Hybrid Computing Platform for Combinatorial Optimization with the Coherent Ising Machine

Junya Arai
NTT Corporation

Satoshi Yagi
NTT Corporation

Hiroyuki Uchiyama
NTT Corporation

Toshimori Honjo
NTT Corporation

Takahiro Inagaki
NTT Corporation

Kensuke Inaba
NTT Corporation

Takuya Ikuta
NTT Corporation

Hiroki Takesue
NTT Corporation

Keitaro Horikawa
NTT Corporation

ABSTRACT

Several institutes are operating cloud platforms that offer Web API access to Ising computers such as quantum annealing machines. Platform users can solve complex combinatorial optimization problems by using hybrid algorithms that utilize both users' conventional digital computers and remote Ising computers. However, communication via the Internet takes an order of magnitude longer time than optimization on Ising computers. This overheads seriously degrade the performance of hybrid algorithms since they involve frequent communication. In this poster, we first state issues in the design of Ising computing platforms, including communication overheads. Then, we answer the issues by introducing the computing platform for the coherent Ising machine (CIM), an Ising computer based on photonics technologies. Our platform offers efficient CIM-digital communication by allowing users to execute their program on digital computers co-located with the CIM. We have released the platform to our research collaborators in this autumn and started the evaluation.

KEYWORDS

coherent Ising machine, combinatorial optimization, non-von Neumann computer, computer cluster

1 INTRODUCTION

Although computing performance is approaching saturation due to the limit of Moore's Law, we are still facing complex problems that overwhelm today's conventional digital computers. *Combinatorial optimization* is one of such problems and ubiquitously appears in scientific and industrial problems [11, 16, 17, 22]. Various instances of combinatorial optimization are reducible to a search problem for the ground state of the *Ising model* [18]. The Ising model is a theoretical model that represents a network of spins. Spin σ_i takes one of the two states, up (+1) or down (−1). Letting J_{ij} and h_i be the strengths of an interaction and a magnetic field, respectively, the ground state is defined as the spin configuration that minimizes the *Ising Hamiltonian* $H = -\sum_{i<j} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i$.

To overcome the performance limitation of digital computers, several institutes have developed computers specialized for Ising optimization such as quantum annealing machines [5, 12] and quantum-inspired semiconductor processors [21, 24]. We term them *Ising computers* in this poster. We are also developing the *coherent Ising machine* (CIM) [15] and call our implementation *LASOLV* (laser solver). While their only function is to solve limited-scale Ising problems, hybrid use with digital computers enables Ising computers to solve complex problems [10, 19, 25]. For example, decomposition heuristics solve large Ising problems by iteratively generating a small subproblem on digital computers and solving it with Ising computers [4, 20, 23]. Thus, Ising computers are widely applicable to real-world use cases.

For promoting the use of Ising computers, some of them are provided as remote computing resources on cloud platforms [2, 9]. Platform users execute their program on local digital computers, and the program offloads Ising optimization to the platform by making a Web API request. However, the existing platforms are blind to the issue of communication overheads. Communication time through the Internet is on the order of *at least* 10 milliseconds [14], and it is too long to fully profit from Ising computers whose solution time is in the order of one millisecond [13]. Communication costs affect the performance of hybrid algorithms even more since they involve frequent communication between Ising computers and digital computers. For dealing with real-world problems, efficient communication is as important as hybrid algorithms.

In this poster, we make two contributions: (i) clarifying the issues in the design of Ising computing platforms, including communication overheads, and (ii) answering the issues by introducing *LASOLV Computing System* (LCS), which is a platform for LASOLV, our implementation of the CIM. As far as we know, LCS is the first Ising computing platform that accepts arbitrary user-defined hybrid algorithms. Specifically, LCS executes users' program on digital computers co-located with the CIM for efficient CIM-digital communication. LCS offers two more features: the group dispatch for effectively utilizing multiple CIMs and the layered reduction for productive programming. We have released LCS to research collaborators in this autumn and started evaluation based on the usage statistics and feedback.

2 COHERENT ISING MACHINE

The CIM [15] is a photonics-based machine that simulates the Ising model. Figure 1 shows its configuration. Spin σ_i corresponds to a phase of the i-th degenerate optical parametric

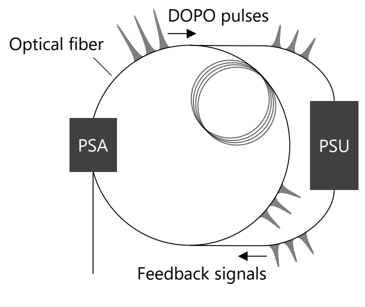

Figure 1: Configuration of the CIM.

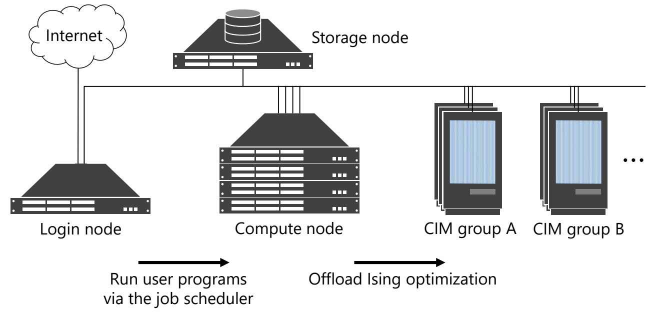

Figure 2: Configuration of LCS.

oscillator (DOPO) pulse in the loop. The phase sensitive amplifier (PSA) on the loop most efficiently amplifies 0 or π phase components of DOPO pulses, and thus each pulse oscillates only at phases 0 or π at above the oscillation threshold. Each phase corresponds to $\sigma = -1$ and 1, respectively. J_{ij} and h_i are set in the problem setting unit (PSU). The CIM emulates couplings between a pair of pulses via measuring amplitude of the pulses and feeding back signals calculated in the PSU for each pulse. The solution is obtained as the final phase configuration of the pulses. With a high probability, the pulses take a ground state, which is a phase configuration that minimizes the energy. Compared with D-Wave 2000Q [12], the CIM has advantages such as room-temperature operation, all-to-all spin connection, and efficiency in dense networks [13]. Currently the CIM can simulate up to 2,000 spins; moreover, we are aiming to extend it to 100,000 spins.

3 ISSUES IN PLATFORM DESIGN

In this section, we state three issues in the design of Ising computing platforms.

Efficiency. As mentioned in Section 1, communication via the Internet takes an order of magnitude longer time than Ising optimization. The communication efficiency has increasingly large impacts since the data size of an Ising problem (i.e., J_{ij} and h_i) gets larger along with the growth in the spin capacity of Ising computers. Hence, we must design the platforms so that communication overheads are minimized.

Extensibility. Ising computers are evolving continuously. Since new-generation hardware will be developed with new specifications, the platforms inevitably include heterogeneous computing resources for Ising optimization. They need to be utilized effectively and provided in an easy-to-use manner.

Productivity. While by-hand Ising reduction requires specialized skills and is time-consuming, it is impractical to provide reduction algorithms for every problem. To handle broad use cases, we need a universal interface that offers both convenience and generality.

4 LASOLV COMPUTING SYSTEM

LCS is a platform for computation using the CIM; specifically, it is a cluster equipped with software such as Slurm [26] for job scheduling and a Python library for assisting in Ising reduction. LCS offers the following features that correspond to the design issues described in Section 3, respectively.

CIM-digital integration. For efficient CIM-digital communication, we integrate the CIMs and digital computers in the cluster. Figure 2 shows the configuration of LCS. Users submit their jobs via SSH or JupyterHub [7], and then the job scheduler executes it on a compute node. When the user program requests Ising optimization, it is implicitly offloaded to CIMs with efficient communication in the cluster.

Group dispatch. LCS has a scale-out capability. Since specifications of the CIMs may vary, LCS makes groups of compatible CIMs. Users can select a target group for the offloading, and the job scheduler dispatches optimization requests to one of the CIMs in the group. Note that it is not difficult to automatically determine a target group from properties of the Ising problem such as the number of spins. We have not yet implemented it, but we are going to do it for usability and effective resource utilization.

Layered reduction. The library provides the following layers so that users can choose a suitable one: (i) the solver layer, a set of reduction algorithms for popular NP-hard problems such as the traveling salesman problem and the graph coloring problem, (ii) the polynomial layer, the embedded domain-specific language to convert polynomial objective functions to Ising problems, and (iii) the Hamiltonian layer, raw interface of Ising problems. The polynomial layer is helpful when the solver layer does not fit users' problem, or users need more flexibility for hand tuning. For solving large Ising problems, the Hamiltonian layer also provides decomposition heuristics based on previous studies [4, 20, 23].

The LCS library instantly provides users our research-and-development outcomes such as the decomposition heuristics, and this is why we have developed our original library while there are open source libraries for Ising computing [1, 3, 6, 8].

5 CONCLUSION

In this poster, we stated the issues in platform design and introduced LCS. 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 currently evaluating LCS based on the usage statistics and the feedback.

REFERENCES

[1] [n.d.]. Blueqat/Wildqat: Python Framework for QUBO. https://github.com/Blueqat/Wildqat.

[2] [n.d.]. D-Wave Leap. https://cloud.dwavesys.com/leap/.

[3] [n.d.]. D-Wave's Ocean Software. https://ocean.dwavesys.com/.

[4] [n.d.]. dwavesystems/dwave-hybrid: Hybrid Asynchronous Decomposition Sampler prototype framework. https://github.com/dwavesystems/dwave-hybrid.

[5] [n.d.]. NEC's Quantum Annealing Technology. https://www.nec.com/en/global/rd/technologies/quantum_annealing.html.

[6] [n.d.]. OpenJij/OpenJij: OpenJij : Framework for the Ising model and QUBO. https://github.com/OpenJij/OpenJij.

[7] [n.d.]. Project Jupyter — JupyterHub. https://jupyter.org/hub.

[8] [n.d.]. recruit-communications/pyqubo: Python DSL for constructing QUBOs from mathematical expressions. https://github.com/recruit-communications/pyqubo.

[9] [n.d.]. Services : Fujitsu Global. https://www.fujitsu.com/global/digitalannealer/services/.

[10] Alastair A Abbott, Cristian S Calude, Michael J Dinneen, and Richard Hua. 2018. A Hybrid Quantum-Classical Paradigm to Mitigate Embedding Costs in Quantum Annealing. *CoRR* abs/1803.0 (2018), 1–34.

[11] Maggie Xiaoyan Cheng, Yingshu Li, and Ding-Zhu Du. 2006. *Combinatorial Optimization in Communication Networks.*

[12] Elizabeth Gibney. 2017. D-Wave upgrade: How scientists are using the world's most controversial quantum computer. *Nature* 541, 7638 (2017), 447–448.

[13] Ryan Hamerly, Takahiro Inagaki, Peter L McMahon, Davide Venturelli, Alireza Marandi, Tatsuhiro Onodera, Edwin Ng, Carsten Langrock, Kensuke Inaba, Toshimori Honjo, Koji Enbutsu, Takeshi Umeki, Ryoichi Kasahara, Shoko Utsunomiya, Satoshi Kako, Ken-ichi Kawarabayashi, Robert L Byer, Martin M Fejer, Hideo Mabuchi, Dirk Englund, Eleanor Rieffel, Hiroki Takesue, and Yoshihisa Yamamoto. 2019. Experimental investigation of performance differences between coherent Ising machines and a quantum annealer. *Science Advances* 5, 5 (2019), 1–10.

[14] Toke Høiland-Jørgensen, Bengt Ahlgren, Per Hurtig, and Anna Brunstrom. 2016. Measuring Latency Variation in the Internet. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies.* 473–480.

[15] Takahiro Inagaki, Yoshitaka Haribara, Koji Igarashi, Tomohiro Sonobe, Shuhei Tamate, Toshimori Honjo, Alireza Marandi, Peter L McMahon, Takeshi Umeki, Koji Enbutsu, Osamu Tadanaga, Hirokazu Takenouchi, Kazuyuki Aihara, Ken-Ichi Kawarabayashi, Kyo Inoue, Shoko Utsunomiya, and Hiroki Takesue. 2016. A coherent Ising machine for 2000-node optimization problems. *Science* 354, 6312 (2016), 603–606.

[16] Asep Juarna. 2017. Combinatorial Algorithms for Portfolio Optimization Problems – Case of Risk Moderate Investor. *Journal of Physics: Conference Series* 820, 1 (2017), 1–6.

[17] Douglas B Kell. 2012. Scientific discovery as a combinatorial optimisation problem: how best to navigate the landscape of possible experiments? *BioEssays : news and reviews in molecular, cellular and developmental biology* 34, 3 (2012), 236–244.

[18] Andrew Lucas. 2014. Ising formulations of many NP problems. *Frontiers in Physics* 2 (2014), 1–15.

[19] Catherine C. McGeoch, Richard Harris, Steven P. Reinhardt, and Paul I. Bunyk. 2019. Practical Annealing-Based Quantum Computing. *Computer* 52, 6 (2019), 38–46.

[20] Gili Rosenberg, Mohammad Vazifeh, Brad Woods, and Eldad Haber. 2016. Building an iterative heuristic solver for a quantum annealer. *Computational Optimization and Applications* 65, 3 (2016), 845–869.

[21] Masataka Sao, Hiroyuki Watanabe, Yuuichi Musha, and Akihiro Utsunomiya. 2019. Application of Digital Annealer for Faster Combinatorial Optimization. *FUJITSU SCIENTIFIC & TECHNICAL JOURNAL* 55, 2 (2019), 45–51.

[22] Tobias Stollenwerk, Elisabeth Lobe, and Martin Jung. 2019. Flight Gate Assignment with a Quantum Annealer. In *Proceedings of First International Workshop on Quantum Technology and Optimization Problems (QTOP'19)*, Vol. 11413 LNCS. 99–110.

[23] Yang Wang, Zhipeng Lü, Fred Glover, and Jin-Kao Hao. 2012. Path relinking for unconstrained binary quadratic programming. *European Journal of Operational Research* 223, 3 (2012), 595–604.

[24] Masanao Yamaoka, Chihiro Yoshimura, Masato Hayashi, Takuya Okuyama, Hidetaka Aoki, and Hiroyuki Mizuno. 2016. A 20k-Spin Ising Chip to Solve Combinatorial Optimization Problems With CMOS Annealing. *IEEE Journal of Solid-State Circuits* 51, 1 (2016), 303–309.

[25] Sheir Yarkoni, Hao Wang, Aske Plaat, and Thomas Bäck. 2019. Boosting Quantum Annealing Performance Using Evolution Strategies for Annealing Offsets Tuning. In *Proceedings of First International Workshop on Quantum Technology and Optimization Problems (QTOP'19)*, Sebastian Feld and Claudia Linnhoff-Popien (Eds.). 157–168.

[26] Andy B Yoo, Morris A Jette, and Mark Grondona. 2003. SLURM: Simple Linux Utility for Resource Management. In *Job Scheduling Strategies for Parallel Processing*, Dror Feitelson, Larry Rudolph, and Uwe Schwiegelshohn (Eds.). 44–60.