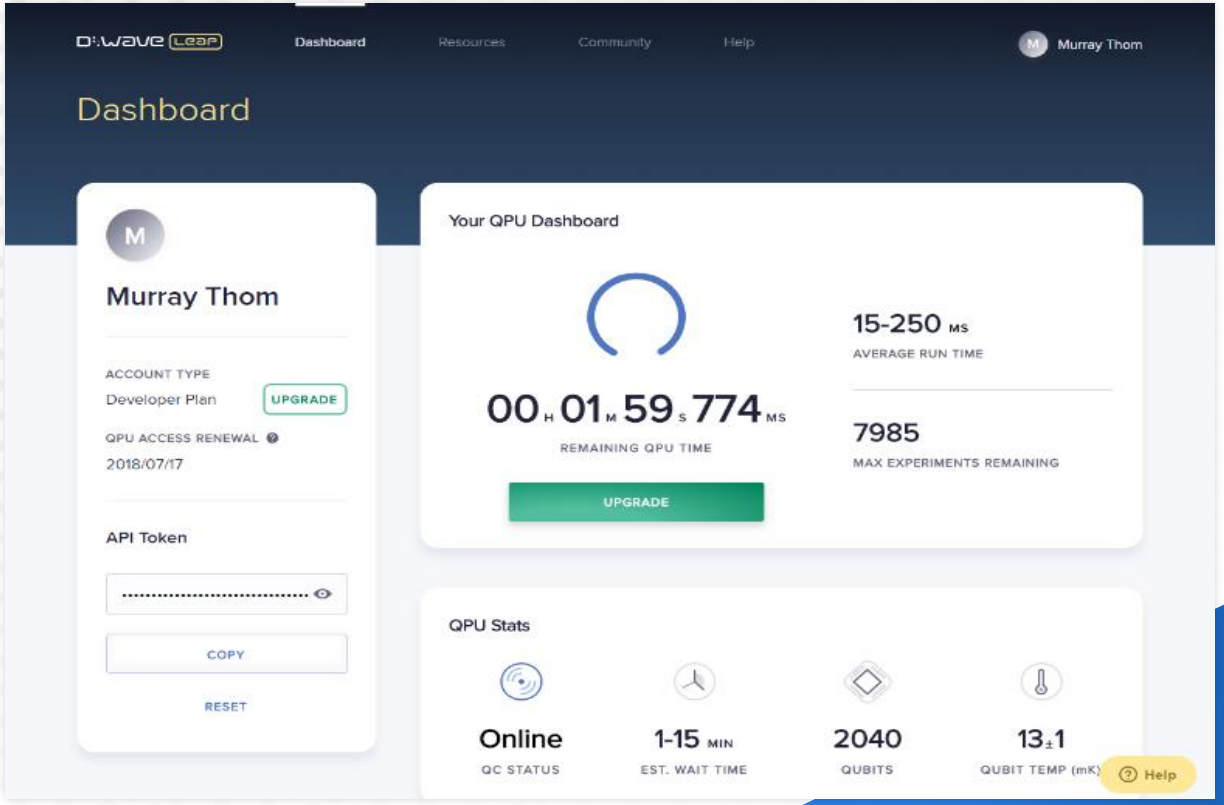
Quantum computing offers a radical new solution



D-Wave Leap™

Real-Time Cloud Access & Quantum Application Environment

Enabling a New Developer Community



- Free Real-Time Cloud Access
- Integrated Open Source SDK
- Demos and Reference Code
- Community Support
- Online Training Resources

What Is Most Sought After

Quantum Processing

Classical Input

Classical Output

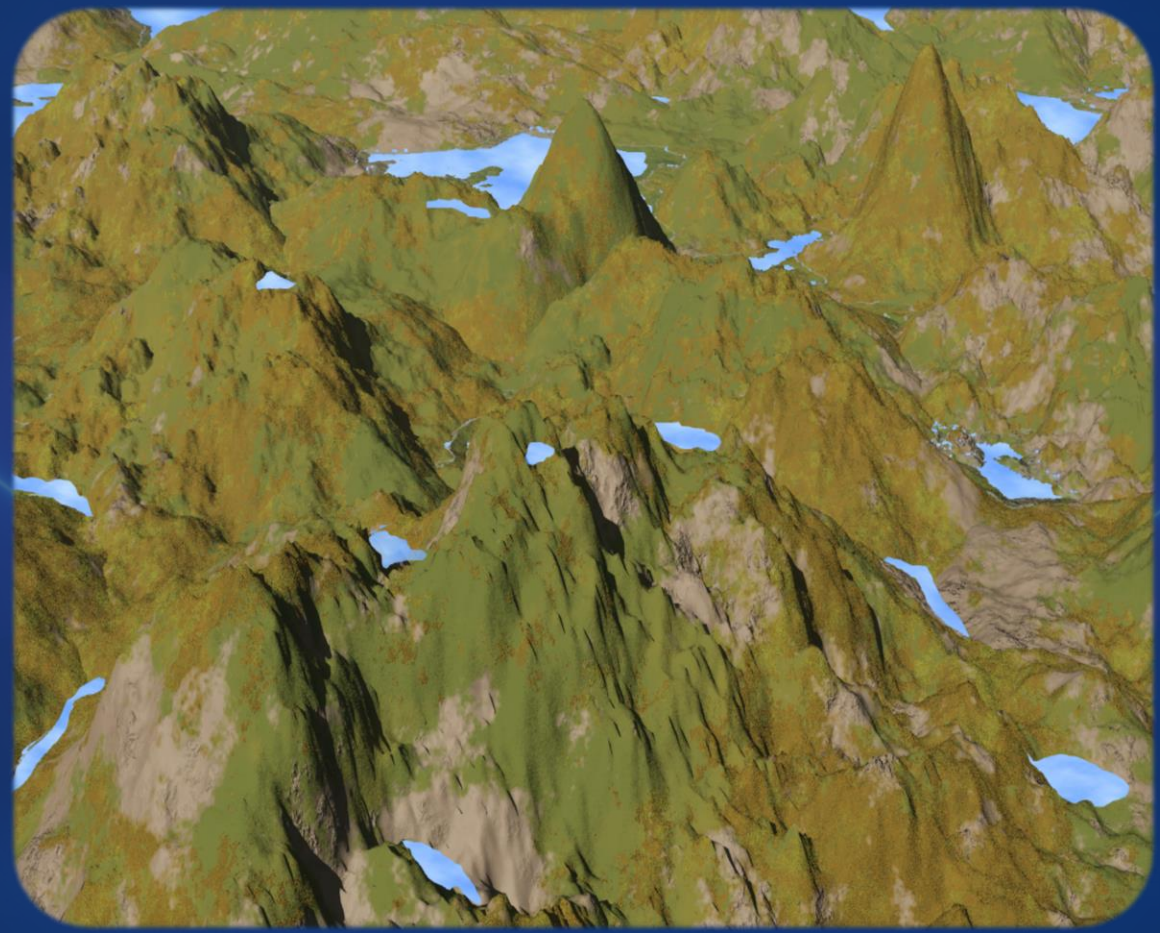
Credit to Ewin Tang
U Washington

Searching A Vast Solution Landscape

Problem: find the lowest point

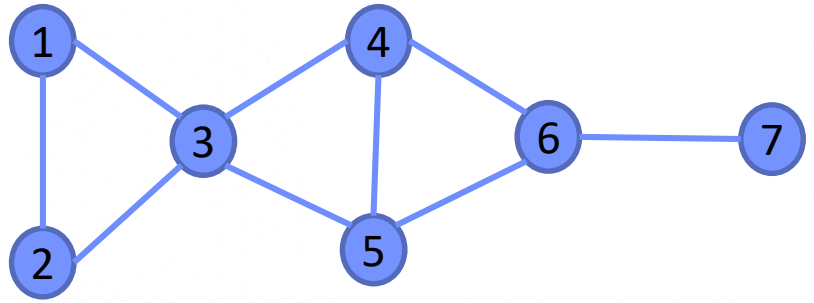
Classical solution: Run very fast, to each point

Quantum solution:
Visit many valleys at once, tunneling through the mountains

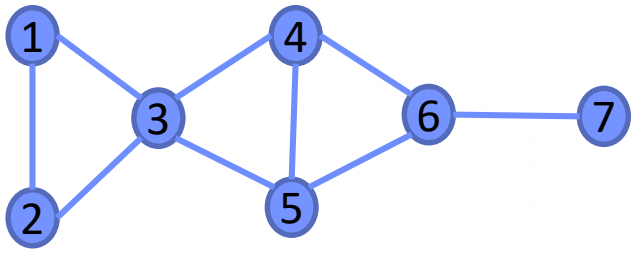


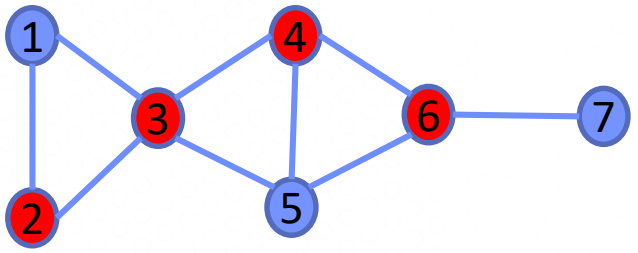
A Simple Example

- Given:
  - Network of pipelines
- What do we want:
  - A set of junctions from which we can monitor every pipeline segment
  - Minimize cost



Example





Junctions for monitoring pipes

Goal:

Nodes that *cover* every edge

A vertex cover.

Without Ocean tools...

1. Write our problem goals in mathematical form with binary variables

$$\text{minimize} \quad \sum_i x_i \qquad , x_i \in [0,1]$$

Minimize the number of monitoring sites

$$\text{such that} \quad \sum_{(u,v)\in E} (x_u \cdot x_v - x_u - x_v) < 0$$

Penalize cases where both ends of a pipe are unmonitored

2. Translate to a minimization formula that yields our objective

$$\text{minimize} \left(\sum_i x_i \right) + \gamma \left(\sum_{(u,v)\in E} (x_u \cdot x_v - x_u - x_v) \right) \qquad , \gamma \in \mathbb{R}$$

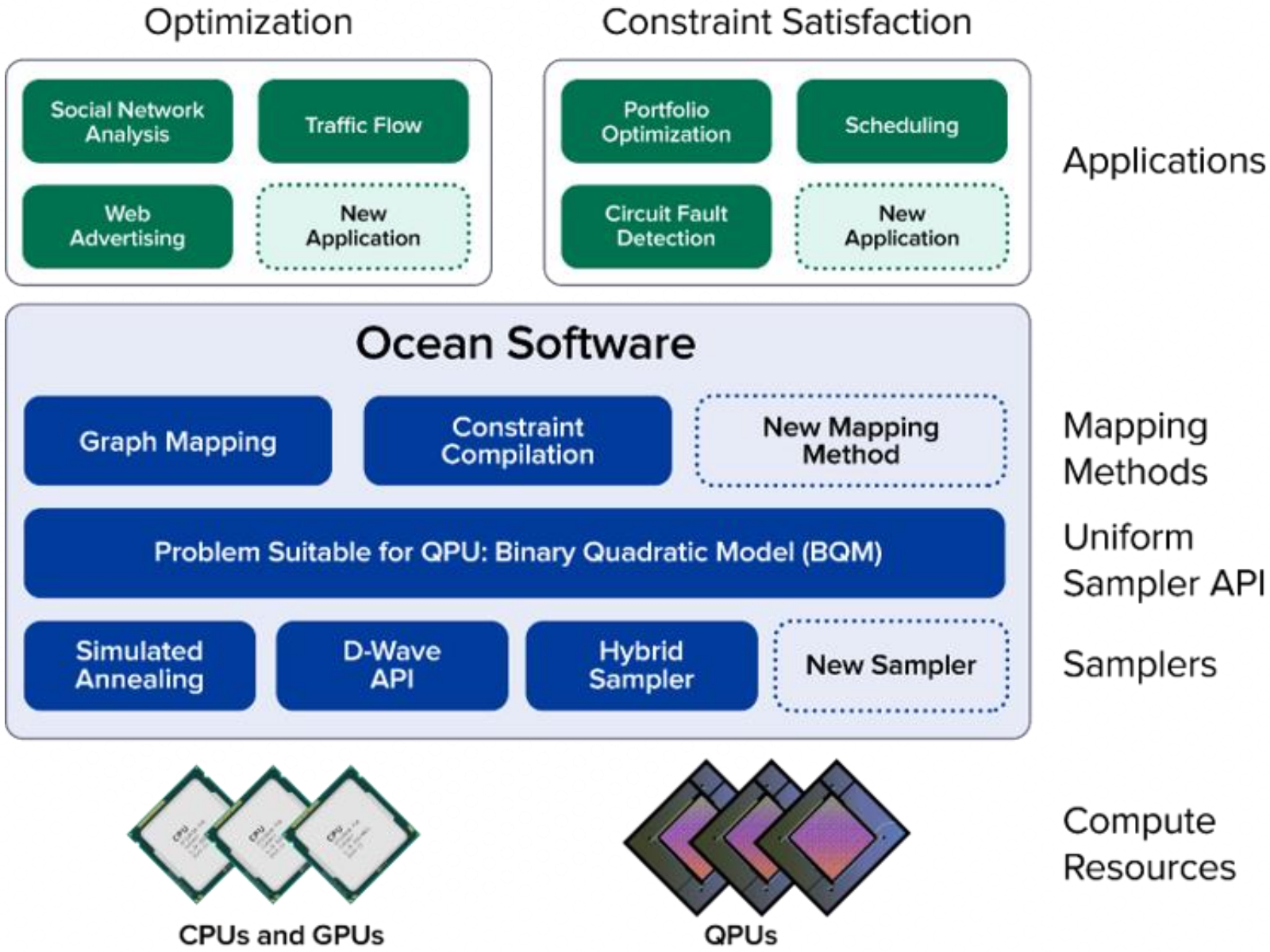
3. Simplify down to a matrix of coefficients from 2. (BQM matrix)

$$\begin{matrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & -3 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{matrix}$$

With the Ocean tools...

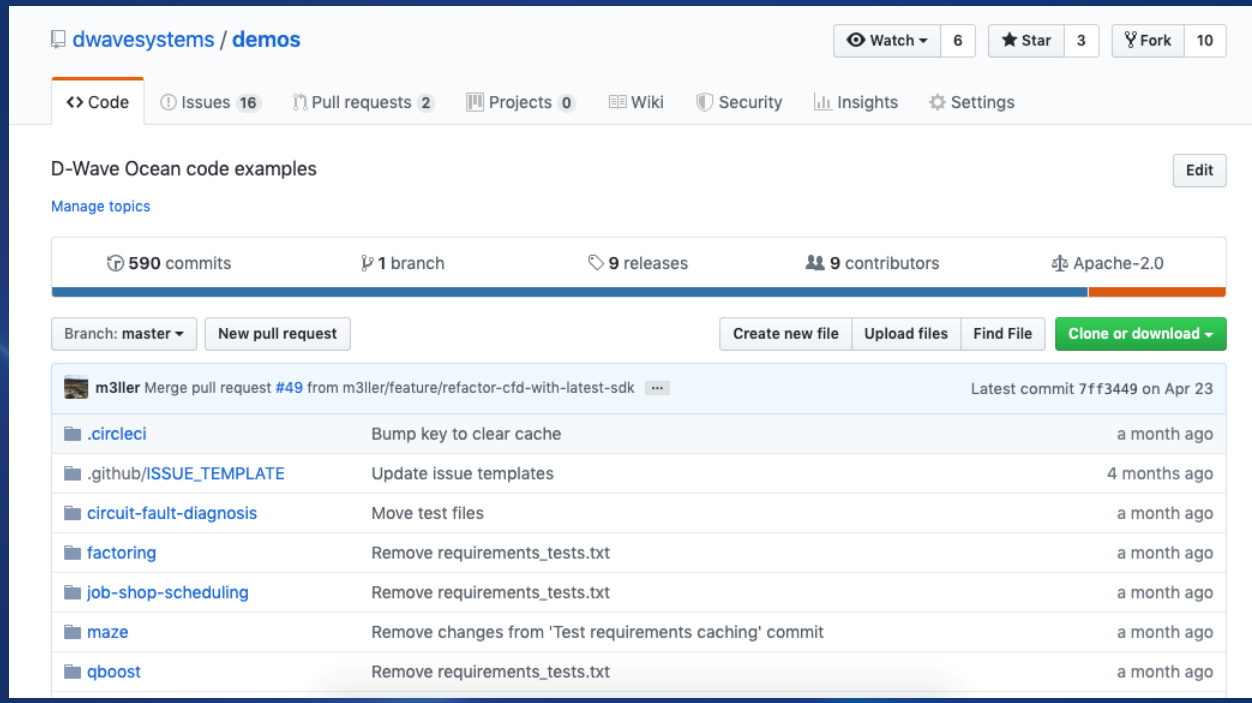
```
import networkx as nx

import dwave_networkx as dnx

from dwave.system import DWaveSampler, EmbeddingComposite


sampler = EmbeddingComposite(DWaveSampler())


G = nx.Graph()

G.add_edges_from([(1,2),(1,3),(2,3),(3,4),(3,5),(4,5),(4,6),(5,6),(6,7)])


cover = dnx.min_vertex_cover(G, sampler=sampler)
```

Ocean Software Stack



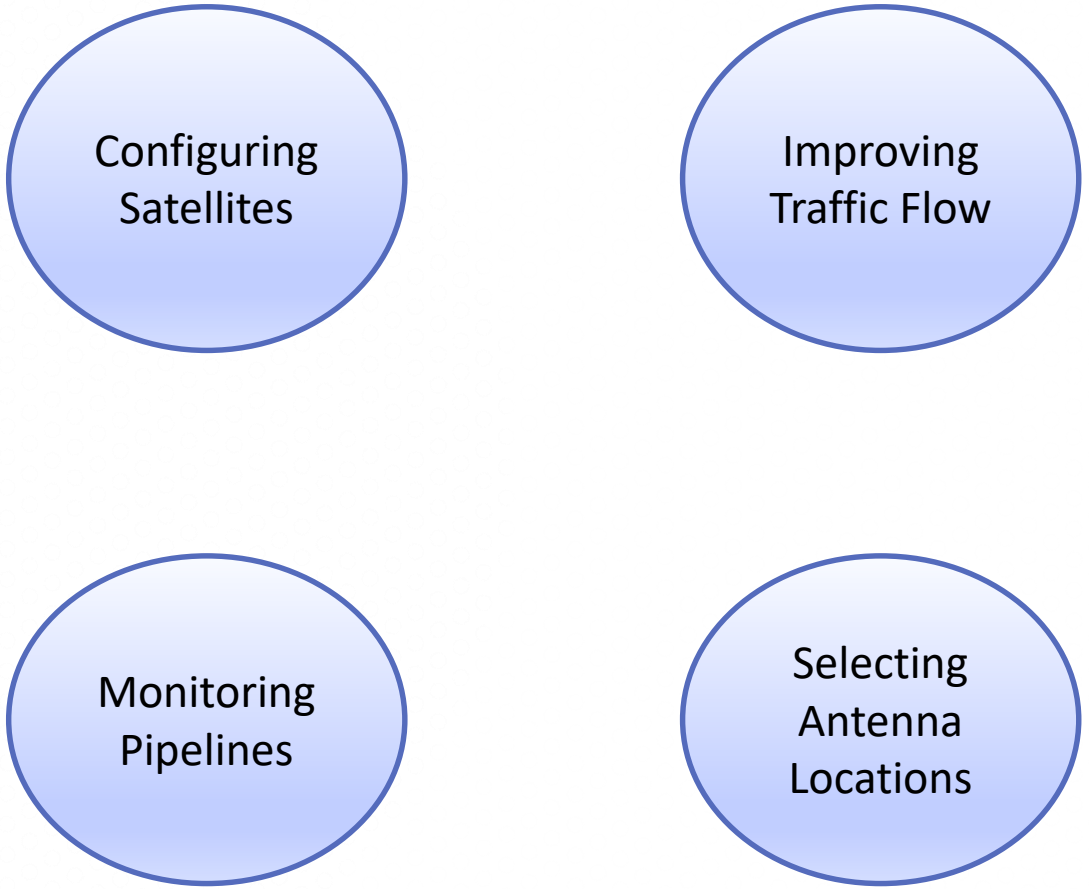
Library of Examples

- Currently in a github repository
  - https://github.com/dwavesystems/demos
- Improving and expanding continuously
- Issues and pull requests encouraged



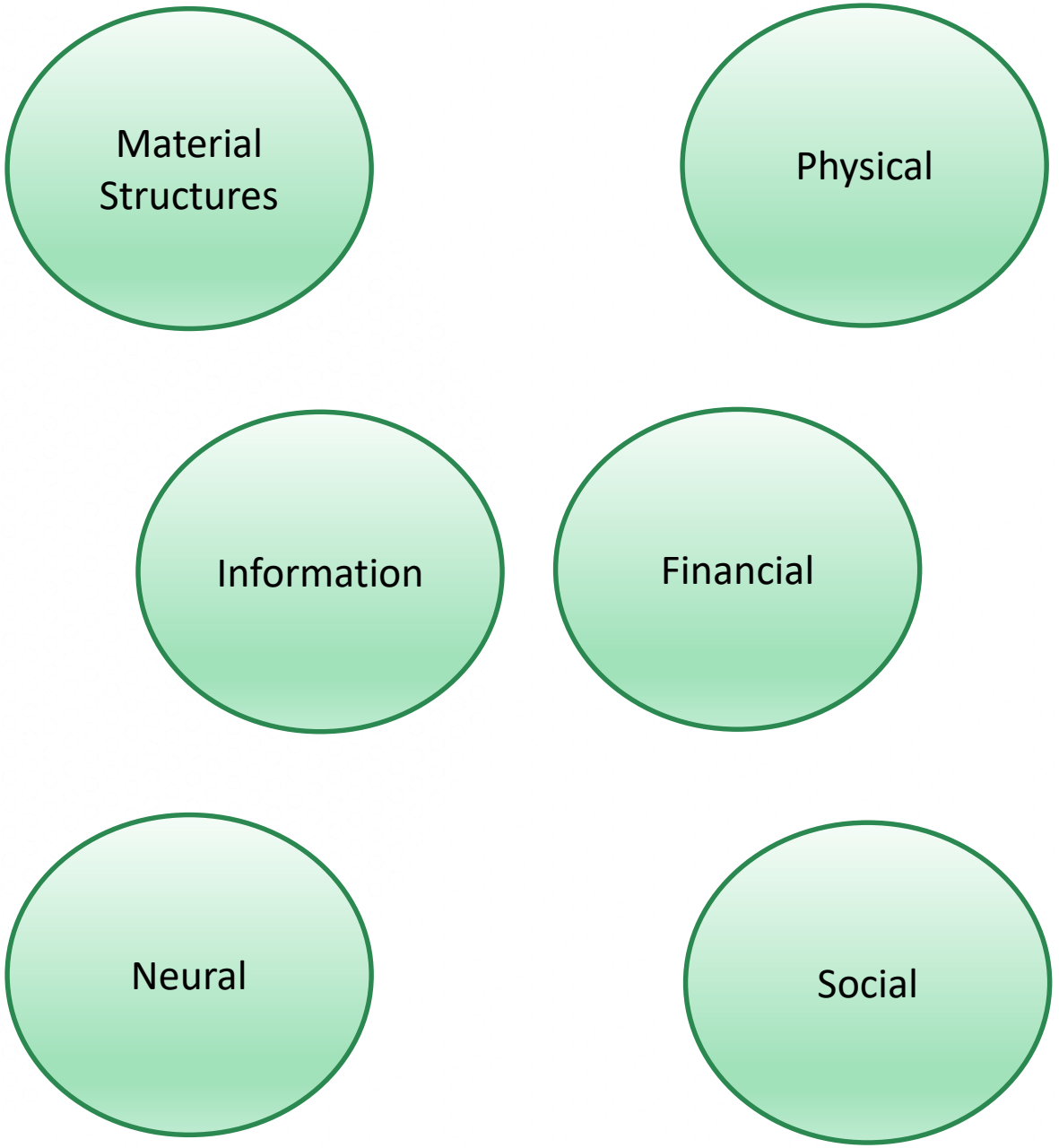
Physical Networks

- Binary characteristic of physical objects
  - Selection, failure
- Events have a correlated effect
- Vast space of potential solution configurations



Network Optimization

- Still related to the binary and correlated
  - Activation, purchase, default, friendship, bonding
- Broader domains of simulation, modeling, or prediction
- Branching into entertainment
  - Maze Solving, Sudoku puzzles



Materials Properties

Atomic magnetometer

Solid state materials simulation

Quantum molecular dynamics

Quantum chemistry computation

Optimization

Radiotherapy

Multi-period portfolios

Satellite placement

Traffic flow

Internet ad placement

150+ EARLY APPLICATIONS

Finding Higgs Boson

Image recognition

Tree cover classifier

DNA binding

Individual cancer drugs

Machine Learning

Formation of Terrorist Networks

Fault detection in circuits

Facial recognition

Cyber Security & Fault Detection

D-Wave

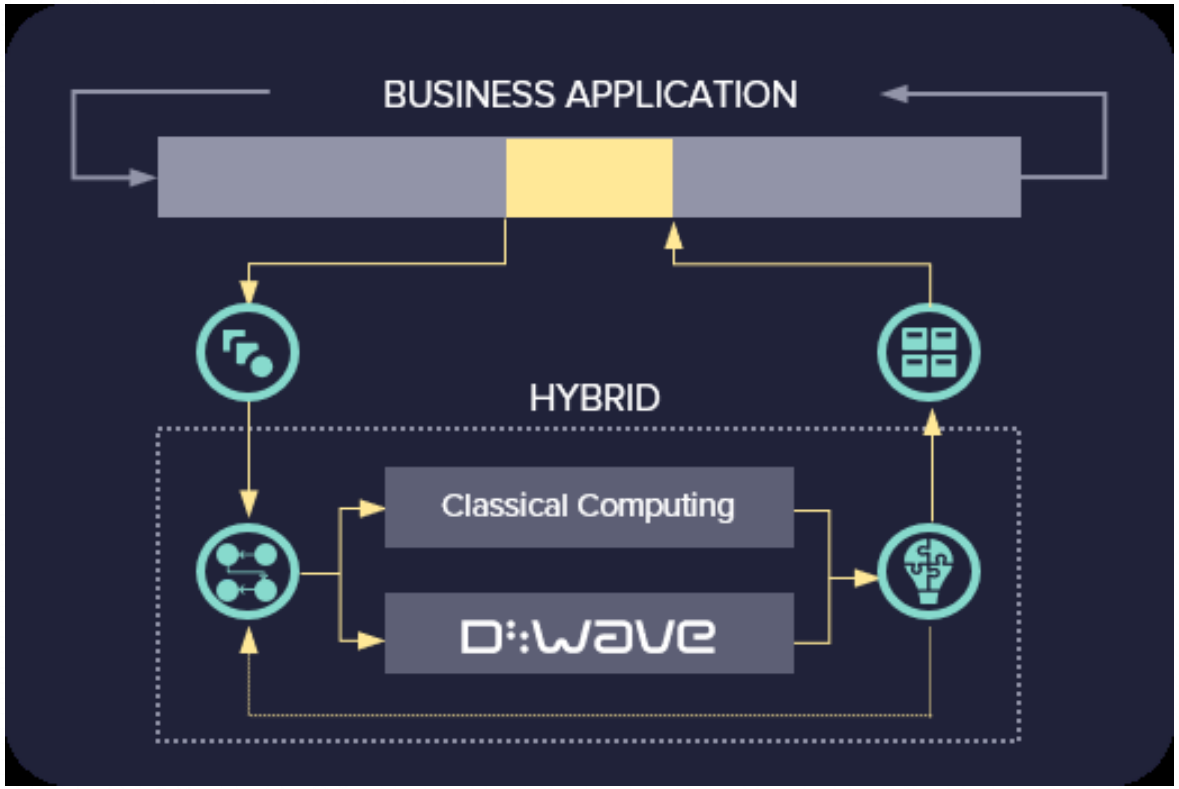
Live Quantum Applications

"We made history today, 1st commercial application powered by quantum computing is in action“
Martin Hofmann
Volkswagen Group CIO
Web Summit, Nov 2019

Hybrid Algorithm Development

dwave-hybrid

- Minimal, Python, solver/sampler-building framework, built atop Ocean tools
- Leverages **quantum** and **classical** resources
- Independent parts are executed **concurrently**
- Problems are **broken into pieces** that fit the compute resources
- Uses sample sets (probabilistic approach)



Motivation

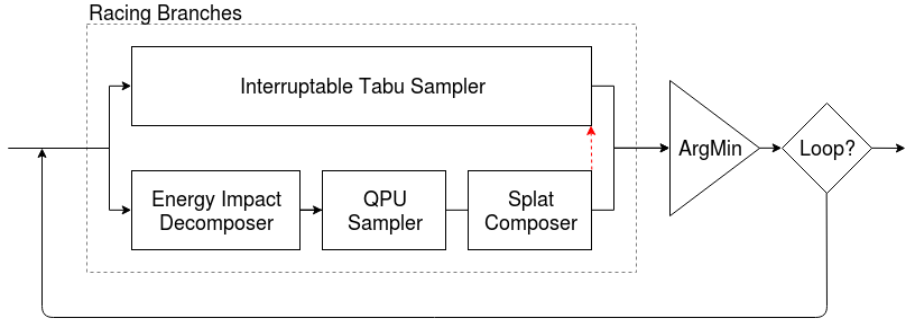
Algorithm 1 Partitioning algorithm implemented by `qbsolv`

Input: `QUBO` instance
`# best_energy` is the lowest value found to date
`# best_solution` is the solution bit vector corresponding to the lowest value so far
`# index` is the indices of the bits in the solution, sorted from
`#` most to least impact on value

`# Get initial estimate of minimum value and backbone`
`solution` ← random 0/1 vector
(`best_energy`, `best_solution`) ← *TabuSearch*(`QUBO`, `solutio`
`index` ← *OrderByImpact*(`QUBO`, `best_solution`)
`passCount` ← 0
`solution` ← `best_solution`
while `passCount` < `numRepeats` **do**
  `change` ← `false`
  for `i` = 0; `i` < `fraction` ∗ *Size*(`QUBO`); `i` += `subQUBOSize`
    `# select subQUBO with other variables clamped`
    `sub_index` ← `i` : `i`+`subQUBOSize`−1
    `subQUBO` ← *Clamp*(`QUBO`, `solution`, `index`[`sub_index`])
    (`sub_energy`, `sub_solution`) ← *DWaveSearch*(`subQUB`
    `# project onto full solution`
    if (`solution`[`sub_index`] ≠ `sub_solution`) **then**
      `solution`[`sub_index`] ← `sub_solution`
      `change` ← `true`
    end if
  end for
  if not `change` **then**
    Randomize(`solution`[0 : `i` − 1])
  end if
  (`energy`, `solution`) ← *TabuSearch*(`QUBO`, `solution`)
  if `energy` < `best_energy` **then**
    `best_energy` ← `energy`
    `best_solution` ← `solution`
    `passCount` ← 0
  else
    `passCount` + +
  end if
  `index` ← *OrderByImpact*(`QUBO`, `solution`)
end while
Output: `best_energy`, `best_solution`

```
double evaluate(int8_t *const solution, const uint qubo_size, const double **const qubo, double *const flip_cost) {
    double result = 0.0;

    for (uint ii = 0; ii < qubo_size; ii++) {
        double row_sum = 0.0;
        double col_sum = 0.0;

        // qubo an upper triangular matrix, so start right of the diagonal
        // for the rows, and stop at the diagonal for the columns
        for (uint jj = ii + 1; jj < qubo_size; jj++)
            if (solution[jj]) row_sum += qubo[ii][jj];

        for (uint jj = 0; jj < ii; jj++)
            if (solution[jj]) col_sum += qubo[jj][ii];

        // If the variable is currently 1, then by flipping it we
        if (CPSECONDS >
                ContinueWhile = false;
        }
    } // end of outer loop

    // all done print results if needed and free allocated arrays
    if (WriteMatrix_) print_solution_and_qubo(Qbest, qubo_size, qubo);

    if (Verbose_ == 0) {
        Qbest = &solution_list[Qindex[0]][0];
        best_energy = energy_list[Qindex[0]];
        // printf(" evaluated solution %8.2lf\n",
        //     sign * Simple_evaluate(Qbest, qubo_size, (const double **)qubo));
        print_output(qubo_size, Qbest, numPartCalls, best_energy * sign, CPSECONDS, param);
    }

    free(solution);
    free(tabu_solution);
    free(flip_cost);
    free(index);
    free(TabuK);
    free(Pcompress);

    return;
}
```

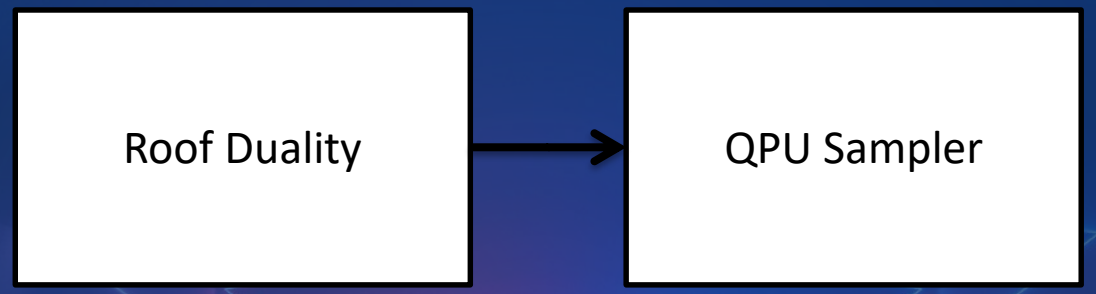
```
Loop(RacingBranches(
    InterruptableTabuSampler(),
    EnergyImpactDecomposer(size=50)
    | QPUSubproblemAutoEmbeddingSampler()
    | SplatComposer()
) | ArgMin())
```



Pre-processing

Example

- Use roof duality to determine and fix the assignments of some variables in polynomial time




```
RoofDualityDecomposer()
| QPUSubproblemAutoEmbeddingSampler()
| SplatComposer()
```

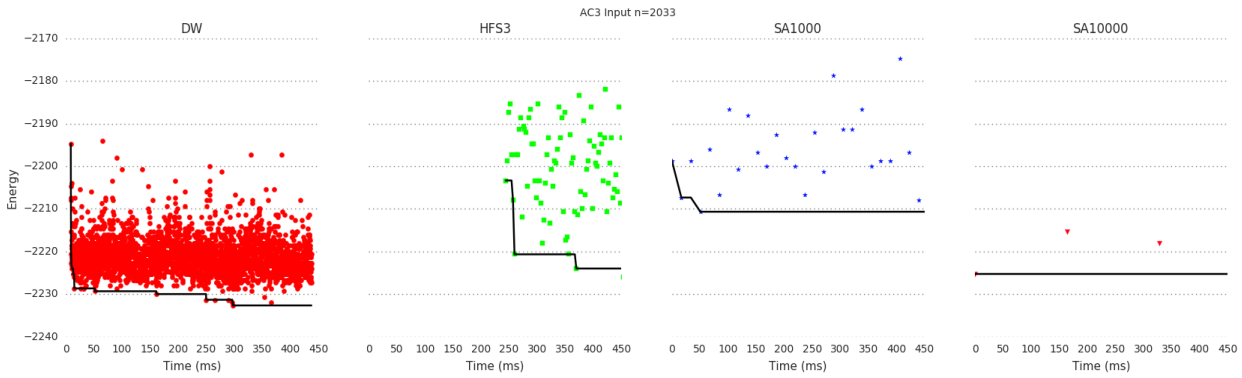
Boros, E., P.L. Hammer, G. Tavares. Preprocessing of Unconstrained Quadratic Binary Optimization. Rutcor Research Report 10-2006, April, 2006

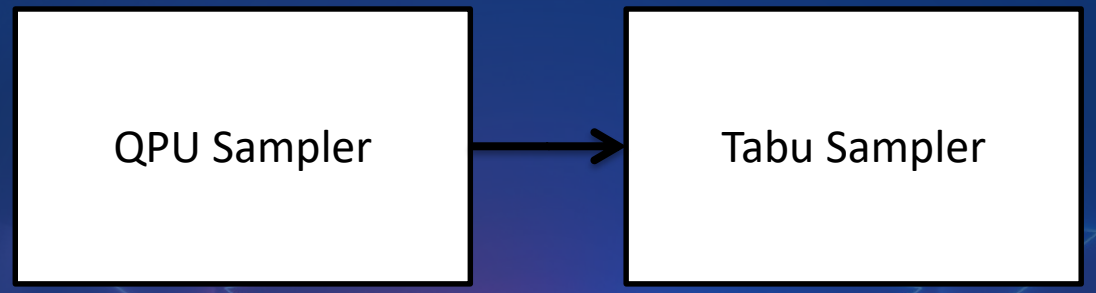
Boros, E. P.L. Hammer. Pseudo-Boolean optimization. Discrete Applied Mathematics 123, 2002, pp. 155-225

Post-processing

Example

- Use the QPU to seed another classical algorithm



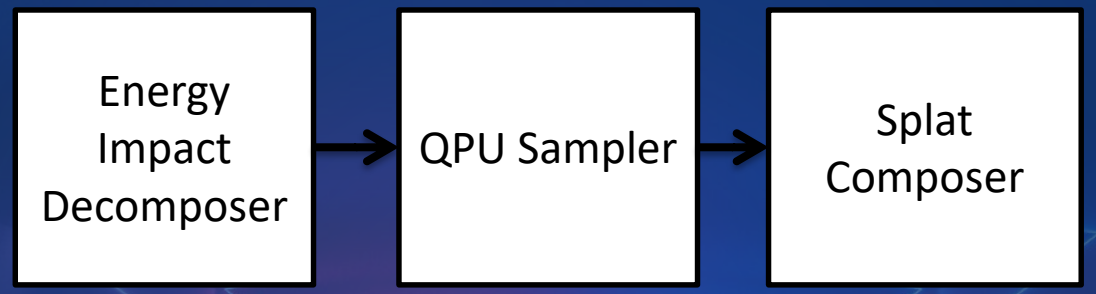


QPUSubproblemAutoEmbeddingSampler()
| TabuSubproblemSampler()

Decomposition

Example

- Find a sub problem with a high energy impact and solve that on the QPU

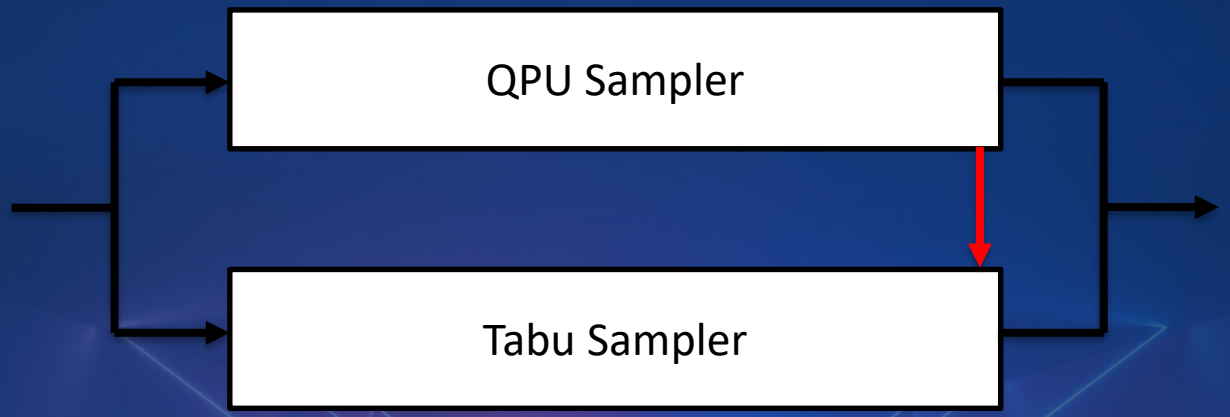



```
EnergyImpactDecomposer(size=50)
| QPUSubproblemAutoEmbeddingSampler()
| SplatComposer()
```

Racing

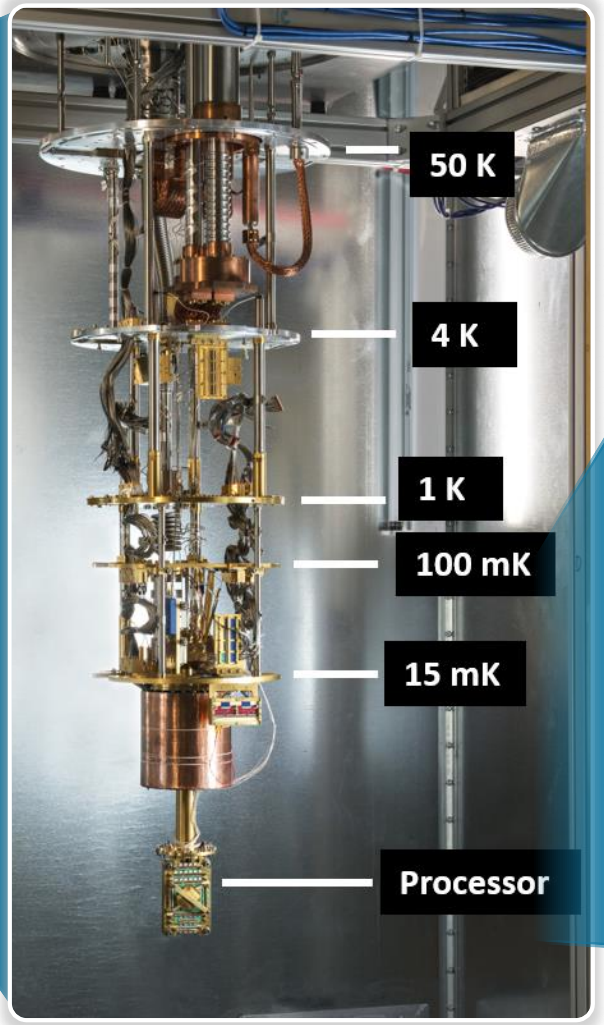
Example

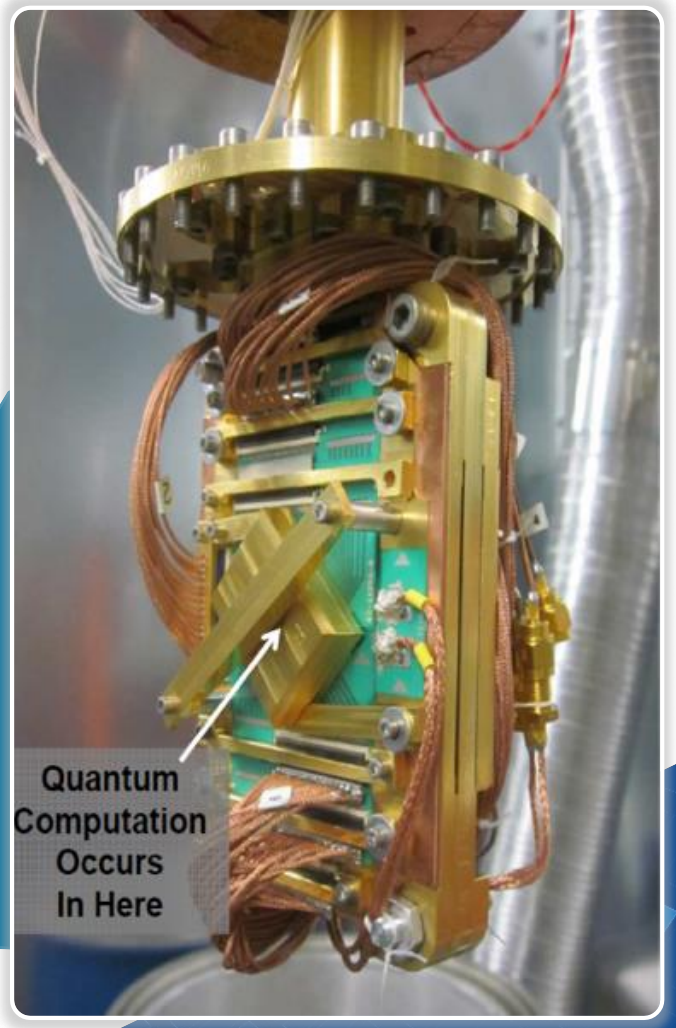
- Run tabu while waiting on the result from the QPU



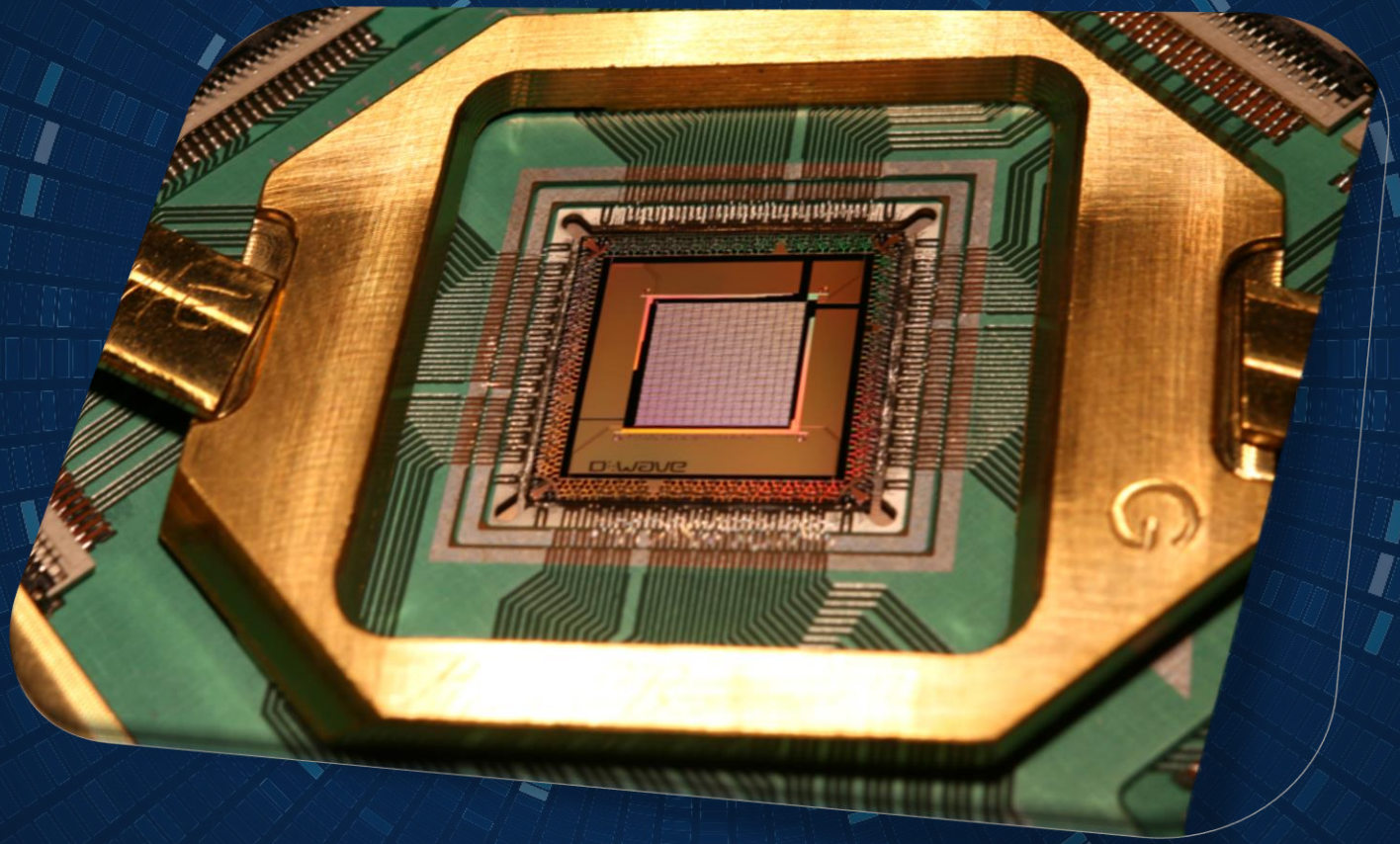

```
RacingBranches(
    InterruptableTabuSampler(),
    EnergyImpactDecomposer(size=50)
    | QPUSubproblemAutoEmbeddingSampler()
    | SplatComposer()
) | ArgMin())
```

What Is A Quantum Computer?





Quantum Processing Unit (QPU)



Leap

https://cloud.dwavesys.com/leap/

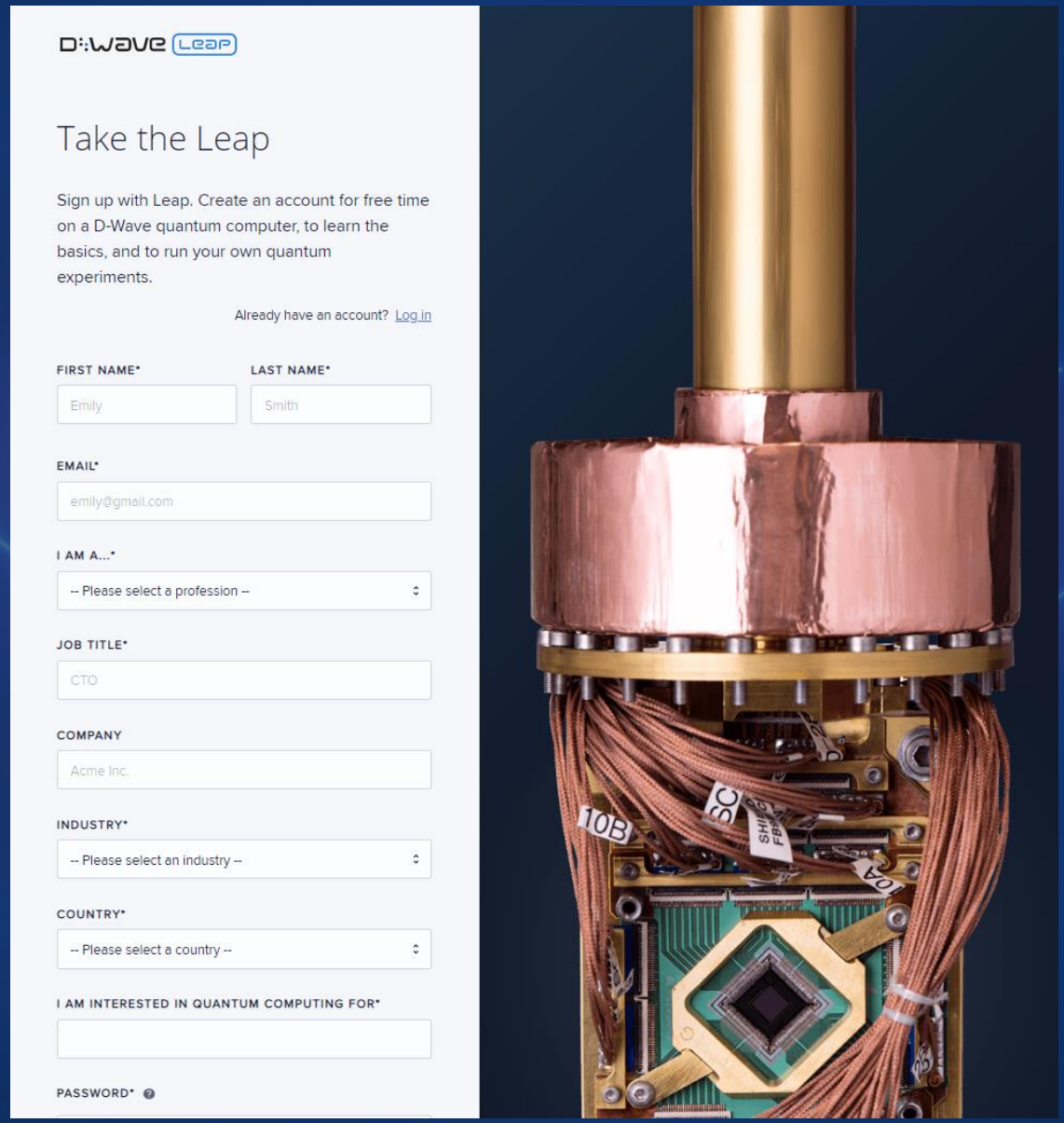
Find More And Send Feedback!

Leap

- Access
  https://cloud.dwavesys.com/leap
- Community & Help
  https://support.dwavesys.com/

Ocean

- Landing page
  https://ocean.dwavesys.com/
- Github
  https://github.com/dwavesystems
- Read-the-docs
  https://docs.ocean.dwavesys.com



goto; copenhagen

GOTO Copenhagen 2019

Conference Nov. 18 - 20

Please

Remember to rate this session

Thank you!

Follow us @gotocph