PRACTICAL QUANTUM COMPUTING

CINECA Workshop · November 2022

COPYRIGHT © D-WAVE SYSTEMS INC.

VICTORIA GOLIBER

SENIOR TECHNICAL ANALYST

vgoliber@dwavesys.com

 COPYRIGHT © D-WAVE SYSTEMS INC.

AGENDA

Time	Topic
09:30 – 10:45	Constrained Quadratic Models / Hybrid Solvers
Break	
11:00-12:15	Binary Quadratic Models / QPU Solvers
Break	
12:30-13:00	Use cases and wrap up
13:00-13:30	Q&A

 COPYRIGHT © D-WAVE SYSTEMS INC.

Constrained Quadratic Models

 COPYRIGHT © D-WAVE SYSTEMS INC.

POWERFUL HYBRID SOLVERS

CONSTRAINED QUADRATIC MODEL SOLVER

- Up to 500,000 variables and 100,000 constraints
- Provide separate objective and constraints
- Inequality & equality constraints

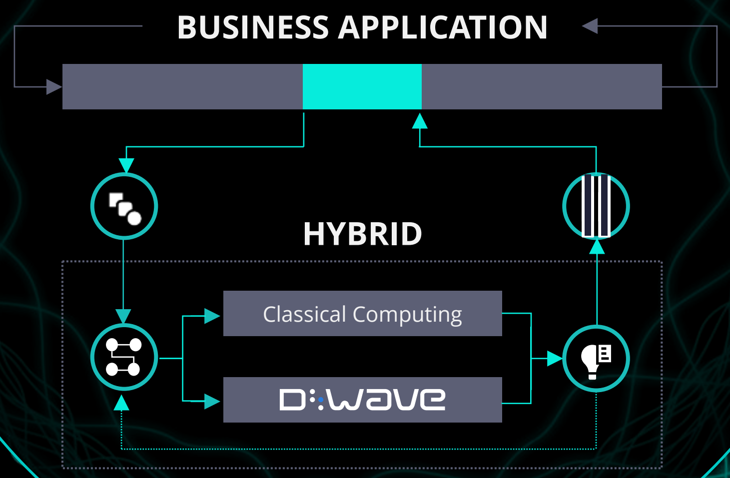
BINARY QUADRATIC MODEL SOLVER

- Up to 1,000,000 binary variables
- Full flexibility in problem formulation

DISCRETE QUADRATIC MODEL SOLVER

- Enables optimization with option selection: e.g., Choose one of 11, 19, 29
- Accepts up to 5,000 discrete variables

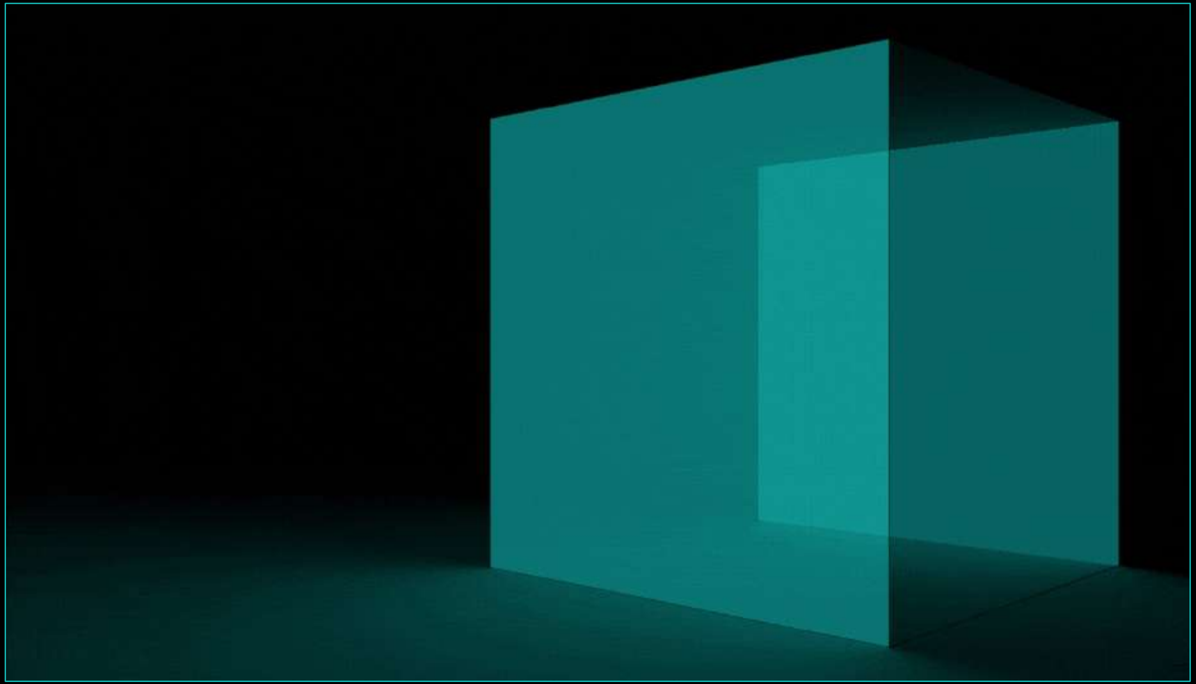
SOLVERS THAT RUN PROBLEMS ON A COMBINATION OF QUANTUM AND CLASSICAL RESOURCES



 COPYRIGHT © D-WAVE SYSTEMS INC.

CQM WITH CONTINUOUS VARIABLES

- ✓ **EXPANDS HYBRID SOLVER PORTFOLIO**
- ✓ **ALLOWS FOR MORE NATIVE REPRESENTATIONS**
- ✓ **UNLOCKS LARGER APPLICATION PROBLEMS**



FEATURES:

- Binary, integer and real/continuous variables
- Linear and quadratic terms
- Up to 100,000 constraints
- Inequality & equality constraints

 COPYRIGHT © D-WAVE SYSTEMS INC.

3D BIN PACKING

BINARY
and
CONTINUOUS
variables

GOAL: Determine the optimal placement and orientation of boxes within containers

CHALLENGES:

- Which container to put each box in?
- Where to place each box in the container?
- What orientation for each box?

This is a challenging, NP-hard problem.

COPYRIGHT © D-WAVE SYSTEMS INC.

TECHNICAL DESCRIPTION

Objective:

- Stack boxes from the floor upwards
  - Minimize average height
  - Minimize total height
- Minimize number of bins used

Constraints:

- Choose an orientation for each box
- Put each box in a container
  - Assign to one from a list
  - Ensure dimensions are within boundaries
  - Ensure boxes don’t overlap
- Place boxes into containers in order

COPYRIGHT © D-WAVE SYSTEMS INC.

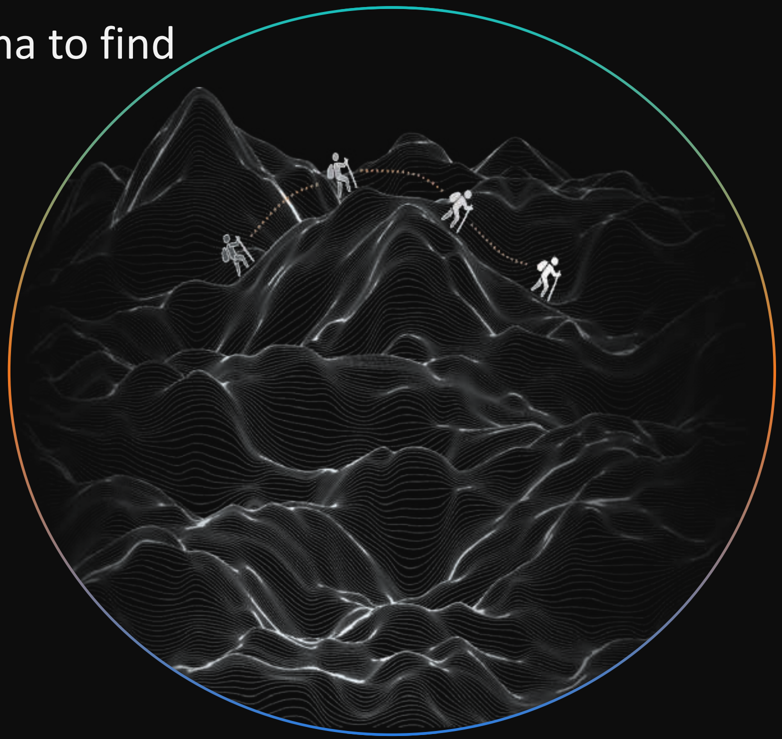
DEMO

COPYRIGHT © D-WAVE SYSTEMS INC.

Quantum Annealing

Classical solutions are sequential

- Traverse landscape to find low points
- Must climb out of any local minima to find global solution
- Energy and time intensive
- Certain problems intractable



Quantum annealing is simultaneous

- Can tunnel through the hills
- Entanglement and superposition accelerate discovery of deep valleys
- Fast and energy efficient
- Programmable, no quantum expertise needed, large scale systems deployed today

 COPYRIGHT © D-WAVE SYSTEMS INC.

QUADRATIC MODELS

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

Control bias on qubits

Control bias on couplers

 COPYRIGHT © D-WAVE SYSTEMS INC.

CQM DEVELOPMENT PROCESS

1. Write out the objective and constraints in your problem domain

2. Define the binary and/or integer variables for your problem

3. Convert the objective and constraints to math statements with binary and/or integer variables

4. Build the CQM model in Ocean from the individual objectives and constraints

 COPYRIGHT © D-WAVE SYSTEMS INC.

EXAMPLE – KNAPSACK

The knapsack problem is a well-known optimization problem. It is encountered, for example, in packing shipping containers. A shipping container has a weight capacity which it can hold. Given a collection of items to be shipped, where each item has a value and a weight, the problem is to select the optimal items to pack in the shipping container.

This optimization problem can be defined as an objective with a constraint:

- **Objective:** Maximize freight value (sum of values of the selected items)
- **Constraint:** Total freight weight (sum of weights of the selected items) must be less than or equal to the container's capacity

COPYRIGHT © D-WAVE SYSTEMS INC.

KNAPSACK – FORMULATION

Variables	Binary variable for each item	x_i
Objective	Maximize the value of the selected items	$\sum_{i=0}^{N-1} v_i x_i$
Constraints	Weight of all the selected items cannot exceed the knapsack's capacity	$\sum_{i}^{N-1} w_i x_i \leq W$
	A maximum of 2 items can fit in the knapsack	$\sum_{i=0}^{N-1} x_i \leq 2$

 COPYRIGHT © D-WAVE SYSTEMS INC.

KNAPSACK – PROGRAMMING

Variables

Objective

Constraints

Samplers minimize

```
from dimod import ConstrainedQuadraticModel, Binary, quicksum
from dwave.system import LeapHybridCQMSampler

values = [34, 25, 78, 21, 64]
weights = [3, 5, 9, 4, 2]
W = 10
n = len(values)

# Create the binary variables
x = [Binary(i) for i in range(n)]

# Construct the CQM
cqm = ConstrainedQuadraticModel()

# Add the objective
cqm.set_objective(quicksum(-values[i]*x[i] for i in range(n)))

# Add the two constraints
cqm.add_constraint(quicksum(weights[i]*x[i] for i in range(n)) <= W, label='max weight')
cqm.add_constraint(quicksum(x[i] for i in range(n)) <= 2, label='max items')

# Submit to the CQM sampler
sampler = LeapHybridCQMSampler()
sampleset = sampler.sample_cqm(cqm)

print("\nFull sample set:")
print(sampleset)
```

 COPYRIGHT © D-WAVE SYSTEMS INC.

KNAPSACK – SAMPLE SET

Energy of the objective function

Constraint satisfaction array

Solution feasibility

Variable values

	0	1	2	3	4	energy	num_oc.	is_sat.	is_fea.	object.
1	0.0	0.0	1.0	0.0	1.0	-142.0	3	arra...	False	-142.0
2	1.0	0.0	0.0	0.0	1.0	-98.0	36	arra...	True	-98.0
0	0.0	0.0	1.0	0.0	0.0	-78.0	13	arra...	True	-78.0
3	0.0	0.0	0.0	0.0	1.0	-64.0	1	arra...	True	-64.0

['INTEGER', 4 rows, 53 samples, 5 variables]

COPYRIGHT © D-WAVE SYSTEMS INC.

KNAPSACK – SAMPLE SET

```
    0    1    2    3    4 energy num_oc. is_sat. is_fea.
0 0.0 0.0 1.0 0.0 1.0 -142.0      16 arra...   False
1 1.0 0.0 0.0 0.0 1.0  -98.0      21 arra...    True
2 0.0 0.0 0.0 0.0 1.0  -64.0       1 arra...    True
['INTEGER', 3 rows, 38 samples, 5 variables]
```

Total items	Total weight	Total value
2	11	142
2	5	98
1	2	64

 COPYRIGHT © D-WAVE SYSTEMS INC.

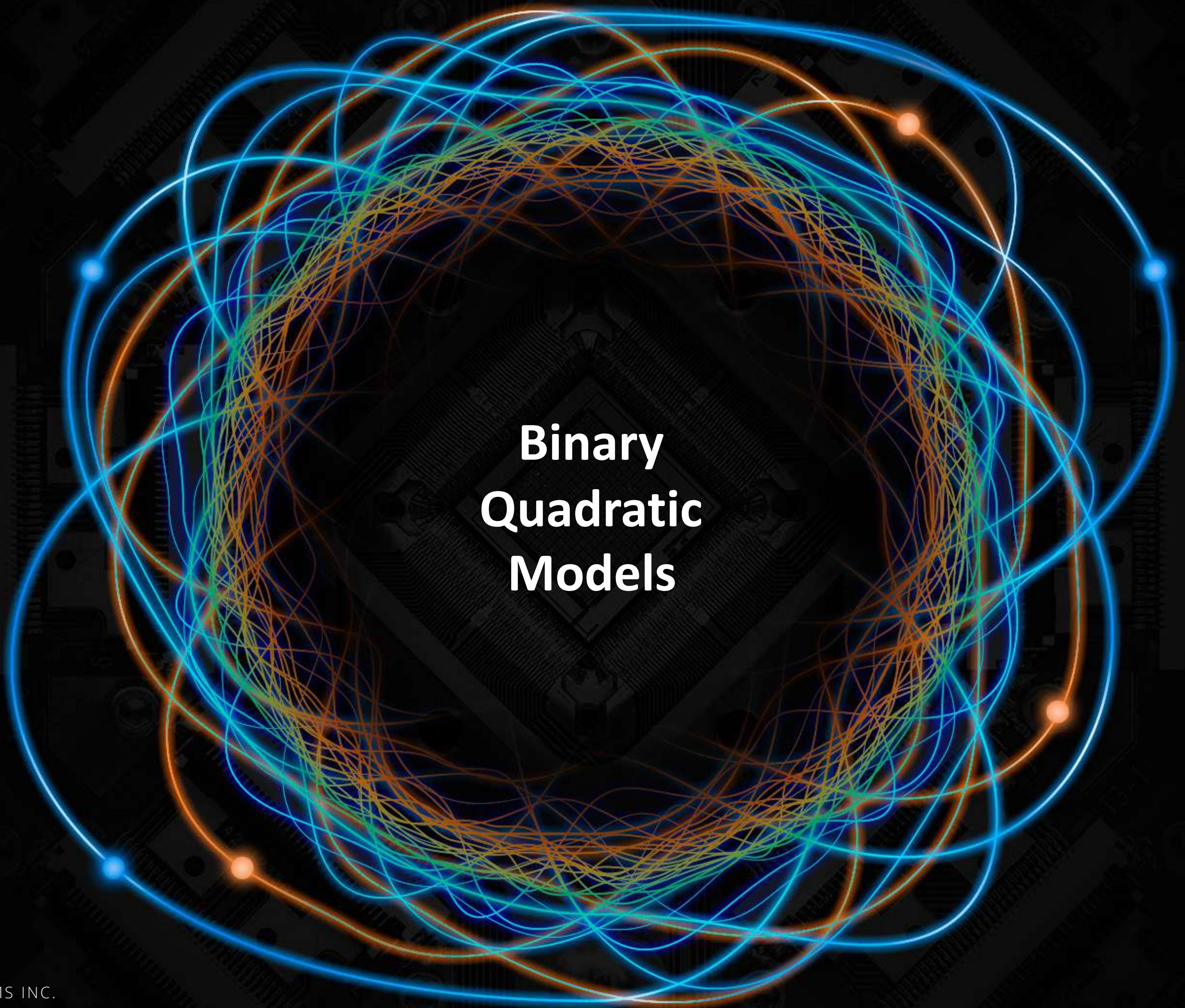
KNAPSACK – FILTERING SAMPLES

```
feasible_sampleset = sampleset.filter(lambda row: row.is_feasible)
```

 COPYRIGHT © D-WAVE SYSTEMS INC.

Questions?

 COPYRIGHT © D-WAVE SYSTEMS INC.



Binary Quadratic Models

 COPYRIGHT © D-WAVE SYSTEMS INC.

POWERFUL QUANTUM COMPUTERS BUILT FOR BUSINESS

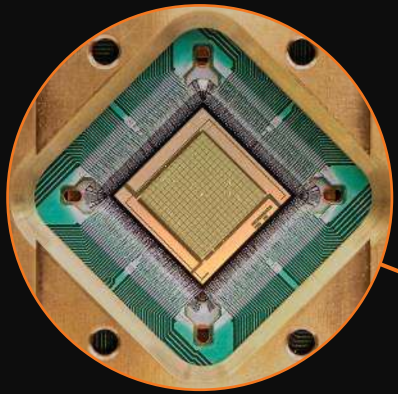
D-WAVE advantage

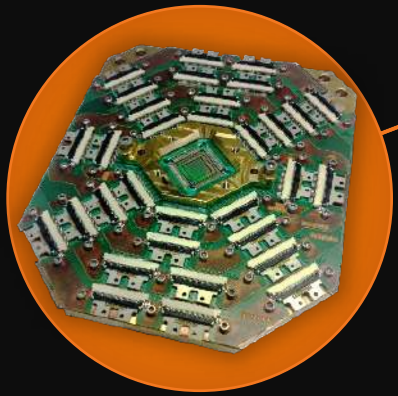
SUPPORTS HYBRID APPLICATIONS OF REAL-WORLD SIZE
Up to 1 million variables

ANNEALING QUANTUM PROCESSOR DESIGN
5,000+ qubits

ONGOING INCREASES IN COHERENCE & CONNECTIVITY

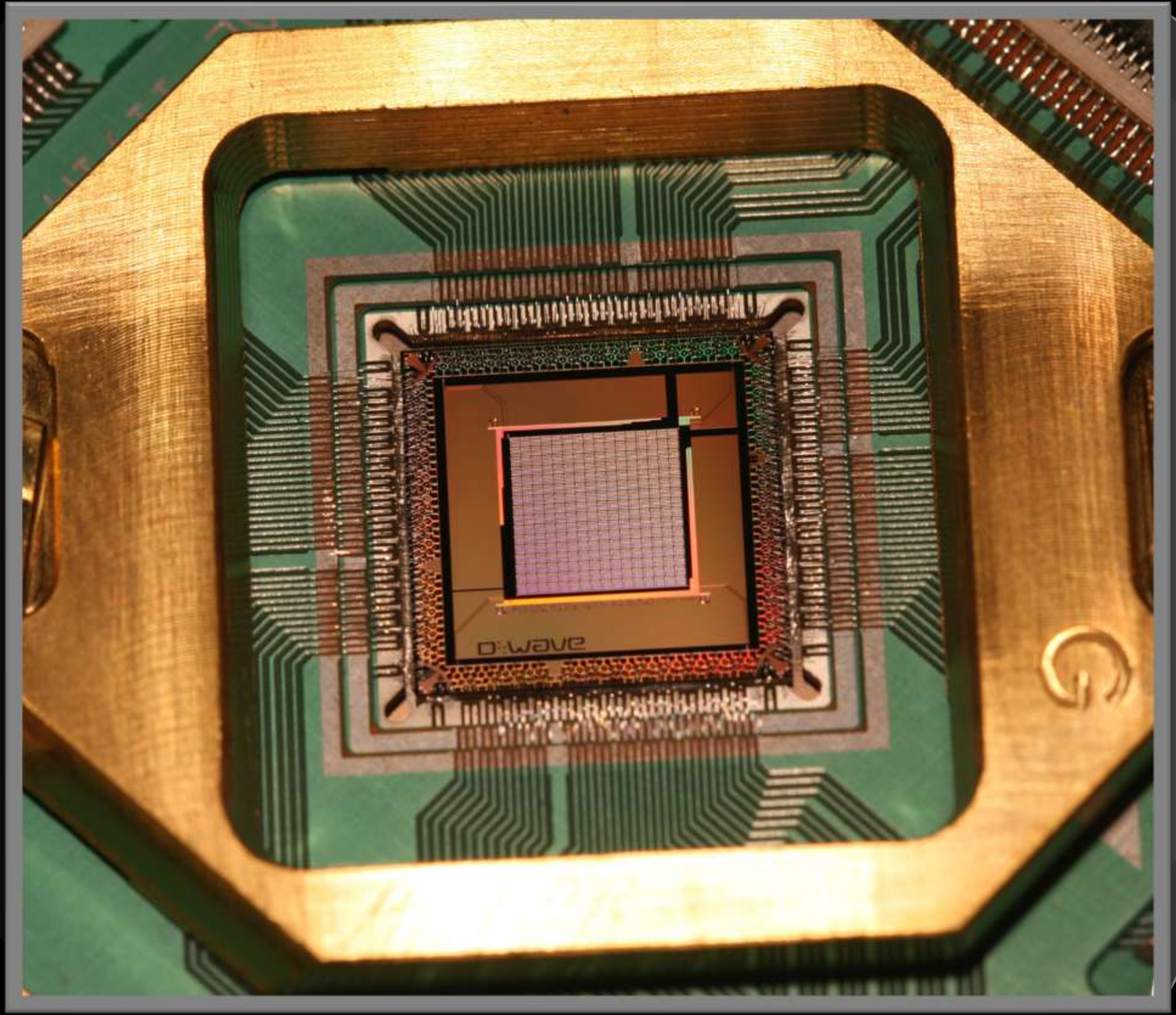
NOW SYSTEMS LOCATED IN US, EUROPE & CANADA





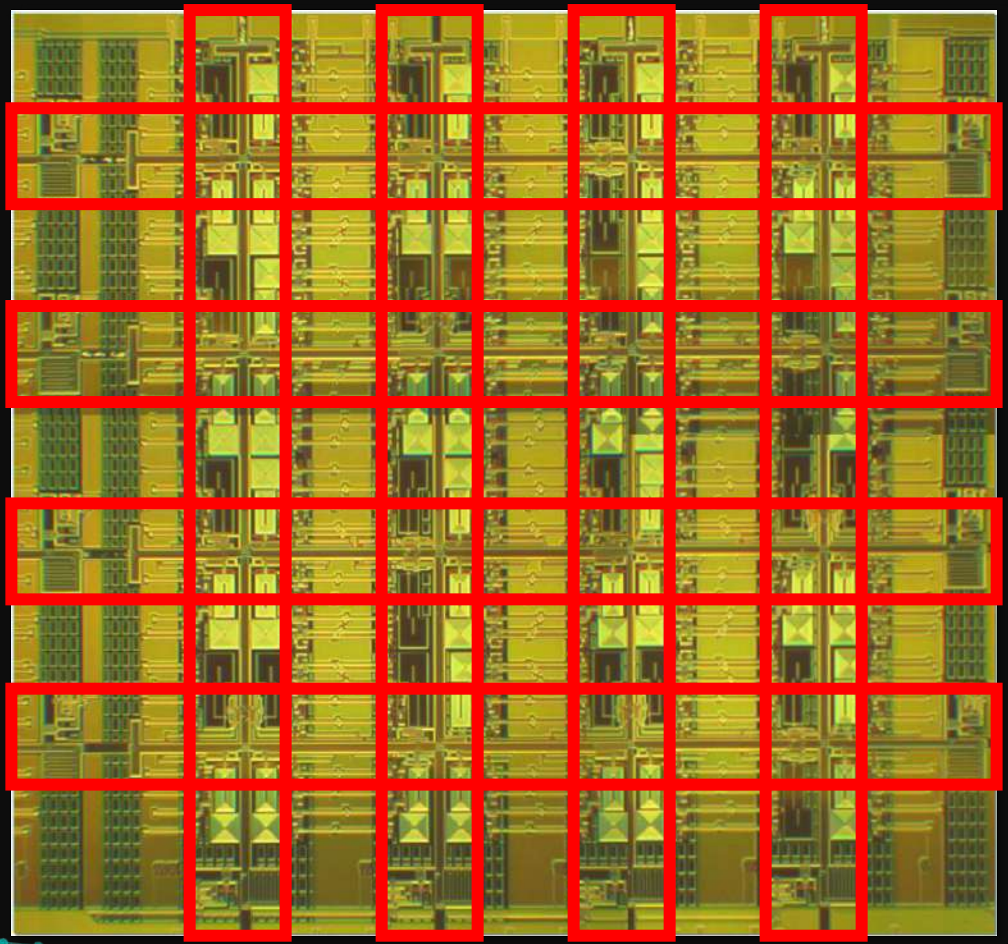
 COPYRIGHT © D-WAVE SYSTEMS INC.

D-Wave 2000Q QPU



 COPYRIGHT © D-WAVE SYSTEMS INC.

A Chimera Unit Cell



The D-Wave QPU is built from tiny loops of metal, each of which is one qubit.

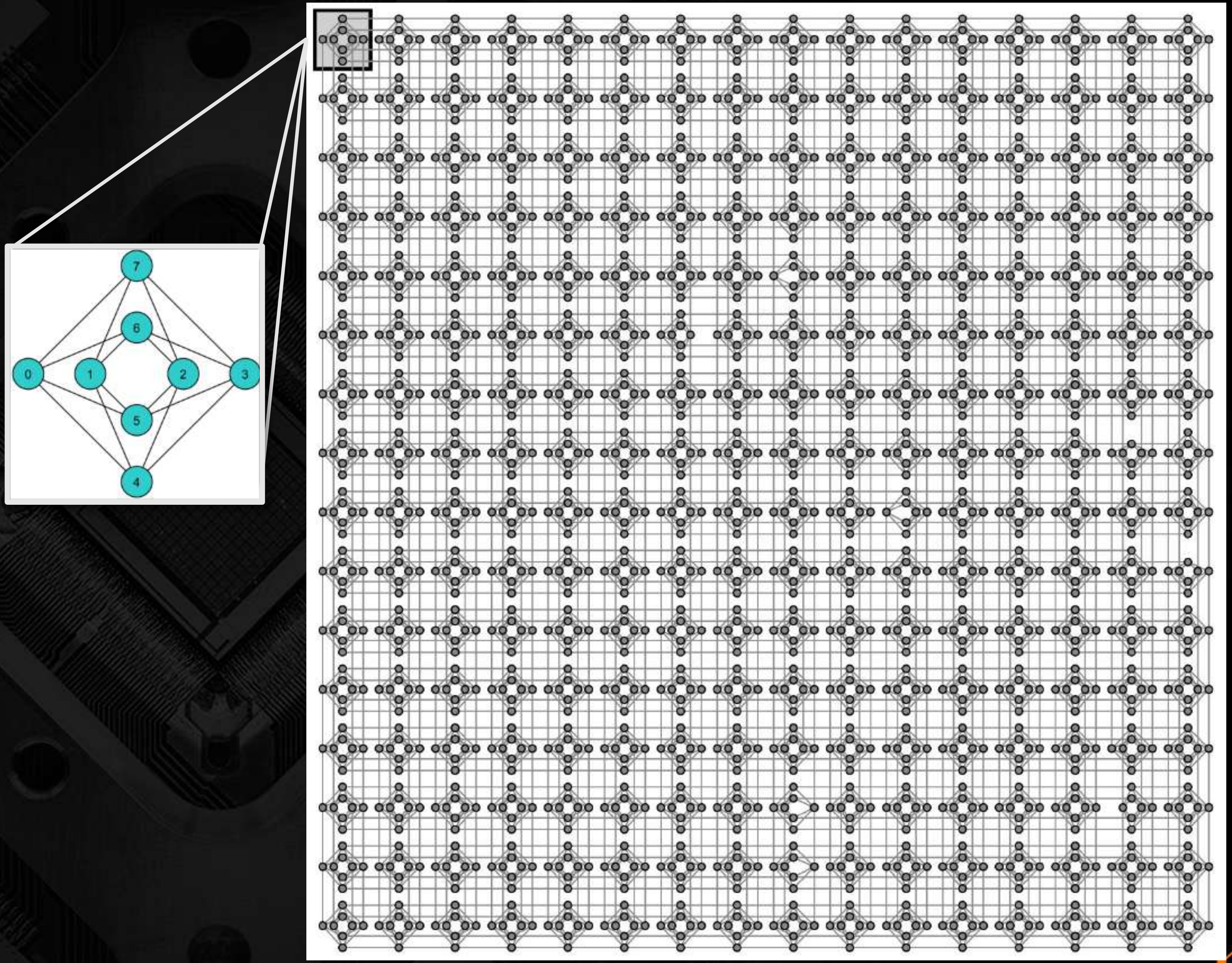
 COPYRIGHT © D-WAVE SYSTEMS INC.

D:WAVE

The Chimera Graph

The underlying architecture of the QPU looks like a graph.

In the 2000Q D-Wave System the native QPU topology is a *Chimera graph.*



 COPYRIGHT © D-WAVE SYSTEMS INC.

The Pegasus Graph

In the Advantage system the native QPU topology is a *Pegasus graph*.

 COPYRIGHT © D-WAVE SYSTEMS INC.

Quadratic Models

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

Control bias on qubits

Control bias on couplers

 COPYRIGHT © D-WAVE SYSTEMS INC.

Developing a BQM

We start with a problem in the real world (problem domain)

A Binary Quadratic Model (BQM) is a minimization function that incorporates:

- **Constraints**: The rules that we must follow
- **Objective**: What we want to minimize

General form:

$$\text{BQM} = \min\,(\text{objective}) + \gamma(\text{constraints})$$

COPYRIGHT © D-WAVE SYSTEMS INC.

BQM Development Process

1. Write out your objective and constraints in your problem domain

2. Convert your objective and constraints to math statements with binary variables

3. Modify your objective and constraints for the BQM form
   - Objective is a minimization function
   - Constraints are either:
     - Linear/quadratic equalities/inequalities, or
     - Satisfied at their minimum values

4. Combine your objective and constraints

 COPYRIGHT © D-WAVE SYSTEMS INC.

Program Structure

Ocean Program

- Establish sampler
- Define QM
- Send info to sampler
- Evaluate response

 COPYRIGHT © D-WAVE SYSTEMS INC.

Defining a BQM

Initialize: bqm = BinaryQuadraticModel(‘BINARY’)

To add “x + 5y + 8xy” to our BQM:

Terms:

```
bqm.add_variables_from( [ (‘x’, 1), ('y’, 5) ] )
bqm.add_interactions_from( [ (‘x’, ‘y’, 8) ] )
```

Symbolic:

```
x = dimod.Binary('x’)
y = dimod.Binary(‘y')
bqm = x + 5*y + 8*x*y
```

 COPYRIGHT © D-WAVE SYSTEMS INC.

Sending our problem to the sampler

BQM: sampleset = Sampler.sample(bqm)

Additional parameters are available depending on what sampler you have decided to use.

```
ocean % dwave solvers -s Advantage_system1.1
Solver: Advantage_system1.1
  Parameters:
    anneal_offsets: Anneal offsets for each working qubit, formatted as a lis...
    anneal_schedule: Annealing schedule formatted as a piecewise linear list o...
    annealing_time: Quantum annealing duration, in microseconds, as a positiv...
    answer_mode: Format of returned answers, as 'histogram' or 'raw' samples.
    auto_scale: Automatic rescaling of h and J values to their available ...
    flux_biases: Flux biases for each working qubit, in normalized offset ...
    flux_drift_compensation: Activation of flux drift compensation, as a boolean flag.
    h_gain_schedule: h-gain schedule formatted as a piecewise linear list of f...
    initial_state: Initial states to use for a reverse-anneal request, as a ...
    max_answers: Maximum number of answers to return.
    num_reads: Number of states to read (answers to return), as a positi...
    num_spin_reversal_transforms: Number of spin-reversal transforms (gauge transformations...
    programming_thermalization: Time in microseconds to wait after programming the proces...
    readout_thermalization: Time in microseconds to wait after each state is read fro...
    reduce_intersample_correlation: Addition of pauses between samples, as a boolean flag.
    reinitialize_state: Reapplication of the initial_state for every read in reve...
```

 COPYRIGHT © D-WAVE SYSTEMS INC.

What does a sampleset look like?

```
   w  x  y  z energy num_oc. chain_.
0  0  0  0  1    0.0      18     0.0
1  1  1  1  0    0.0       3     0.0
2  1  0  0  1    0.0      31     0.0
3  1  1  1  1    0.0       4     0.0
4  0  0  0  0    0.0      12     0.0
5  0  1  1  0    0.0       6     0.0
6  0  0  1  0    0.0       4     0.0
7  1  1  0  1    0.0      22     0.0
['BINARY', 8 rows, 100 samples, 4 variables]
```


COPYRIGHT © D-WAVE SYSTEMS INC.

Example

Suppose we have this list of projects we might work on, and each project has the associated profit.

How should we formulate a BQM to maximize profit?

What would the expected solution be?

How would we write a program to get this solution?

Project	Profit
P1	20
P2	18
P3	22
P4	26
P5	21

COPYRIGHT © D-WAVE SYSTEMS INC.

Example

Now suppose that we:

1. Want to maximize profit, and
2. Must choose either P1 or P2, but not both.

How would we formulate this constraint?

How can we add this to our program?

Project	Profit
P1	20
P2	18
P3	22
P4	26
P5	21

 COPYRIGHT © D-WAVE SYSTEMS INC.

Equality Constraints

Constraint:

$$\left(\sum a_i x_i\right) + k = 0$$

1. **Build a BQM object**

2. **Use bqm.add_linear_equality_constraint(…) to add the constraint**

    Parameters:

    - Terms: [(var1, coeff1), (var2, coeff2), …]
    - Constant: k in above equation
    - Lagrange_multiplier: weight for constraint

Example:

Choose 2 numbers that sum to 5

from the set A = {1, 2, 3, 4}.

```
from dimod import BinaryQuadraticModel

bqm = BinaryQuadraticModel('BINARY')

bqm.add_linear_equality_constraint(
    [(1,1), (2,1), (3,1), (4,1)],
    constant = -2,
    lagrange_multiplier = 1)

bqm.add_linear_equality_constraint(
    [(1,1), (2,2), (3,3), (4,4)],
    constant = -5,
    lagrange_multiplier = 1)
```

 COPYRIGHT © D-WAVE SYSTEMS INC.

Questions?

 COPYRIGHT © D-WAVE SYSTEMS INC.

Industry
Use Cases

 COPYRIGHT © D-WAVE SYSTEMS INC.

APPLICATIONS ACROSS KEY VERTICALS

LOGISTICS

Shipping container logistics
Employee scheduling
Farm to market food delivery
Last mile vehicle routing

PHARMA

Protein folding
Clinical trials
Drug discovery

FINANCE

Portfolio risk reduction and return optimization
Marketing campaign optimization
Fraud detection

 COPYRIGHT © D-WAVE SYSTEMS INC.

REDUCE WASTE IN THE AUTOMOTIVE SUPPLY CHAIN

80%

REDUCTION IN WASTE

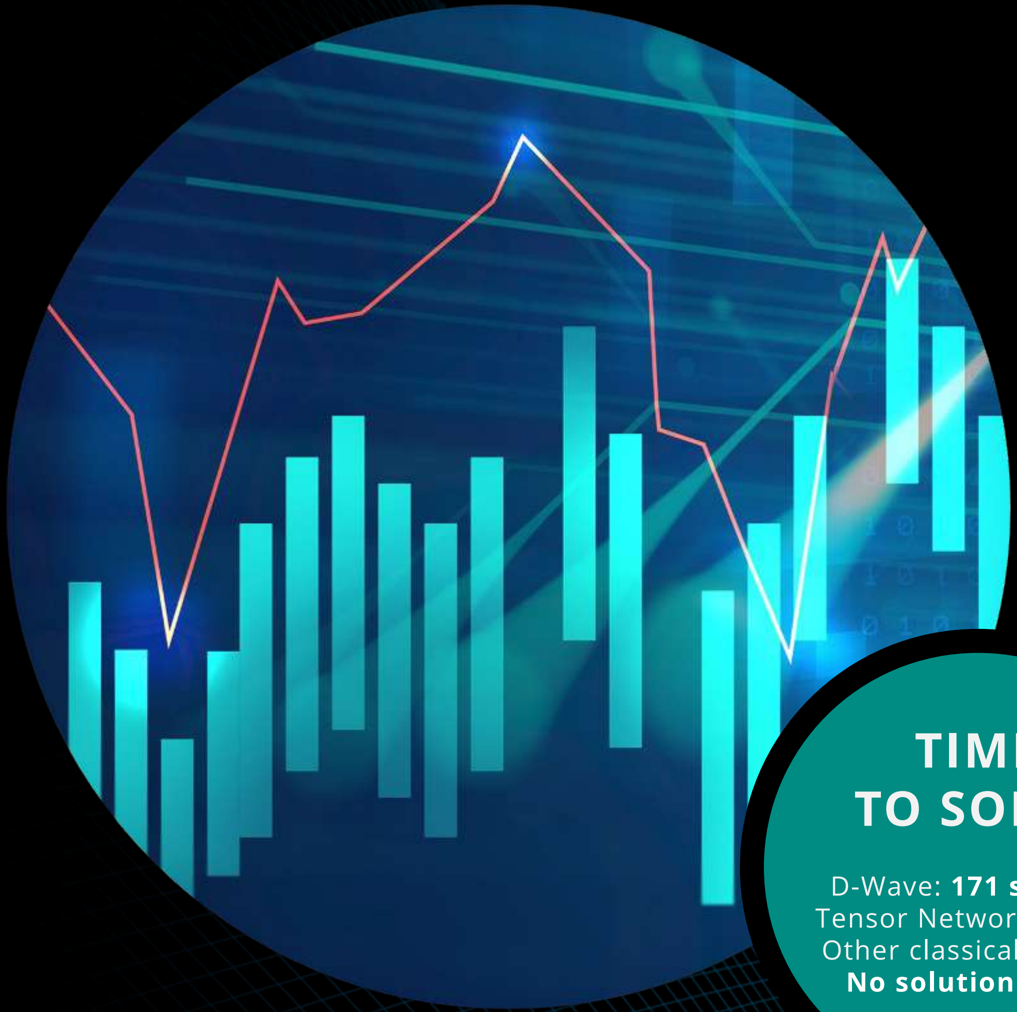
Volkswagen

"By continuing to research and develop these types of algorithms, we hope to have a significant impact on Volkswagen's core business throughout multiple units. This application has immediate, real-world implications for production and logistics."

—VOLKSWAGEN QUANTUM COMPUTING RESEARCHER SHEIR YARKONI

COPYRIGHT © D-WAVE SYSTEMS INC.

A QUANTUM SOLUTION FOR PREDICTING PROFITABILITY



TIME TO SOLVE

D-Wave: **171 seconds**
Tensor Networks: **1 day**
Other classical solvers: **No solution found**

BBVA

Maximum value at lowest risk
With 10 ^ 382 possible portfolios, the hybrid quantum/classical system made short work of it.

"These results are exciting because they really show a commercially valuable application of quantum computing today."

—SAM MUGEL, CTO, MULTIVERSE

 COPYRIGHT © D-WAVE SYSTEMS INC.

BUSINESS-CRITICAL GROCERY TASKS IN MINUTES INSTEAD OF HOURS

< 2 MINUTES

25 HOURS
TO < 2 MINUTES
4K COMBINATIONS
& 4M VARIABLES

save on foods

"What Advantage gives us, is the ability to seamlessly integrate quantum into our business problems. We've been able to decrease the amount of time to get a result from 25 hours down to seconds."

—ANDREW DONAHER, VP DIGITAL & ANALYTICS, SAVE-ON-FOODS

 COPYRIGHT © D-WAVE SYSTEMS INC.

LOGISTICS OPTIMIZATION AT PORT OF LA WITH QUANTUM COMPUTING

60%

MORE CARGO HANDLED EACH DAY PER CRANE

SAVANT X

D-Wave’s quantum system is used as part of the SavantX HONE optimization engine at the Port of Los Angeles. The goal is to expedite delivery of containers out of the terminal while increasing the amount of cargo that can be handled.

> “With HONE and D-Wave, each huge crane handled 60% more cargo per day, while the turnaround time for trucks was reduced by 12%.”
>
> — SAVANTX TEAM

COPYRIGHT © D-WAVE SYSTEMS INC.

OPTIMIZING THE RENEWABLE ELECTRIC GRID

600x

FASTER WITH D-WAVE THAN CLASSICAL

E.ON

Explored how D-Wave's quantum hybrid technology can more efficiently manage modern power grids, which have an increasingly diverse and decentralized set of generating facilities.

"The IEEE 118-bus test case shows time performance speed-up. The increment in performance would enable real time planning and operations of electrical grids."

—E.ON TEAM

 COPYRIGHT © D-WAVE SYSTEMS INC.

A QUANTUM SOLUTION FOR INVESTMENT APPLICATIONS

90%

REDUCTION IN TIME TO SOLUTION for HEDGING APP

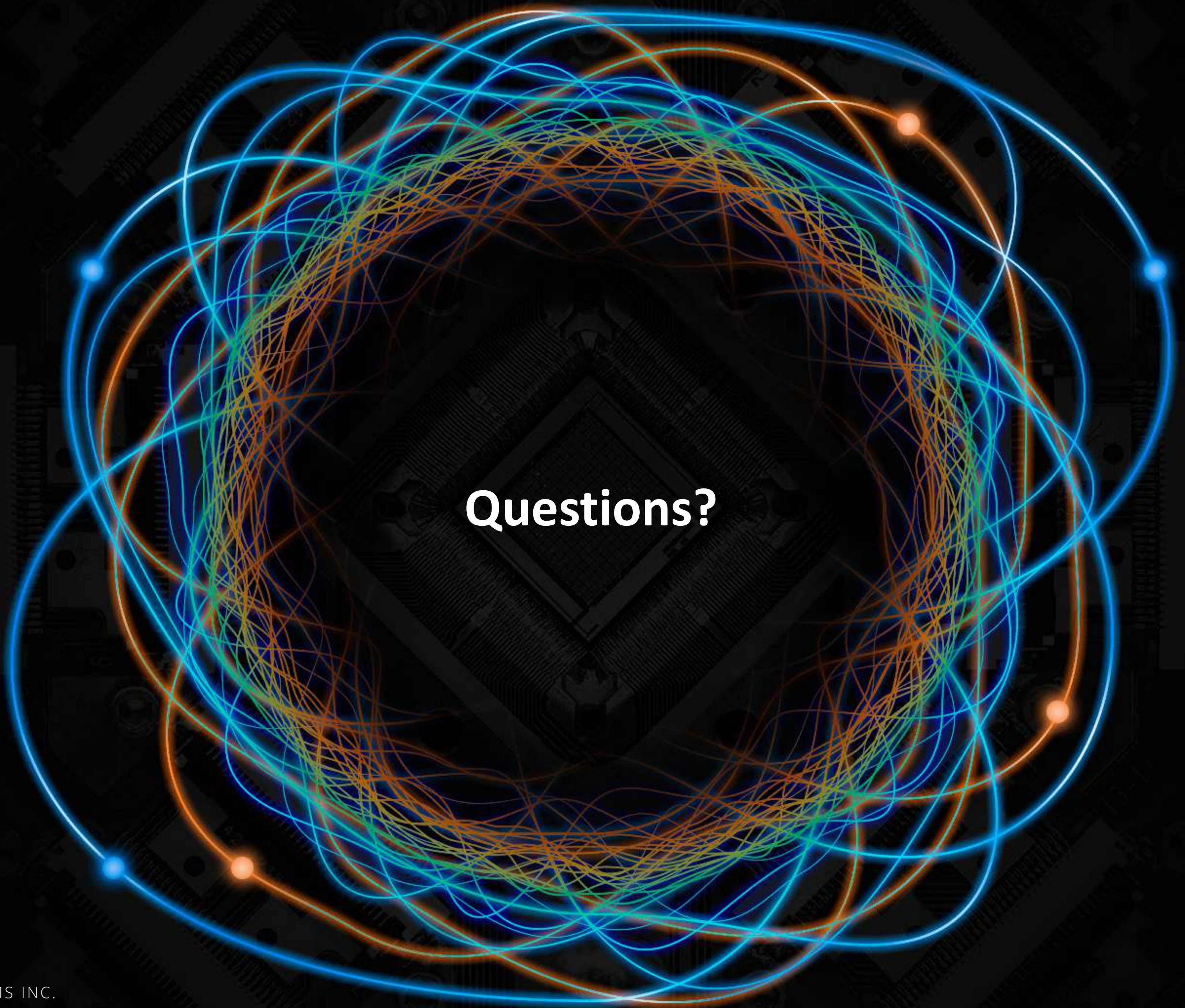
CaixaBank

The D-Wave hybrid solver significantly decreased compute time to minimize the capital needed for hedging operations, optimized investment portfolios, and increased a bond portfolio internal rate of return (IRR).

“What normally took the bank several hours of compute time was reduced to just minutes via quantum computing technology – an up to 90% decrease in compute time over the traditional solution.”

—CAIXA BANK TEAM

 COPYRIGHT © D-WAVE SYSTEMS INC.



COPYRIGHT © D-WAVE SYSTEMS INC.