SATO Shigeyuki (佐藤 重幸)

Address (office)	Room 101D1 (10F), Faculty of Engineering Bldg. 2, 7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan (ZIP: 113-8656)
Tel. (office)	03-5841-6762
Email	sato.shigeyuki@mi.u-tokyo.ac.jp

Work Experience

The University of Tokyo 2018.12–Present

Assistant Professor, Graduate School of Information Science and Technology (Mathematics and Informatics Center)

Kochi University of Technology 2016.4–2018.11

Research Associate (Postdoc), School of Information

The University of Tokyo 2015.4–2016.3

Project Researcher, Graduate School of Information Science and Technology

Education

University of Electro-Communications 2011.4–2015.3

Ph.D. (Engineering), Department of Communication Engineering and Informatics, Graduate School of Informatics and Engineering

University of Electro-Communications 2009.4–2011.3

M.E., Department of Computer Science, Graduate School of Electro-Communications

University of Electro-Communications 2005.4–2009.3

B.E., Department of Computer Science, School of Electro-Communications

Skills

- Languages: Japanese (native), English (proficient).
- Academic expertise in programming languages and programming: especially, compilers and parallel programming.
- Principal investigation: I have been conducting 4 projects granted by JSPS Kakenhi and JST ACT-I as a PI.
- Research guidance: I have been co-advising many students of different attributes (e..g, bachelor, master, doctor, and foreign ones).
- Teaching in programming courses: I have been co-organizing a large-scale all-campus course on introductory Python programming (about 6 to 7 hundred annual participants) and leading establishing materials for it; I have solely designed and organized another advanced online course on Python programming.
- Software development and solution service for teaching: a developed online system has been running for one thousand and hundreds of student users annually in different courses.

Academic Services

IPSJ PRO Steering Committee and Editorial Board for Transactions on Programming 2023.4–Present

JSSST PPL Workshop

Program Committee	2016, 2019, 2021, and 2022
Organizing Committee	2017, 2018, and 2021

ACM SIGPLAN PPoPP 2020 Artifact Evaluation Committee

Grants

Research on Managed Languages and Runtime Systems to Utilize Memory with Computational Capabilities 2022.4–2027.3

Grant type	JSPS Kakenhi Kiban (B)
Position	Co-Investigator
Direct cost	800,000 JPY (tentative)
Reference	`https://kaken.nii.ac.jp/en/grant/KAKENHI-PROJECT-22H03566/`

Program synthesis for Processing-in-Memory architectures 2022.4–2026.3

Grant type	JSPS Kakenhi Wakate
Position	Principal investigator
Direct cost	3,500,000 JPY
Reference	`https://kaken.nii.ac.jp/en/grant/KAKENHI-PROJECT-22K17872/`

Advanced Loop Parallelization and Integrated Vectorization 2018.4–2022.4

Grant type	JSPS Kakenhi Wakate
Position	Principal investigator
Direct cost	3,200,000 JPY
Reference	`https://kaken.nii.ac.jp/en/grant/KAKENHI-PROJECT-18K18032/`

自動チューニング可能な一般化 N 体問題解法枠組みの開発 (Development of an auto-tunable generalized N-body problem solving framework) 2019.4–2021.3

Grant type	JST ACT-I (Kasoku phase)
Position	Principal investigator
Direct cost	22,000,000 JPY
Reference	`https://projectdb.jst.go.jp/grant/JST-PROJECT-19189186/`

自動チューニング可能な一般化 N 体問題解法枠組みの開発 (Development of an auto-tunable generalized N-body problem solving framework) 2017.10–2019.3

Grant type	JST ACT-I
Position	Principal investigator
Direct cost	3,000,000 JPY
Reference	`https://projectdb.jst.go.jp/grant/JST-PROJECT-17940532/`

Research

I have been doing research in programming languages and programming, particularly focusing on design and implementation. A series of my work falls roughly into the following three categories.

Algebraic Approaches to Parallelization and Optimization

I have dealt with algebraic approaches to program synthesis for parallelization and optimization as my primary expertise. The work [21] dealt with parallelizing complicated reduction loops. It developed techniques to extract matrix multiplication over a semiring automatically from a loop body through symbolic execution based on SMT solvers. The later work [5] developed techniques to extract linear polynomials over a semiring from loop nests through reverse engineering based on testing, which is more robust and lightweight than the prior by trading soundness. Semiring-based approaches to reduction parallelization were also applied to tree reduction [20], program analysis [19] and big data processing [12]. The recent work [26] has also a technique to extract the symmetry of functions and a whole-function optimization based on the symmetry.

Implementation of Language Systems

I have dealt with designing and implementing a broad scope of language systems: libraries for concurrency control [3, 16], a library for developing embedded languages [18], a tool for blaming and debugging [13], runtime systems for remote memory access [2, 7, 8], and memory management for Haskell [17]. Most recently, checkpointing techniques for non-volatile memory [1] and data science programming [23] have been studied.

Data-Parallel Programming

I have dealt with the design and implementation of parallel programming on various data structures and data models: arrays on GPUs [22], distributed trees [15], XML databases [10, 25], distributed graphs [11, 14], distributed hypergraphs [4], large-scale text data [6, 9], and space-partitioning trees [24]. All these efforts were based on data-parallel patterns, which enable programmers to write data-parallel programs being oblivious to the details of data and scheduling.

Publications (Refereed)

[1] Nakata, T., Sato, S., and Ugawa, T. To appear. General-purpose Asynchronous Periodic Checkpointing in Hybrid Memory. In Proc. the 52th International Conference on Parallel Processing (ICPP 2023).

[2] Hideshima, T., Sato, S., and Taura, K. 2022. Cost-aware Programming on Page-based Distributed Shared Memory. J. Inf. Process., 30:463–475.

[3] Endo, W., Sato, S., and Taura, K. 2022. ComposableThreads: Rethinking User-level Threads with Composability and Parametricity in C++. J. Inf. Process., 30:269–282. (Specially Selected Paper)

[4] Fujimura, S., Sato, S., and Taura, K. 2021. An Efficient and Scalable Distributed Hypergraph Processing System. J. Inf. Process., 29:812–822.

[5] Morihata, A. and Sato, S. 2021. Reverse Engineering for Reduction Parallelization via Semiring Polynomials. In Proc. the 42nd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2021), pp.820–834.

[6] Li, L., Sato, S., Liu, Q., and Taura, K. 2021. Plex: Scaling Parallel Lexing with Backtrack-Free Prescanning. In Proc. the 35th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2021), pp.693–702.

[7] Fukuoka, T., Sato, S., and Taura, K. 2021. Pitfalls of InfiniBand with On-Demand Paging. In Proc. 2021 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS 2021), pp.265–275.

[8] Endo, W., Sato, S., and Taura, K. 2020. MENPS: A Decentralized Distributed Shared Memory Exploiting RDMA. In Proc. the 4th IEEE/ACM Annual Workshop on Emerging Parallel and Distributed Runtime Systems and Middleware (IPDRM 2020), pp.9–16.

[9] Sato, S., Ihara, H., and Taura, K. 2020. CENTAURUS: A Dynamic Parser Generator for Parallel Ad Hoc Data Extraction. J. Inf. Process., 28:724–732.

[10] Sato, S., Hao, W., Matsuzaki, K. 2018. Parallelization of XPath Queries using Modern XQuery Processors. In Proc. the 22nd European Conference on Advances in Databases and Information Systems (ADBIS 2018), Short Paper, pp.54–62.

[11] Sato, S. 2018. On Implementing the Push-Relabel Algorithm on top of Pregel. New Gener. Comput., 36(4):419–449.

[12] Miyazaki, R., Matsuzaki, K., and Sato, S. 2017. A Generator of Hadoop MapReduce Programs that Manipulate One-dimensional Arrays. J. Inf. Process., 25:841–851.

[13] Arai, R., Sato, S., and Iwasaki, H. 2016. A Debugger-Cooperative Higher-Order Contract System in Python. In Proc. the 14th Asian Symposium on Programming Languages and Systems (APLAS 2016), pp.148–168.

[14] Coll Ruiz, O., Matsuzaki, K., and Sato, S. 2016. s6raph: Vertex-Centric Graph Processing Framework with Functional Interface. In Proc. Proceedings of the 5th International Workshop on Functional High-Performance Computing (FHPC 2016), pp.58–64.

[15] Sato, S. and Matsuzaki, K. 2016. A Generic Implementation of Tree Skeletons. Int. J. Parallel Program., 44(3):686–707.

[16] Kobayashi, T., Sato, S., and Iwasaki, H. 2015. Efficient Use of Hardware Transactional Memory for Parallel Mesh Generation. In Proc. the 44th International Conference on Parallel Processing (ICPP 2015), pp.600–609.

[17] Takano, Y., Iwasaki, H., and Sato, S. 2015. Design and Implementation of Thunk Recycling in the Glasgow Haskell Compiler. Computer Software, 32(1):253–287, in Japanese.

[18] Shioda, M., Iwasaki, H., and Sato, S. 2014. LibDSL: A Library for Developing Embedded Domain Specific Languages in D via Template Metaprogramming. In Proc. the 13th International Conference on Generative Programming: Concepts and Experiences (GPCE 2014), pp.63–72.

[19] Sato, S. and Morihata, A. 2014. Syntax-Directed Divide-and-Conquer Data-Flow Analysis. In Proc. the 12th Asian Symposium on Programming Languages and Systems (APLAS 2014), pp.392–407.

[20] Sato, S. and Matsuzaki, K. 2013. An Operator Generator for Skeletal Programming on Trees. IPSJ Trans. PRO, 6(4):38–49, in Japanese.

[21] Sato, S. and Iwasaki, H. 2011. Automatic Parallelization via Matrix Multiplication. In Proc. the 32nd ACM SIGPLAN conference on Programming Language Design and Implementation (PLDI 2011), pp.470–479.

[22] Sato, S. and Iwasaki, H. 2009. A Skeletal Parallel Framework with Fusion Optimizer for GPGPU Programming. In Proc. the 7th Asian Symposium on Programming Languages and Systems (APLAS 2009), pp.79–94.

Publications (Posters, Non-refereed)

[23] Nakamaru, T. and Sato, S. 2022. Multiverse Notebook: A Notebook Environment for Safe and Efficient Exploration. In Proc. the 2022 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity (SPLASH Companion 2022), pp.7–8. (Extended Poster Abstract)

[24] Sato, S., Iizuka, K., Yoshifuji, N., and Natsume, M. 2022. VIPP: Validation-Included Precision-Parametric N-Body Benchmark Suite. In Proc. 2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS 2022), pp.156–158. (Extended Poster Abstract)

[25] Hao, W., Matsuzaki, K., and Sato, S. 2021. A Dual-Index Based Representation for Processing XPath Queries on Very Large XML Documents. In Proc. the 10th EAI International Conference on Cloud Computing (CloudComp 2020), pp.18–30.

[26] Sato, S. 2019. A Symmetry-Based N-Body Solver Compiler. In Proc. the 2019 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity (SPLASH Companion 2019), pp.21–22. (Extended Poster Abstract)