

# Practical Quantum Computing

An Introduction

Alexander Condello

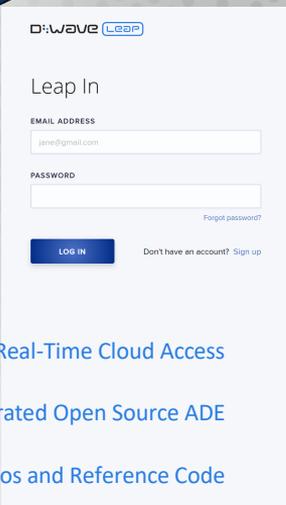
D-Wave



# D-Wave Leap™

The *Only* Real-Time  
Quantum Application Environment

Enabling a New Developer Community



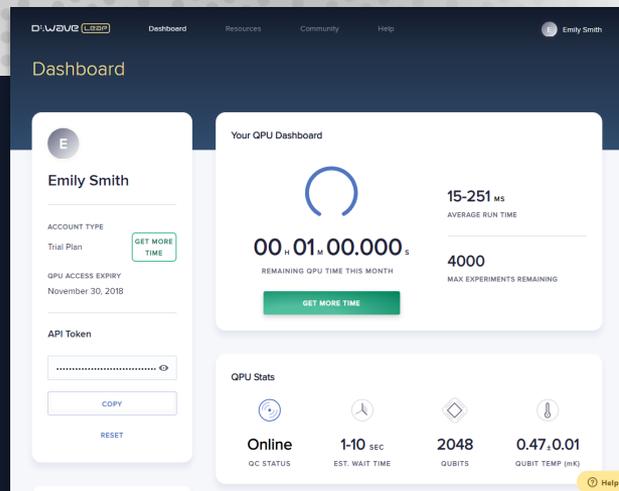
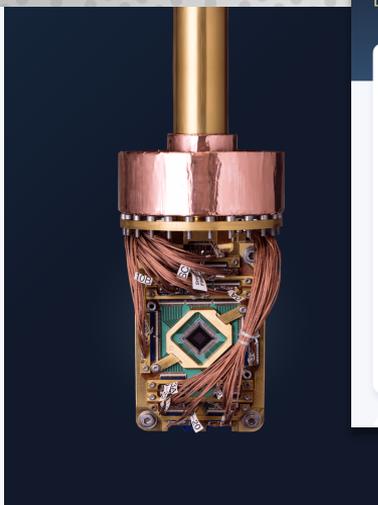
Free Real-Time Cloud Access

Integrated Open Source ADE

Demos and Reference Code

Community Support

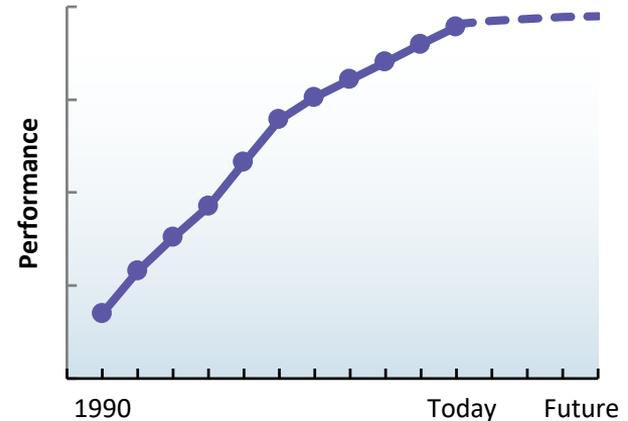
Online Training



# Why Quantum Computing Gets So Much Attention

- End of Moore's Law
- Classical systems' power consumption is reaching its limits
- Potential to address NP-hard problems
- Huge potential speedup over classical approaches on some problems

Classical computers are reaching limits...



<sup>1</sup> E.g., Speed, compactness, power consumption  
SOURCE: Intel presentation; IEEE Transactions on Semiconductor Manufacturing May 2011, McKinsey

# Practical

---

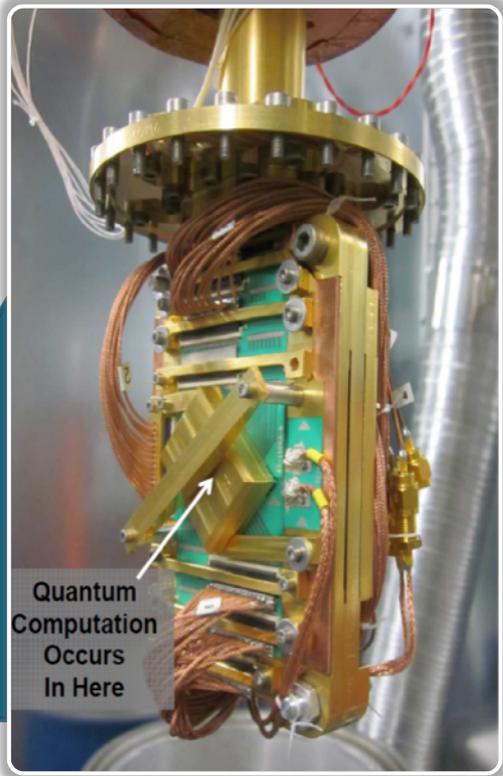
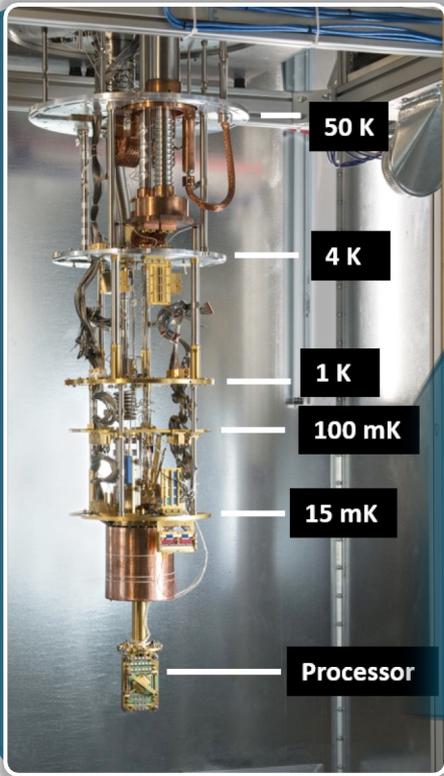
## Adiabatic Quantum Computing (AQC) [Farhi et al. '01]

- System is evolved from the lowest-energy state of an easy problem to the lowest-energy state of a target problem
- If evolution is gradual enough, system remains in the lowest energy state
- Initially, the state is a superposition of all classical states

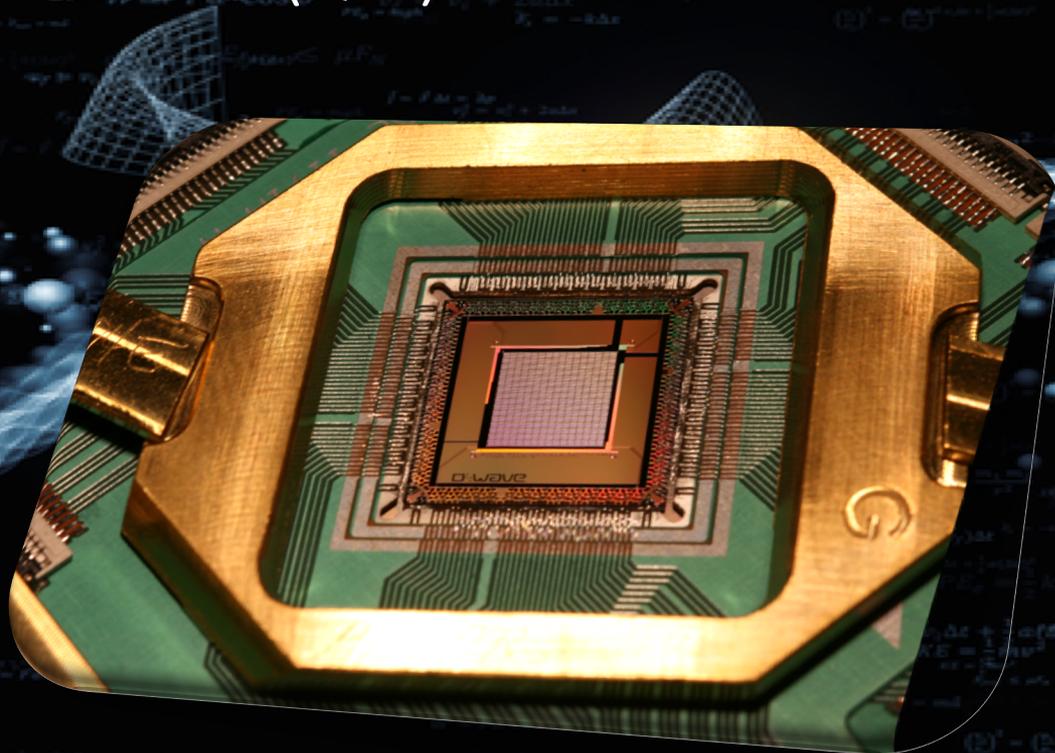
## Why AQC?

- Computationally equivalent to gate-model [Aharonov et al. '08]

# How To Build a Quantum Computer



# Quantum Processor Unit (QPU)



# Quantum Machine Language Programming

QUBIT	$q_i$	<b>Quantum bit</b> which participates in annealing cycle and settles into one of two possible final states: $\{0,1\}$
COUPLER	$q_i q_j$	Physical device that allows one <b>qubit to influence another qubit</b>
WEIGHT	$a_i$	Real-valued <b>constant associated with each qubit</b> , which influences the qubit's tendency to collapse into its two possible final states; controlled by the programmer
STRENGTH	$b_{ij}$	Real-valued <b>constant associated with each coupler</b> , which controls the influence exerted by one qubit on another; controlled by the programmer
OBJECTIVE	$Obj$	Real-valued <b>function which is minimized</b> during the annealing cycle

$$Obj(a_i, b_{ij}; q_i) = \sum_i a_i q_i + \sum_{ij} b_{ij} q_i q_j$$

# Binary Quadratic Model

---

$$E(v) = \sum_{i,j} b_{i,j} v_i v_j + \sum_i a_i v_i + c$$

$$a_i, b_{i,j}, c \in \mathbb{R}$$

$$v_i \in \{-1, +1\} \text{ or } v_i \in \{0, 1\}$$

# Synonyms

---

- Binary Quadratic Model
- Ising Model
- Quadratic Unconstrained Binary Optimization Problem
- Probabilistic Graphical Model
- Restricted Boltzman Machine

# Binary Quadratic Model

---

$$E(v) = \sum_{i,j} b_{i,j} v_i v_j + \sum_i a_i v_i + c$$

$$a_i, b_{i,j}, c \in \mathbb{R}$$

$$v_i \in \{-1, +1\} \text{ or } v_i \in \{0, 1\}$$

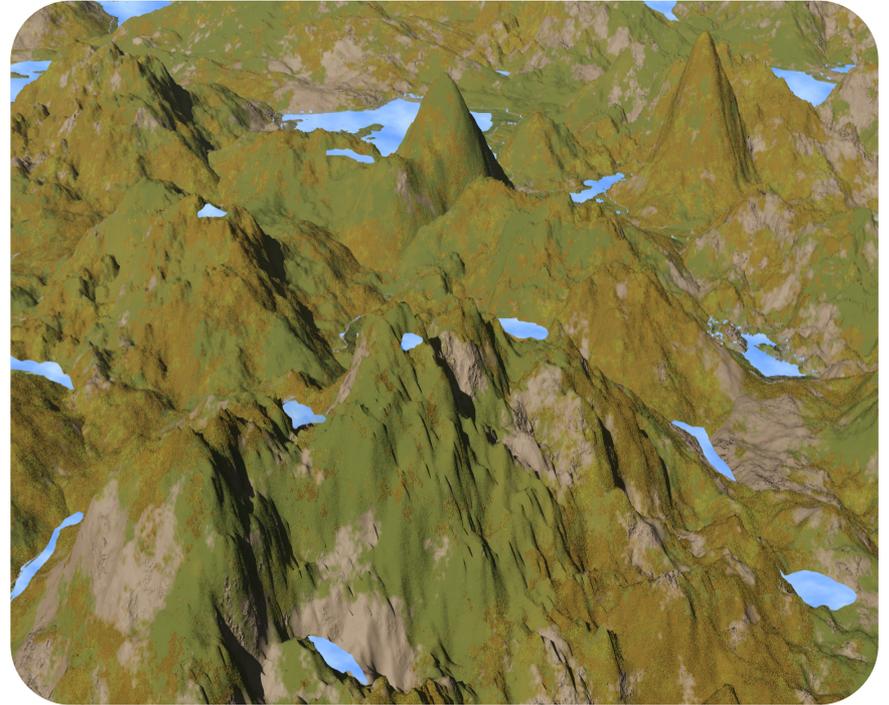
# Landscape metaphor

---

Space of solutions defines an energy landscape and the best solution is the lowest valley

Classical algorithms can only walk over this landscape

Quantum annealing uses quantum effects



# Sampling

---

## QPU is probabilistic

- Finding a ground state
- Sampling ground states
- Characterizing the landscape
  - Machine learning
  - Material simulation

# Is it possible to express problems as BQMs?

frontiers in  
**PHYSICS**

**REVIEW ARTICLE**  
Published: 12 February 2014  
doi: 10.3389/fphy.2014.00005



## Ising formulations of many NP problems

**Andrew Lucas\***  
*Lyman Laboratory of Physics, Department of Physics, Harvard University, Cambridge, MA, USA*

**Edited by:**  
Jacob Biamonte, ISI Foundation, Italy

**Reviewed by:**  
Mauro Paccin, ISI Foundation, Italy  
Ryan Babbush, Harvard University, USA  
Dana A. O'Connell, MIT, USA

We provide Ising formulations for many NP-complete and NP-hard problems, including all of Karp's 21 NP-complete problems. This collects and extends mappings to the Ising model from partitioning, covering, and satisfiability. In each case, the required number of spins is at most cubic in the size of the problem. This work may be useful in designing adiabatic quantum optimization algorithms.

**Keywords:** spin glasses, complexity theory, adiabatic quantum computation, NP, algorithms

- Partitioning Problems
  - Number Partitioning
  - Graph Partitioning
  - Cliques
  - Reducing  $N$  TO  $\log N$  Spins in Some Constraints
  - Binary Integer Linear Programming
- Covering and Packing Problems
  - Exact Cover
  - Set Packing
  - Vertex Cover
  - Satisfiability
  - Minimal Maximal Matching
- Problems with Inequalities
  - Set Cover
  - Knapsack with Integer Weights

- Coloring Problems
  - Graph Coloring
  - Clique Cover
  - Job Sequencing with Integer Lengths
- Hamiltonian Cycles
  - Hamiltonian Cycles and Paths
  - Traveling Salesman
- Tree Problems
  - Minimal Spanning Tree with a Maximal Degree Constraint
  - Steiner Trees
  - Directed Feedback Vertex Set
  - Undirected Feedback Vertex Set
  - Feedback Edge Set
  - Graph Isomorphisms

# Customer's Early Applications

## Optimization



Multi-period  
portfolios



Internet ad  
placement

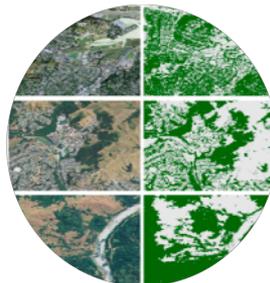


Satellite Placement

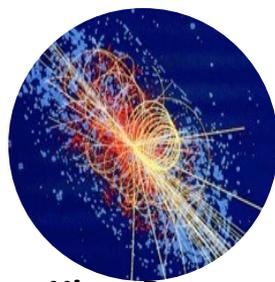
## Machine Learning



Image  
recognition

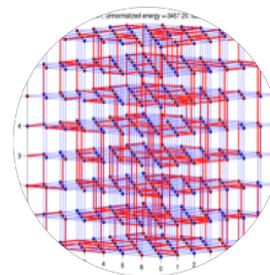


Tree cover  
classifier

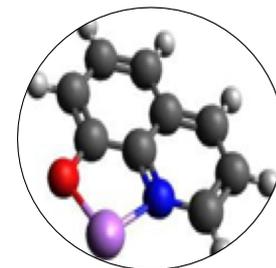


Higgs Boson  
Detection

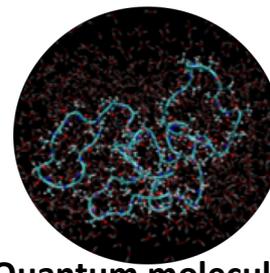
## Quantum Materials



Quantum  
solid state



Quantum  
Molecules



Quantum molecular  
dynamics



# Why is this interesting to the open source community

---

- New and exciting field for computing
- There are problems best solved on classical resources
  - Need developers on the classical side
- You can try these things for yourself
  - We in the open source space thrive on growth and debate

D-Wave Leap Sign Up | x  
https://cloud.dwavesys.com/leap/signup/

**D:WAVE** LEAP

## Take the Leap

Sign up with Leap. Create an account for free time on a D-Wave quantum computer, to learn the basics, and to run your own quantum experiments.

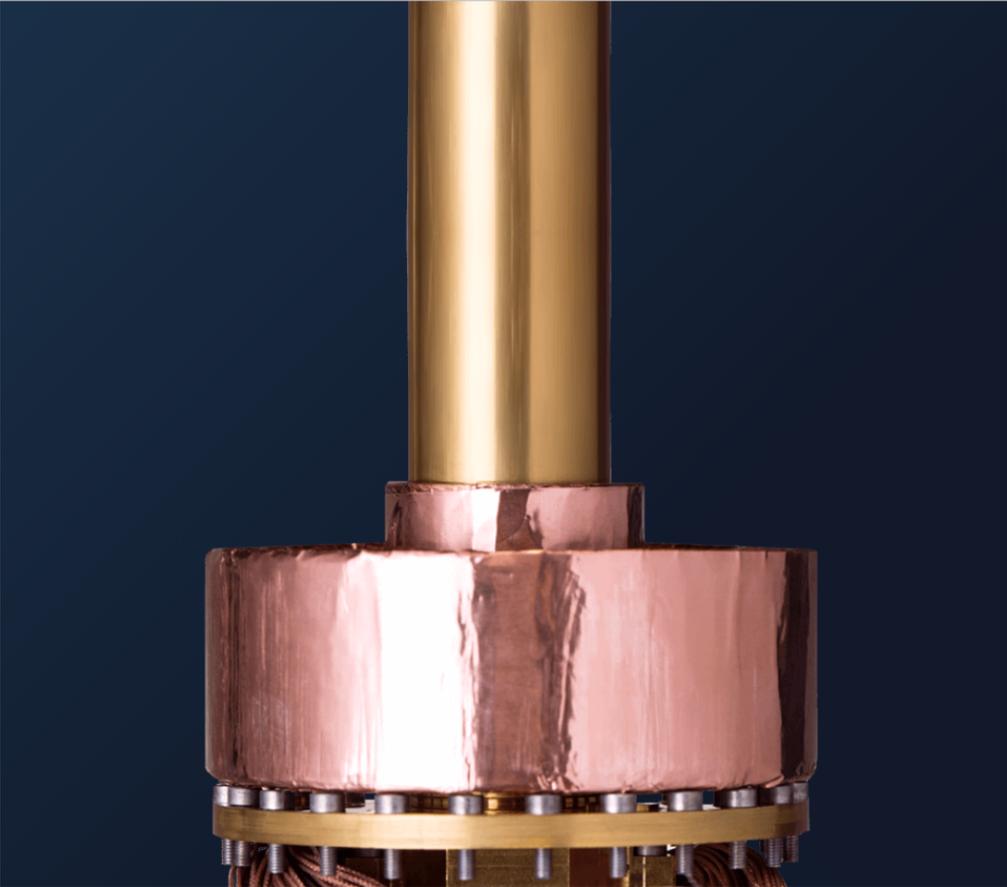
**FIRST NAME\***  **LAST NAME\***

**EMAIL\***

**I AM A...\***

**JOB TITLE\***

**COMPANY**



Demo 2 > Try it on the D-Wave Leap

https://cloud.dwavesys.com/leap/demos/socialnetwork/try-it

D-WAVE LEAP

LEAVE THE DEMO

SOCIAL NETWORK ANALYSIS

SYRIA 2013

# Try it on the D-Wave quantum computer

**SYRIAN DATASET**

A study of the violent extremist network in Syria found that the network was balanced in 2012. However, in 2013 an increase in active groups in the Syrian theatre changed the existing landscape significantly.

Here is the network for the Syrian theatre in 2013. Click to run on the quantum computer to compute its structural balance.

**SOLVE THIS PROBLEM USING THE D-WAVE QUANTUM COMPUTER** **RUN**

PREVIOUS STEP

NEXT: Performance **CONTINUE**

# Dashboard

## Getting Started

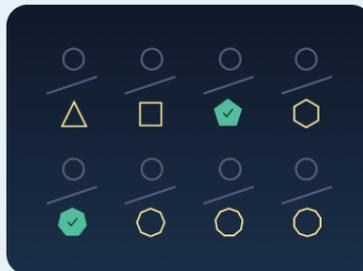
DISMISS ×



LEARN

### Learn about Leap and QC

Watch videos, meet D-Wave customers, and access research publications.



EXPERIMENT ON A QC

### Run a Demo

Learn how D-Wave quantum computers work by running demos.



GET SET UP

### Install our SDK

Get started developing software that will run on the D-Wave system.

? Help

Topics – D-Wave System | X

https://support.dwavesys.com/hc/en-us/community/topics?flash\_digest=ef4de9b91586e20ad123ae4027de70b5be2ec45b

D:WAVE Leap Dashboard Resources Community Help Murray

Leap Help > Community

Search

Topics Posts NEW POST

### Welcome to the Community!

General information about the Leap community.

3 posts · 2 followers

### General Discussion

Topics which do not fall under any of the existing categories.

9 posts · 4 followers

### Quantum Computing Concepts

Discussion related to basic quantum computing concepts.

2 posts · 1 follower

### Coding Tips and Tricks

Share your ideas on how write effective code for the QPU and troubleshoot issues.

0 posts · 1 follower

### Documentation and Learning Resources

Questions and discussion about system documentation, Ocean documentation and Jupyter Notebooks

### Feature Requests

Have a new feature in mind? Share it here.

7 posts · 6 followers

Help

D-Wave Systems x

https://support.dwavesys.com/hc/en-us?flash\_digest=aaa773fb68da2e05292c488daba80a1612f94c0a

D:WAVE LEAP Dashboard Resources Community Help Murray

# How can we help?

Search SEARCH



GETTING STARTED FAQ KNOWLEDGE BASE COMMUNITY

Help

# D-Wave's Software Development Kit

Ocean

Alexander Condello

D-Wave



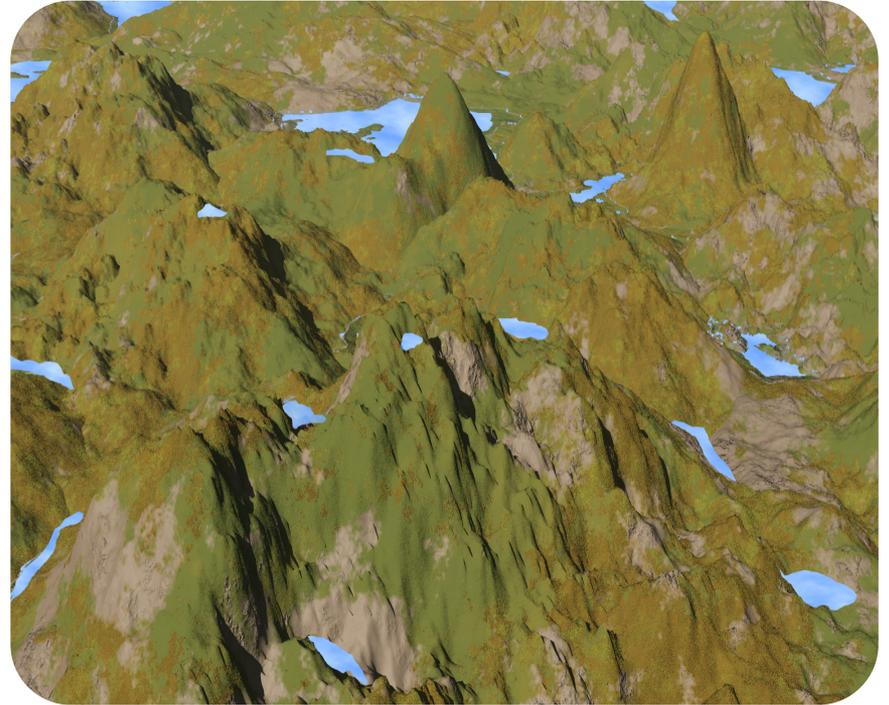
# Landscape metaphor

---

Space of solutions defines an energy landscape and the best solution is the lowest valley

Classical algorithms can only walk over this landscape

Quantum annealing uses quantum effects



# Binary Quadratic Model

---

$$E(v) = \sum_{i,j} v_i v_j a_{i,j} + \sum_i v_i b_i + c$$

$$a_{i,j}, b_i, c \in \mathbb{R}$$

$$v_i \in \{-1, +1\} \text{ or } v_i \in \{0, 1\}$$

# Quantum Machine Instruction

---

$$E(s) = \sum_{i,j} s_i s_j J_{i,j} + \sum_i s_i h_i$$

$$J_{i,j} \in [-2, 1], h_i \in [-2, 2]$$

$$s_i \in \{-1, +1\}$$

Has a particular graph, defined by the hardware

# How do I program with BQMs?

---

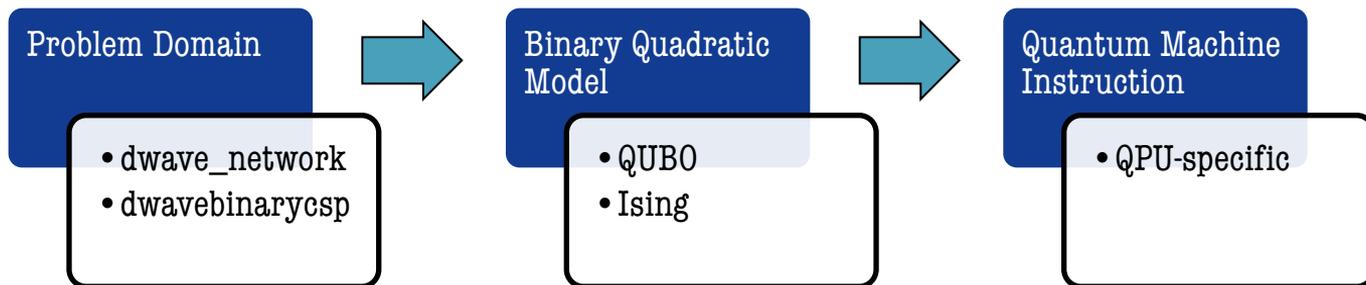
## D-Wave's Ocean Software

Ocean software is a suite of tools D-Wave Systems provides on the D-Wave GitHub repository for solving hard problems with quantum computers.

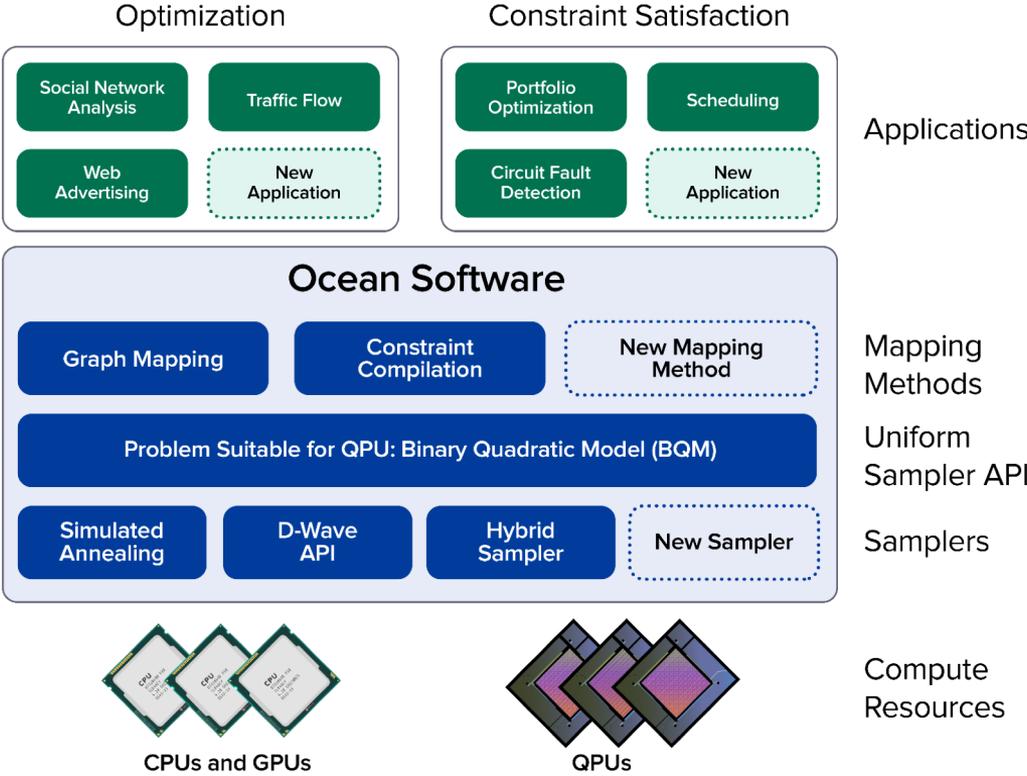
- Tools for solving BQM problems.
- Python front-end, some C++ components
- Majority is open-source code available on GitHub.
- Extensions and features from community welcome!

# How does it all fit together?

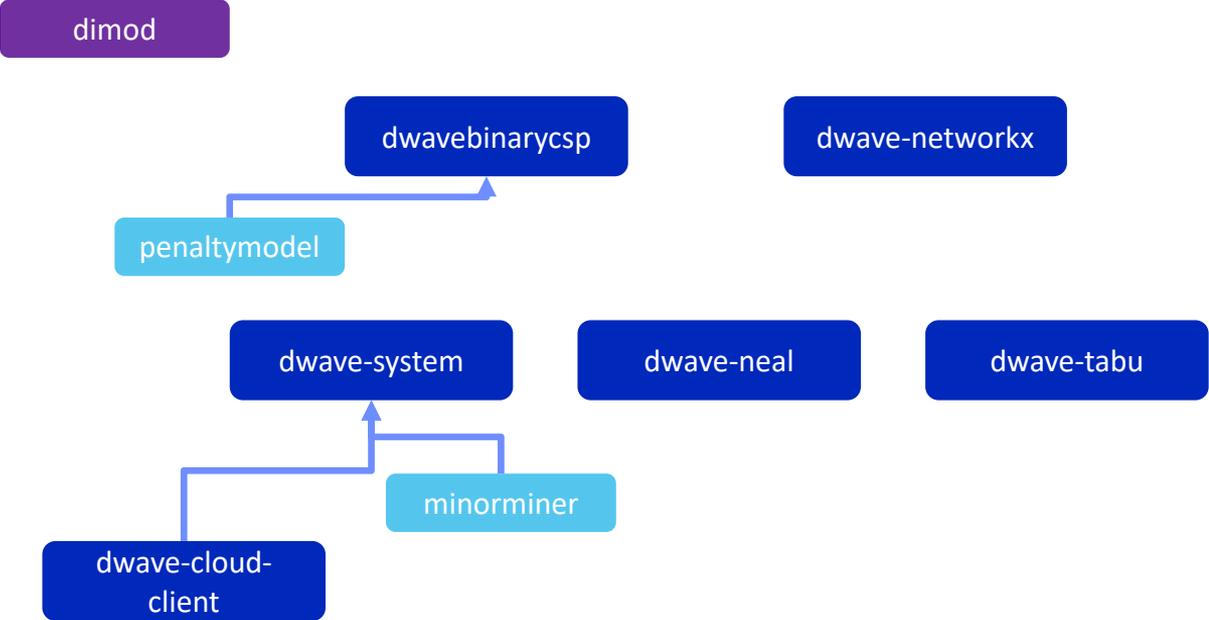
## Programming Model



# Ocean Software Stack



# Ocean Packages



Mapping Methods

Uniform Sampler API

Compute Resources

# Ocean Packages

---

dimod

dimod

- Provides objects used throughout the Ocean stack
  - BQM
  - Sampler ABC

# Ocean Packages

---

dimod

Mapping Methods

Uniform Sampler API

Compute Resources

# Ocean Packages

---

dimod

dwave-cloud-client

- Handles communication with the QPU
- Stores token locally for reuse

Mapping Methods

Uniform Sampler API

dwave-cloud-  
client

Compute Resources

# Ocean Packages

---

dimod

dwave-system

- Access QPU as a Sampler

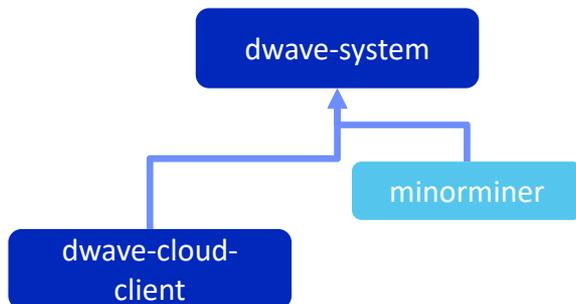
minorminer

- Tools for embedding problems onto the QPU

Mapping Methods

Uniform Sampler API

Compute Resources

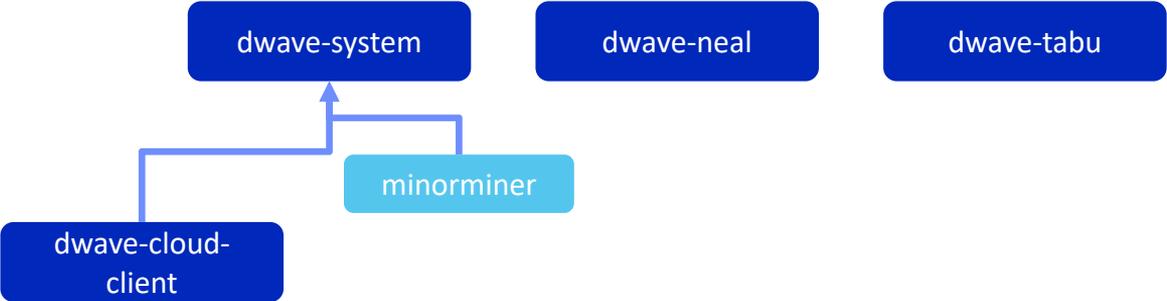


# Ocean Packages

dimod

- dwave-neal
  - Simulated annealing
- dwave-tabu
  - Tabu search

Mapping Methods



Uniform Sampler API

Compute Resources

# Ocean Packages

dimod

dwavebinaryscsp

- Construct a BQM from a constraint satisfaction problem

dwavebinarycsp

penaltymodel

dwave-system

dwave-neal

dwave-tabu

minorminer

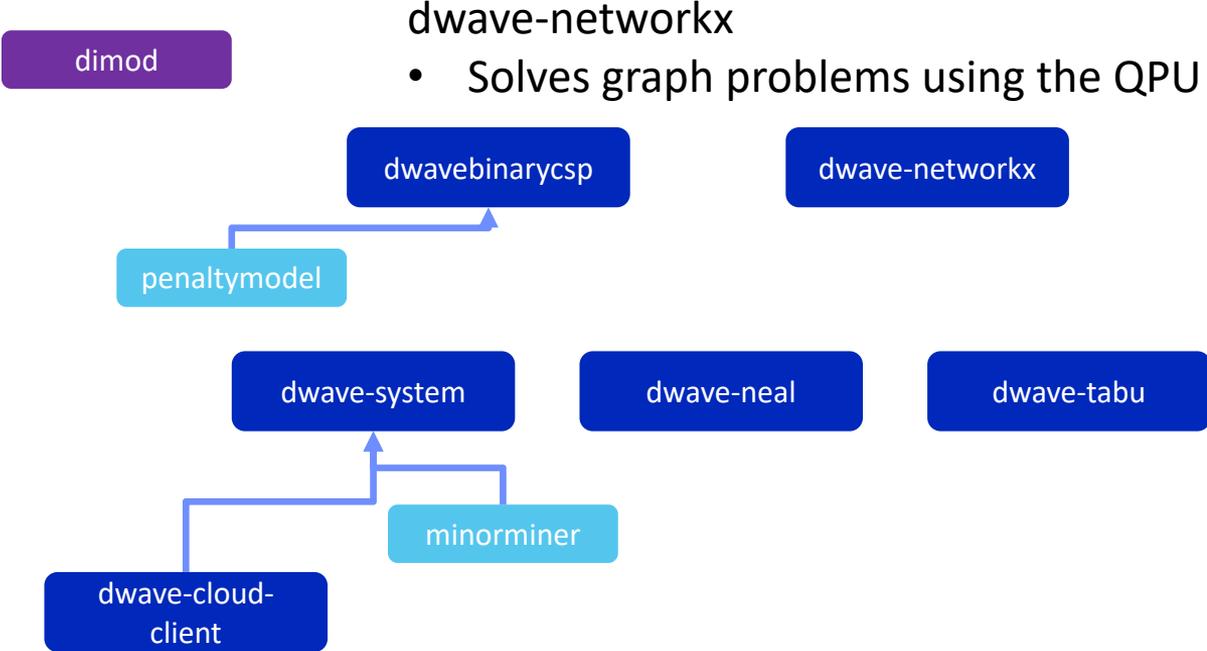
dwave-cloud-client

Mapping Methods

Uniform Sampler API

Compute Resources

# Ocean Packages



dwave-networkx

- Solves graph problems using the QPU

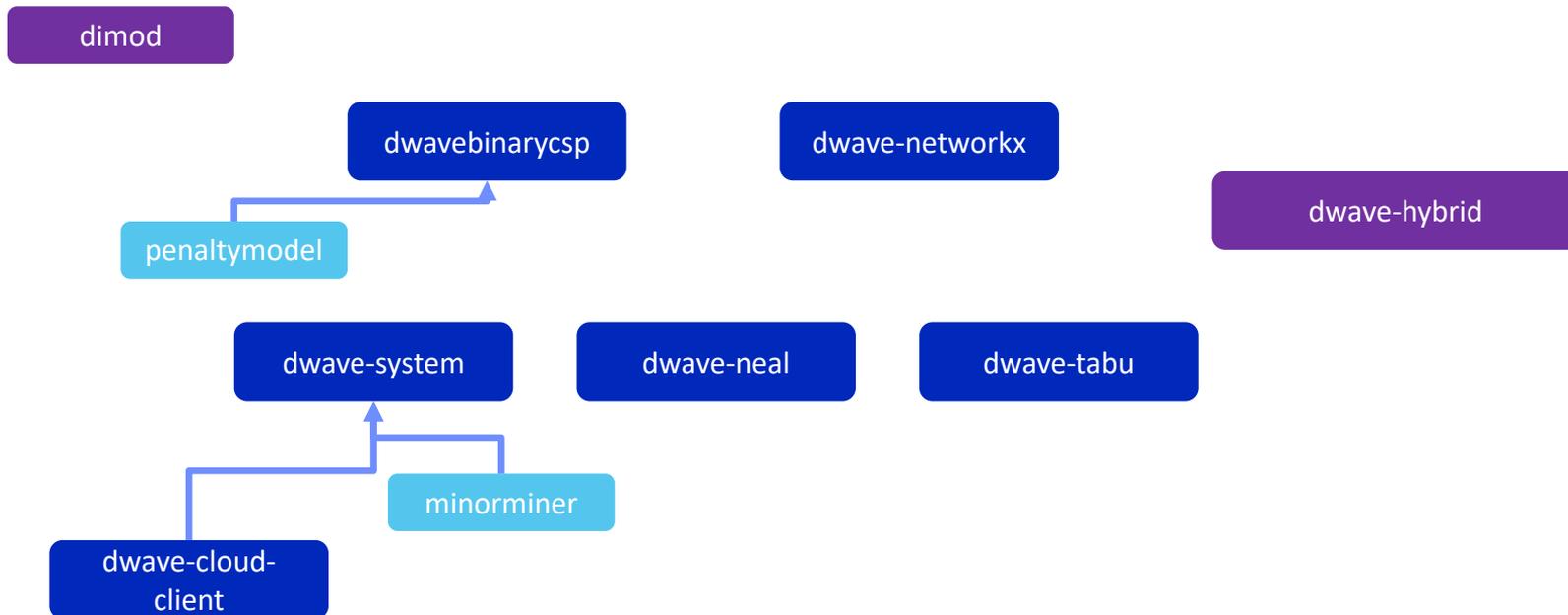
Mapping Methods

Uniform Sampler API

Compute Resources

# Ocean Packages

---



# QPU Programming Method

---

1. Translate your problem to binary optimization
2. Define your BQM function
3. Put the coefficients in matrix form
4. Run the matrix through a sampler
5. Post-process to interpret the results

# Example

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

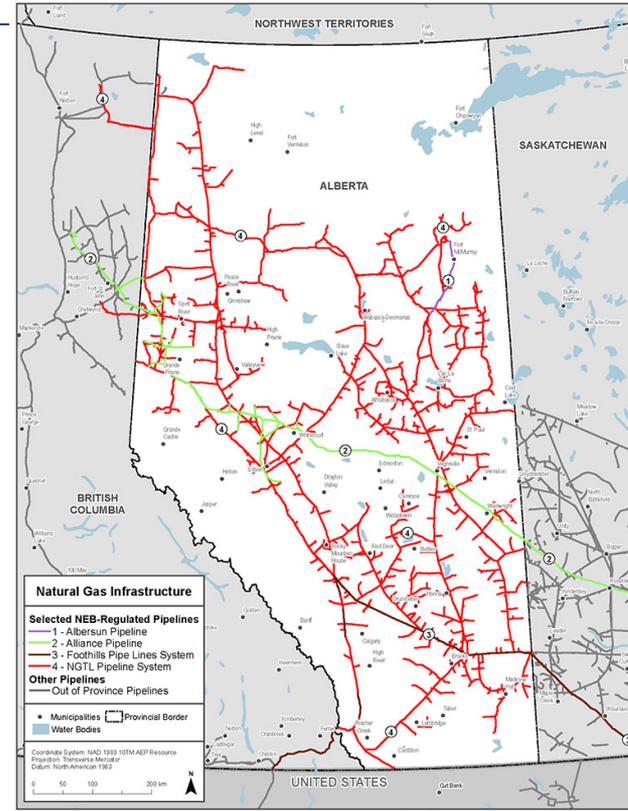
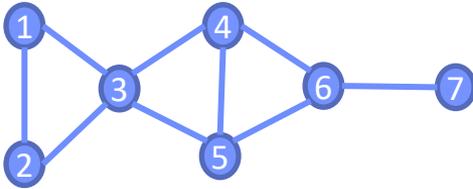
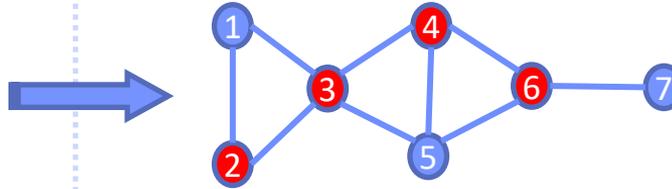


Image from National Energy Board of Canada,  
<https://www.neb-one.gc.ca/nrg/ntgrtd/mrkt/nrgsstmprfls/ab-eng.html>

# Example



Goal:  
Junctions for monitoring pipes



Goal:  
Nodes that *cover* every edge

In graph theory, this is called a  
**vertex cover**.

# Without Ocean algorithm tools....

---

1. Write our problem in binary optimization form

$$\begin{aligned} \min \quad & \sum_i x_i \\ \text{s. t.} \quad & \sum_{(u,v) \in E} (x_u \cdot x_v - x_u - x_v) > 0 \end{aligned}$$

2. Translate to a QUBO

$$\min \left( \sum_i x_i \right) + \gamma \left( \sum_{(u,v) \in E} (x_u \cdot x_v - x_u - x_v) \right)$$

3. Simplify down to a QUBO matrix

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

# Without Ocean algorithm tools....

1. Write our problem in binary optimization form

$$\min \sum_i x_i$$

subject to  $(x_u \cdot x_v - x_u - x_v) \geq 0$

2. Translate to a QUBO

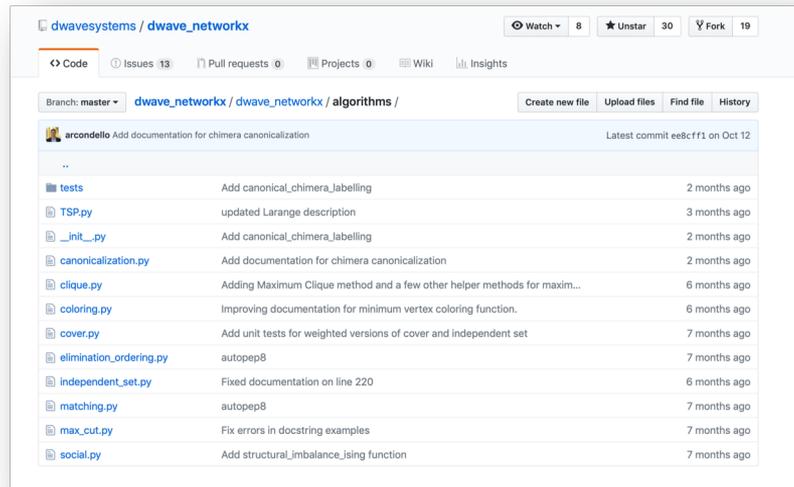
$$\min \left( \sum_i x_i + \gamma (x_u \cdot x_v - x_u - x_v) \right)$$

3. Simplify down to a QUBO matrix

			1	0	0	
-1	1	0	0	0	0	
	1	-3	1	1	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

# With Ocean algorithm tools!

- Identify the type of problem (e.g. min vertex cover)
- Search the Ocean Tool Suite for a tool



- Write a Python program *in our problem domain* and call the tool

# Python Program using QPU

---

```
import networkx as nx
import dwave_networkx as dnx
from dwave.system import DWaveSampler, EmbeddingComposite

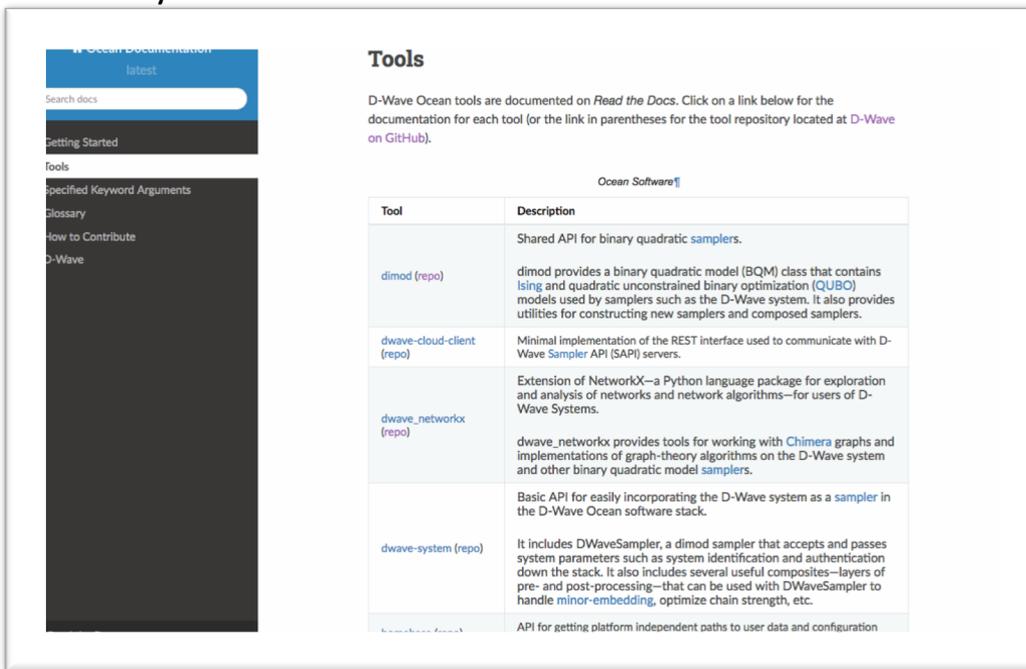
sampler = EmbeddingComposite(DWaveSampler())

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)
```

# Where to find more information

- Read-the-docs
  - <https://docs.ocean.dwavesys.com>



The screenshot shows the D-Wave Ocean Documentation website. On the left is a dark navigation menu with links for 'Getting Started', 'Tools', 'Specified Keyword Arguments', 'Glossary', 'How to Contribute', and 'D-Wave'. The main content area is titled 'Tools' and contains a paragraph of introductory text followed by a table of tools. The table has two columns: 'Tool' and 'Description'. The tools listed are dimod, dwave-cloud-client, dwave\_networkx, and dwave-system.

## Tools

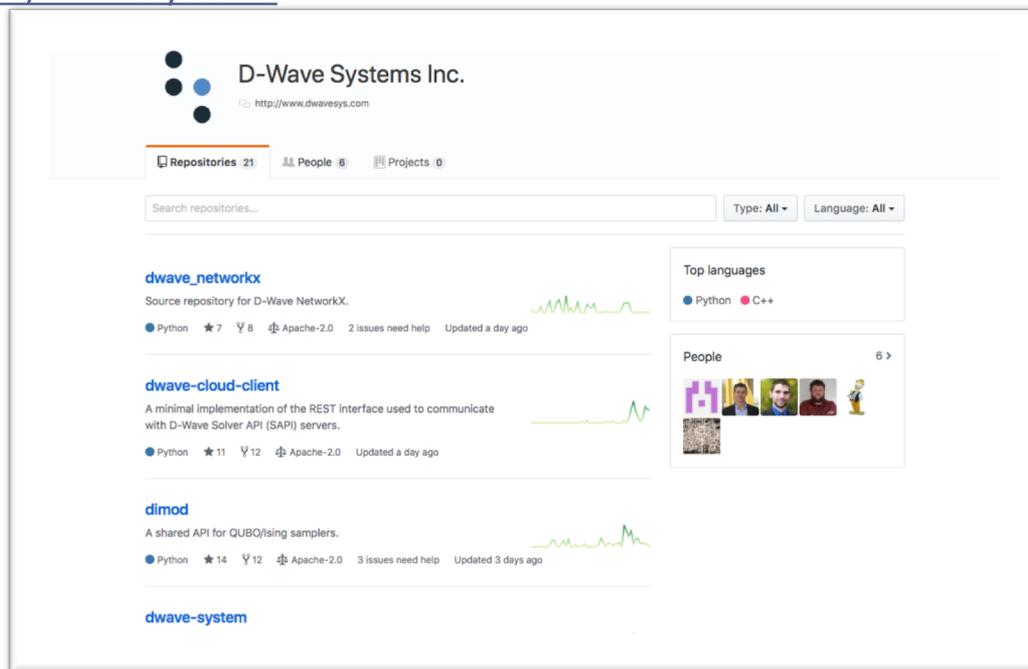
D-Wave Ocean tools are documented on *Read the Docs*. Click on a link below for the documentation for each tool (or the link in parentheses for the tool repository located at *D-Wave on GitHub*).

Ocean Software™

Tool	Description
<a href="#">dimod (repo)</a>	Shared API for binary quadratic <a href="#">samplers</a> .  dimod provides a binary quadratic model (BQM) class that contains <a href="#">Ising</a> and quadratic unconstrained binary optimization (QUBO) models used by samplers such as the D-Wave system. It also provides utilities for constructing new samplers and composed samplers.
<a href="#">dwave-cloud-client (repo)</a>	Minimal implementation of the REST interface used to communicate with D-Wave <a href="#">Sampler API (SAPI)</a> servers.
<a href="#">dwave_networkx (repo)</a>	Extension of NetworkX—a Python language package for exploration and analysis of networks and network algorithms—for users of D-Wave Systems.  dwave_networkx provides tools for working with <a href="#">Chimera</a> graphs and implementations of graph-theory algorithms on the D-Wave system and other binary quadratic model <a href="#">samplers</a> .
<a href="#">dwave-system (repo)</a>	Basic API for easily incorporating the D-Wave system as a <a href="#">sampler</a> in the D-Wave Ocean software stack.  It includes DWaveSampler, a dimod sampler that accepts and passes system parameters such as system identification and authentication down the stack. It also includes several useful composites—layers of pre- and post-processing—that can be used with DWaveSampler to handle <a href="#">minor-embedding</a> , optimize chain strength, etc.
<a href="#">Environment</a>	API for getting platform independent paths to user data and configuration

# Where to look at code and contribute

- D-Wave's Github
  - <https://github.com/dwavesystems>



The screenshot shows the GitHub profile page for D-Wave Systems Inc. The profile header includes the company logo, name, and website URL (http://www.dwavesys.com). Below the header, there are statistics for Repositories (21), People (6), and Projects (0). A search bar for repositories is present, along with filters for Type and Language. The main content area displays a list of repositories:

- dwave\_networkx**: Source repository for D-Wave NetworkX. Language: Python. Stars: 7. License: Apache-2.0. 2 issues need help. Updated a day ago.
- dwave-cloud-client**: A minimal implementation of the REST interface used to communicate with D-Wave Solver API (SAPI) servers. Language: Python. Stars: 11. License: Apache-2.0. Updated a day ago.
- dimod**: A shared API for QUBO/ising samplers. Language: Python. Stars: 14. License: Apache-2.0. 3 issues need help. Updated 3 days ago.
- dwave-system**: (Partially visible)

On the right side of the repository list, there are two sections: "Top languages" showing Python and C++ as the most used languages, and "People" showing a list of contributors.

# What if there isn't a tool for my application?

---

## OPTION 1

1. Develop BQM
2. Write Python program.
  - Set sampler to be used
  - `sampler.sample(bqm)`
  - Interpret results

# What if there isn't a tool for my application?

---

## OPTION 1

1. Develop BQM
2. Write Python program
  - Set sampler to be used
  - `sampler.sample(bqm)`
  - Interpret results

## OPTION 2

1. Develop BQM
2. Write a reusable tool.
3. Write Python program that calls that tool.
4. Submit your tool to be included in our library.

# How to contribute to Ocean

---

- Anyone can contribute to Ocean by making a pull request in GitHub.
- For a guide on how to make a pull request, check out this site:

<https://www.digitialocean.com/community/tutorials/how-to-create-a-pull-request-on-github>

# How to install

---

*Installing the Ocean Tool Suite:*

```
pip install dwave-ocean-sdk
```

- D-Wave's Github
  - <https://github.com/dwavesystems>
- Read-the-docs
  - <https://docs.ocean.dwavesys.com/en/latest/>