Hybrid Computing Platform for Combinatorial Optimization with the Coherent Ising Machine

J. Arai, S. Yagi, H. Uchiyama, T. Honjo, T. Inagaki, K. Inaba, T. Ikuta, H. Takesue, K. Horikawa (Nippon Telegraph and Telephone Corporation, Japan)

Email: junya.arai.at@hco.ntt.co.jp

NTT

Introduction

Background

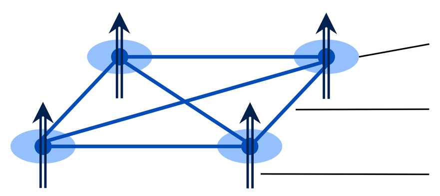
- Various **combinatorial optimization** problems (e.g., TSP and graph coloring) are reducible to **Ising optimization** problems
- **Ising computers** are hardware specialized for Ising optimization
  - E.g., D-Wave, Fujitsu Digital Annealer, and **LASOLV** (laser solver), which is a **coherent Ising machine (CIM)** implemented by NTT [1].
- Cloud platforms for "Ising computers as a service" play an important role in promoting the use of Ising computers
  - E.g., D-Wave Leap and Fujitsu Digital Annealer Cloud Service
- **Hybrid computation** using both Ising computers and conventional **digital computers** is a key to solve real-world problems
- However, communication costs make hybrid computation inefficient

Contribution

1. Clarifying issues in the design of Ising computing platforms
2. **LASOLV Computing System (LCS)** as an answer to the issues

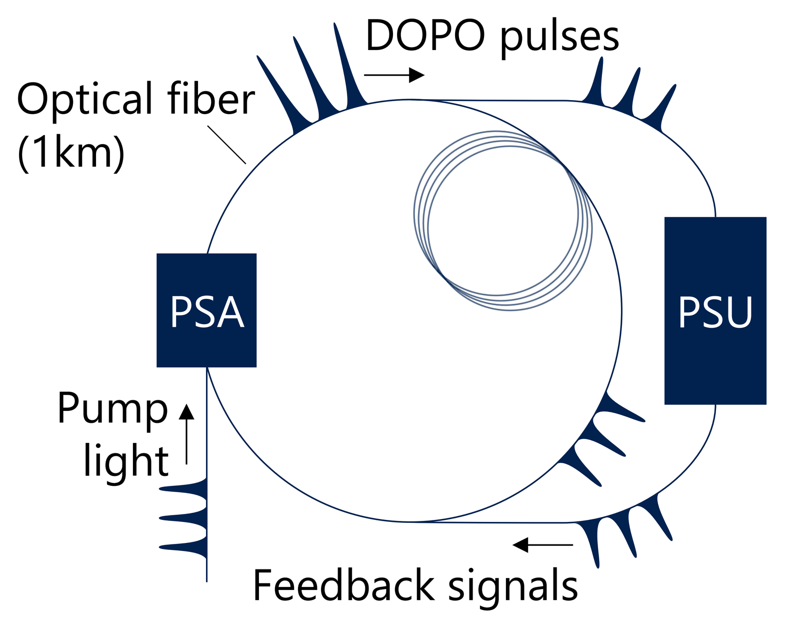
Coherent Ising Machine

- Ising optimization is a ground-state search of the **Ising model**, which represents a network of spins



- The ground state is defined as the spin configuration that minimizes the **Ising Hamiltonian** $H = -\sum_{i<j} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i$
- The CIM simulates the Ising model using photonics technologies
  - Efficient for dense networks compared with D-Wave 2000Q [2]

Appearance of LASOLV

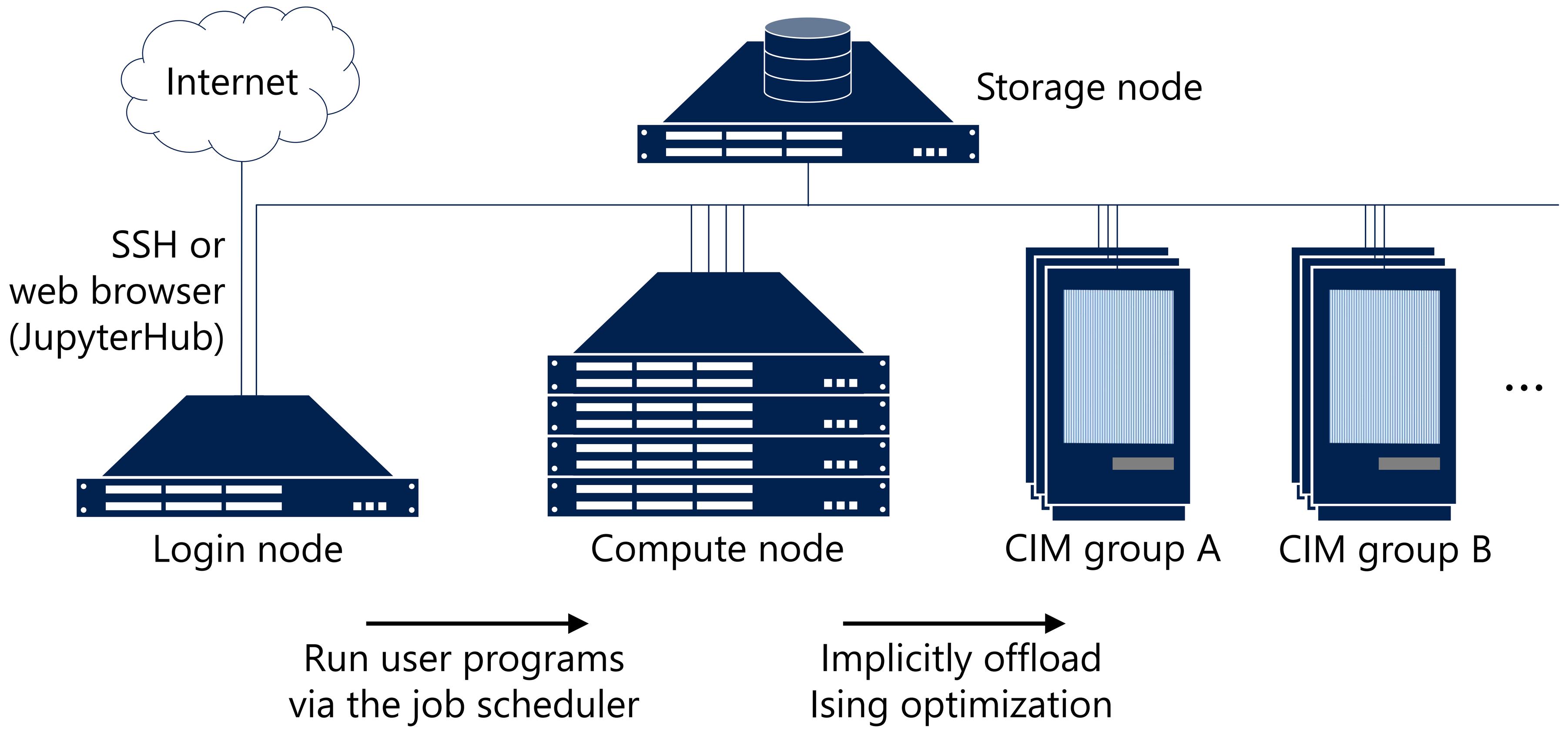


Spin: degenerated optical parametric oscillator (DOPO) pulses in the loop of an optical fiber.
Interaction and magnetic field: feedback signals calculated in the PSU using J, h, and amplitude of the pulses.
Solution: final phase configuration of the pulses

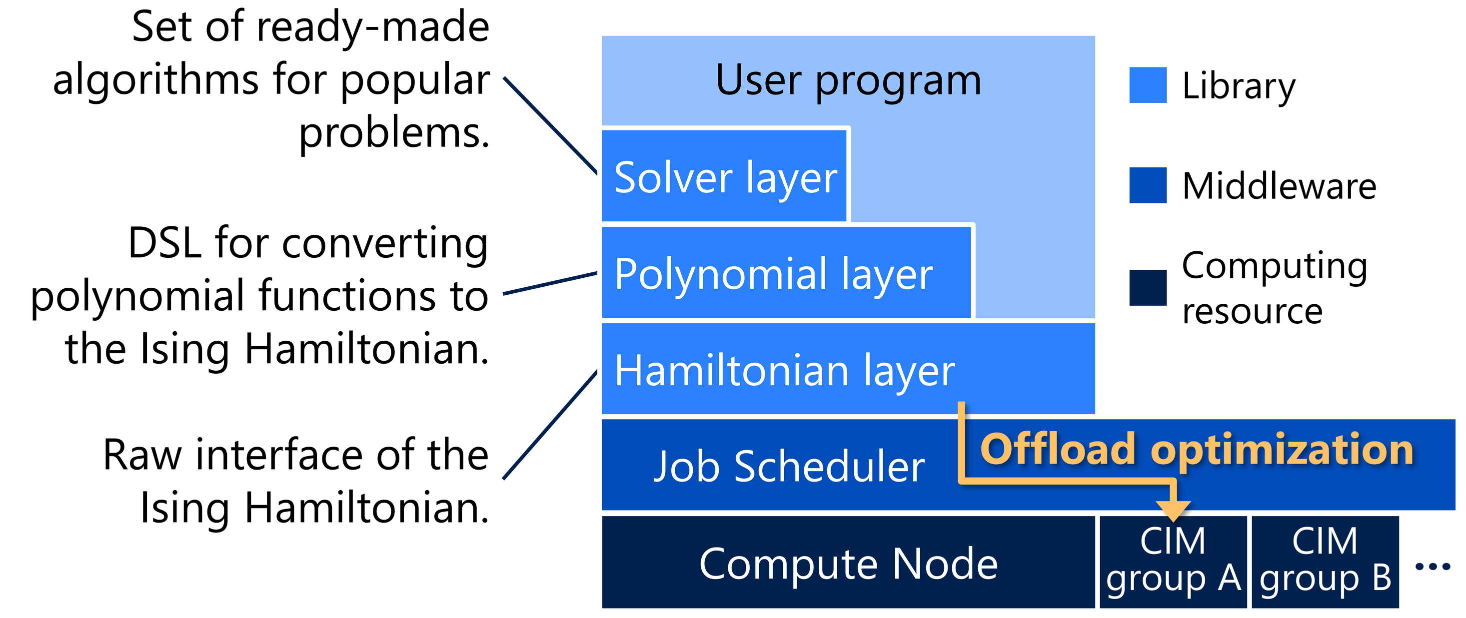
PSA: phase sensitive amplifier
PSU: problem setting unit

LASOLV Computing System

Issues in the design of Ising computing platforms	Our solution
Efficiency. On the existing cloud platforms, **hybrid algorithms involve frequent communication via the Internet** between users' digital computers and Ising computers. This overheads degrade the performance.	**CIM-digital integration.** **LCS co-locates CIMs and digital computers** in a single cluster and allows users to run their programs there for offering efficient CIM-digital communication.
Extensibility. Ising computers are evolving, and so new-generation hardware will have new specifications (e.g., # of spins). **Heterogeneous computing resources** need to be handled effectively.	**Group dispatch.** **The CIMs are grouped by compatibility.** Ising optimization is implicitly offloaded to CIMs in the group that satisfies users' requirements.
Productivity. While **by-hand Ising reduction requires specialized skills**, it is impractical to provide problem-specific reduction algorithms for every use case.	**Layered reduction.** **LCS provides a Python library for assisting in Ising reduction**. It consists of three layers with different degrees of flexibility, and users can choose eligible one from them.



System configuration



Software/hardware stack

Example: graph coloring

Steps of by-hand Ising reduction

1. Define the problem. "Assign one of k colors to each of n vertices in graph G without using the same color to the adjacent vertices."

2. Set up a polynomial objective function. Using variable $x_{v,c} \in \{0,1\}$ and considering vertex v has color c if $x_{v,c} = 1$, the problem is reduced to minimization of the function [3]:

$$f(\boldsymbol{x}) = \sum_{u<v} A_{u,v} \sum_{c=1}^{k} x_{u,c}x_{v,c} + p\sum_{v=1}^{n}\left(\sum_{c=1}^{k} x_{v,c} - 1\right)^2$$

3. Make the Ising Hamiltonian. By deforming $f(\boldsymbol{x})$, we obtain J and h of the Ising Hamiltonian s.t. $f(\boldsymbol{x}) = -\sum_{i<j} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i$ where $\sigma_{vk+c} = 2x_{v,c} - 1$.

Programming on LCS

Use the **solver layer** if it has the problem-specific solver.

```
# A: adjacency matrix of G. p: hyper parameter.
answer = GraphColoringSolver(A, k, p).solve()
```

If more flexibility is needed, minimize an objective function using the **polynomial layer**.

```
f =
  sum(A(u,v) * sum(x(u,c) * x(v,c) for c in range(k))
      for u, v in combinations(range(n), 2))
  + p * sum((sum(x(v,c) for c in range(k)) - 1) ** 2
        for v in range(n))
answer = PolynomialMetasolver(f).solve()
```

Input J and h to the **Hamiltonian layer**. It also provides decomposition heuristics to solve problems larger than the capacity of the CIM.

```
answer = IsingModelMetasolver(J, h).solve()
```

Current Status

LCS is available to research collaborators from this autumn in the small-start configuration, which consists of a 64-core machine serving as the login-compute-storage node and one CIM. We are evaluating LCS based on the usage statistics and the feedbacks.

References

[1] T. Inagaki et al. 2016. A coherent Ising machine for 2000-node optimization problems. Science.
[2] R. Hamerly et al. 2019. Experimental investigation of performance differences between coherent Ising machines and a quantum annealer. Science Advances.
[3] A. Lucas. 2014. Ising formulations of many NP problems. Frontiers in Physics.