The Journal of Open Source Software

MPI-Rockstar: a Hybrid MPI and OpenMP Parallel Implementation of the Rockstar Halo finder

Tomoyuki Tokuue[1], **Tomoaki Ishiyama**[2], **Ken Osato**[3,4,5], **Satoshi Tanaka**[6], **and Peter Behroozi**[7]

1 Department of Applied and Cognitive Informatics, Division of Mathematics and Informatics, Graduate School of Science and Engineering, Chiba University, Chiba 263-8522, Japan **2** Digital Transformation Enhancement Council, Chiba University, Chiba 263-8522, Japan **3** Center for Frontier Science, Chiba University, Chiba 263-8522, Japan **4** Department of Physics, Graduate School of Science, Chiba University, Chiba 263-8522, Japan **5** Kavli Institute for the Physics and Mathematics of the Universe (WPI), The University of Tokyo, Kashiwa, Chiba 277-8583, Japan **6** Centre for Gravitational Physics and Quantum Information, Yukawa Institute for Theoretical Physics, Kyoto University, Kyoto 606-8502, Japan **7** Department of Astronomy, University of Arizona, AZ 85721, USA

DOI: 10.21105/joss.08295

Software

- Review
- Repository
- Archive

Editor: Christoph Junghans

Reviewers:

- @yymao
- @kyleaoman

Submitted: 20 December 2024
Published: 25 April 2026

Summary

According to the concordance cosmological theory, structure formation and evolution proceeds hierarchically in the Universe. Smaller-scale dark matter structures gravitationally collapse first everywhere in the Universe, then merge into larger-scale structures. Such dense gravitationally bound structures of dark matter are called halos. Halos can host smaller halos, so-called subhalos (or substructures). Cosmological N-body simulations are vital in understanding the formation and evolution of halos and subhalos. Halo/subhalo finders are a post-processing step to identify those dense structures in the particle dataset of cosmological N-body simulations.

MPI-Rockstar is a massively parallel halo finder based on the Rockstar phase-space temporal halo finder code (Behroozi et al., 2013), which is one of the most extensively used halo finding codes. Compared to the original code, parallelized by a primitive socket communication library, we parallelized it in a hybrid way using the Message Passing Interface (MPI) and OpenMP, which is suitable for analysis on the hybrid shared- and distributed-memory environments of modern supercomputers. This implementation can easily handle the analysis of more than a trillion particles on more than 100,000 parallel processes, enabling the production of a huge dataset for the next generation of cosmological surveys.

Statement of need

Owing to the advance of supercomputing power and highly scalable parallel gravitational N-body codes (Garrison et al., 2021; Ishiyama et al., 2009, 2012; Potter et al., 2017; Wang et al., 2018), the number of particles in recent massive cosmological simulations exceeds a trillion (Ishiyama et al., 2021; Potter et al., 2017; Wang et al., 2022), posing significant challenges for halo finding. Several halo finding algorithms have been suggested (Knebe et al., 2013), and some of their implementations are publicly available (Behroozi et al., 2013; Elahi et al., 2019; Knollmann & Knebe, 2009; Springel et al., 2021). The computational performance of these implementations differs substantially, and they have not been uniformly tested yet on the large-scale hybrid shared- and distributed-memory environments of modern supercomputers. The original Rockstar is designed to run on distributed-memory environments, however, data communications between multiple processes are performed by one-to-one communications using sockets. The main scaling bottleneck in Rockstar is that, even when the number of sockets is unlimited, a single-threaded server process has to coordinate between all the worker processes. This starts to become a bottleneck around 10,000 processors. Besides, many sockets (file descriptors) are issued simultaneously in the case of analysis with many processes, complicating analysis on modern supercomputers because the number of file descriptors issued simultaneously is normally limited. MPI-Rockstar addresses these issues and is designed to run on more than 100,000 parallel processes in a hybrid way using MPI and OpenMP. As new functions to the original Rockstar code, MPI-Rockstar supports HDF5 as an output format and can output additional halo properties such as the inertia tensor. MPI-Rockstar is not intended to replace the original implementation.

Parallelization

As a process parallelization, the original Rockstar divides a simulation box by the number of parallel processes and assigns each sub-box to each process. Then, each process performs 3D Friends-of-Friends (FoF) to find overdense regions, and FoF halos across processes are linked by communicating boundary regions. Rockstar then performs the subhalo finding for each FoF halo using 6D phase space information. Data communications between multiple processes are performed by one-to-one communications using sockets.

In MPI-Rockstar, we replaced all socket communications in the original Rockstar with MPI communications, while maintaining compatibility with the analysis results. Rather than simply using MPI one-to-one communication, we changed the order of communication and computation to utilize collective communications and to run efficiently on large supercomputers. Furthermore, we parallelized MPI-Rockstar in a hybrid way, where thread parallelization is implemented within each process using OpenMP. The subhalo finding is parallelized not only on a process level but also on a thread level, improving the overall performance of MPI-Rockstar. This hybrid parallel design also reduces the risk of per-process out-of-memory compared with a flat-MPI configuration. Figure 1 illustrates this parallelization strategy.

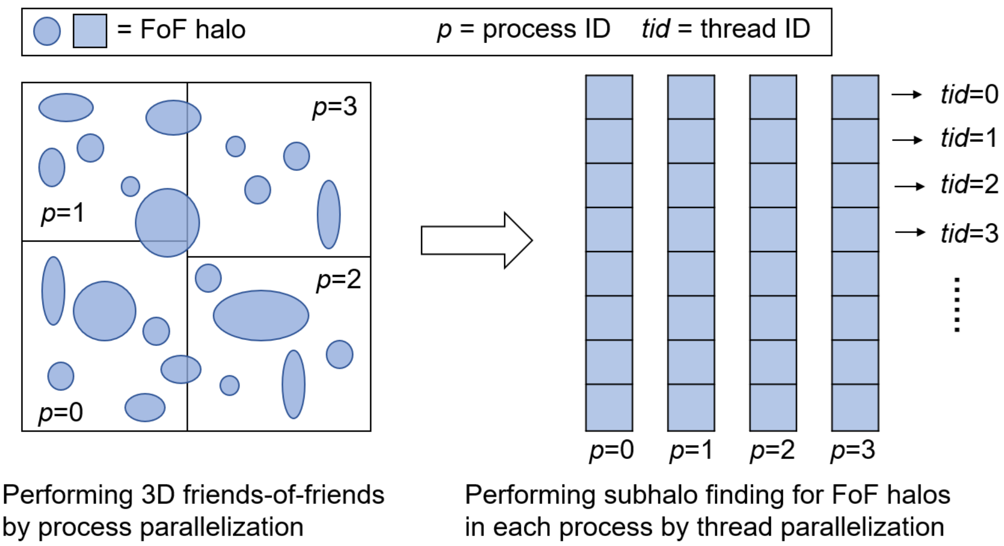

Figure 1: Parallelization strategy

Figure 2 shows a strong scaling of MPI-Rockstar using up to 1,024 nodes (48 CPU cores per node) on supercomputer Fugaku. The horizontal and vertical axes represent the number of computational nodes and the time taken for the halo and subhalo finding of one snapshot, respectively. The blue and green curves show the strong scaling for simulations with 4096^3 particles in a 2 Gpc/h box and with 2560^3 particles in a 400 Mpc/h box, respectively. The dotted curves show the ideal scalings. We analyzed a single snapshot at redshift 2.0. We measured the code's performance using 2 MPI processes per node and 24 OpenMP threads per process. This choice gives an optimal configuration for these snapshots, considering the balance between the computation and I/O time. The parallel efficiency is excellent, ~90% for both the 4096^3 box from 256 to 1024 nodes and the 2560^3 box from 256 to 1024 nodes. Thanks to the communication optimization and hybrid parallelization, MPI-Rockstar could run up to three times faster than the original Rockstar when compared in the same execution environment. For example, MPI-Rockstar took ~200 sec for a snapshot at z=0 with 1024^3 particles in a 250 Mpc/h box using 64 CPU cores (AMD EPYC 7452), while the original Rockstar took 440 sec. Both sets of halo statistics are consistent with each other. For example, the difference of halo mass function is below 0.1% except for the massive and less massive end, where the statistics are influenced by the small halo counts and resolution, respectively. Note that Rockstar is not deterministic, and so identical halo catalogs are generally not possible to obtain. We also confirm that MPI-Rockstar can analyze 2 trillion particle simulations on 16,384 nodes (786,432 CPU cores) of Fugaku.

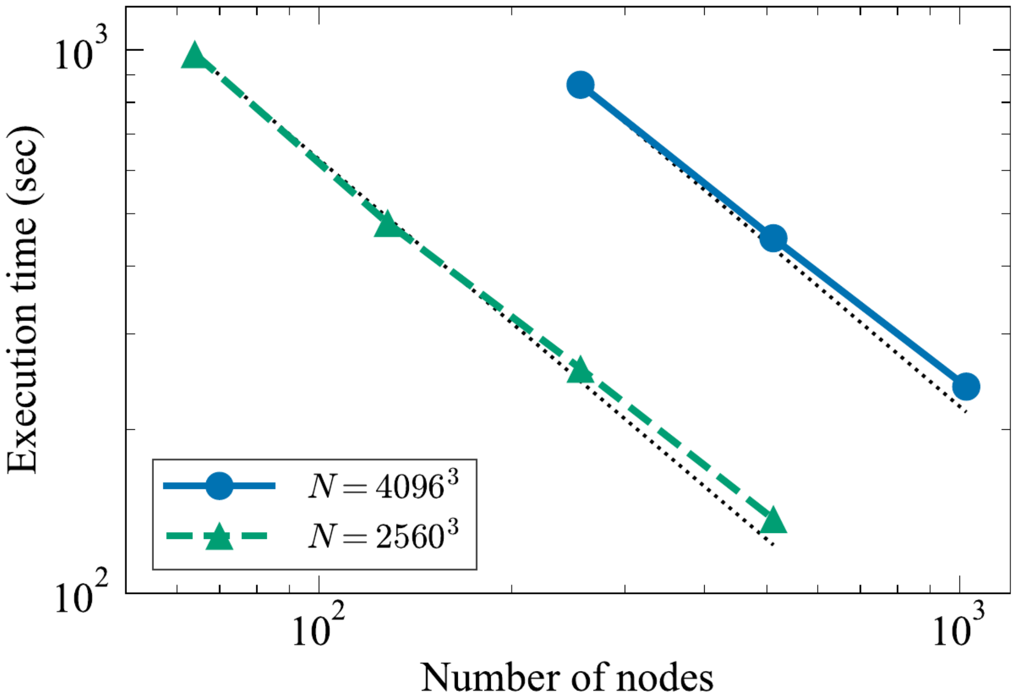

Figure 2: Strong scaling of MPI-Rockstar

Acknowledgements

This work has been supported by IAAR Research Support Program in Chiba University Japan, MEXT/JSPS KAKENHI (Grant Number JP23H04002), MEXT as "Program for Promoting Researches on the Supercomputer Fugaku'' (JPMXP1020230406 and JPMXP1020230407), and JICFuS. This work was partially supported by the Sasakawa Scientific Research Grant from The Japan Science Society. Numerical computations were carried out on the supercomputer Fugaku provided by the RIKEN Center for Computational Science (Project ID: hp230173, and hp240184).

References

Behroozi, P. S., Wechsler, R. H., & Wu, H.-Y. (2013). The ROCKSTAR phase-space temporal halo finder and the velocity offsets of cluster cores. *The Astrophysical Journal*, *762*, 109. https://doi.org/10.1088/0004-637X/762/2/109

Elahi, P. J., Cañas, R., Poulton, R. J. J., Tobar, R. J., Willis, J. S., Lagos, C. del P., Power, C., & Robotham, A. S. G. (2019). Hunting for galaxies and halos in simulations with VELOCIraptor. *Publications of the Astronomical Society of Australia*, *36*, e021. https://doi.org/10.1017/pasa.2019.12

Garrison, L. H., Eisenstein, D. J., Ferrer, D., Maksimova, N. A., & Pinto, P. A. (2021). The ABACUS cosmological N-body code. *Monthly Notices of the Royal Astronomical Society*, *508*(1), 575–596. https://doi.org/10.1093/mnras/stab2482

Ishiyama, T., Fukushige, T., & Makino, J. (2009). GreeM: Massively parallel TreePM code for large cosmological N-body simulations. *Publications of the Astrononmical Society of Japan*, *61*, 1319. https://doi.org/10.1093/pasj/61.6.1319

Ishiyama, T., Nitadori, K., & Makino, J. (2012). 4.45 Pflops astrophysical N-body simulation on K computer: The gravitational trillion-body problem. *Proc. Int. Conf. High Performance Computing, Networking, Storage and Analysis, SC'12 (Los Alamitos, CA: IEEE Computer Society Press), 5:, (arXiv:1211.4406)*. https://doi.org/10.1109/SC.2012.3

Ishiyama, T., Prada, F., Klypin, A. A., Sinha, M., Metcalf, R. B., Jullo, E., Altieri, B., Cora, S. A., Croton, D., de la Torre, S., Millán-Calero, D. E., Oogi, T., Ruedas, J., & Vega-Martínez, C. A. (2021). The Uchuu simulations: Data Release 1 and dark matter halo concentrations. *Monthly Notices of the Royal Astronomical Society*, *506*(3), 4210–4231. https://doi.org/10.1093/mnras/stab1755

Knebe, A., Pearce, F. R., Lux, H., Ascasibar, Y., Behroozi, P., Casado, J., Moran, C. C., Diemand, J., Dolag, K., Dominguez-Tenreiro, R., Elahi, P., Falck, B., Gottlöber, S., Han, J., Klypin, A., Lukić, Z., Maciejewski, M., McBride, C. K., Merchán, M. E., … Zemp, M. (2013). Structure finding in cosmological simulations: The state of affairs. *Monthly Notices of the Royal Astronomical Society*, *435*(2), 1618–1658. https://doi.org/10.1093/mnras/stt1403

Knollmann, S. R., & Knebe, A. (2009). AHF: Amiga's Halo Finder. *The Astrophysical Journal Supplement Series*, *182*(2), 608–624. https://doi.org/10.1088/0067-0049/182/2/608

Potter, D., Stadel, J., & Teyssier, R. (2017). PKDGRAV3: Beyond trillion particle cosmological simulations for the next era of galaxy surveys. *Computational Astrophysics and Cosmology*, *4*(1), 2. https://doi.org/10.1186/s40668-017-0021-1

Springel, V., Pakmor, R., Zier, O., & Reinecke, M. (2021). Simulating cosmic structure formation with the GADGET-4 code. *Monthly Notices of the Royal Astronomical Society*, *506*(2), 2871–2949. https://doi.org/10.1093/mnras/stab1855

Wang, Q., Cao, Z.-Y., Gao, L., Chi, X.-B., Meng, C., Wang, J., & Wang, L. (2018). PHoToNs-A parallel heterogeneous and threads oriented code for cosmological N-body simulation. *Research in Astronomy and Astrophysics*, *18*(6), 062. https://doi.org/10.1088/1674-4527/18/6/62

Wang, Q., Gao, L., & Meng, C. (2022). The ultramarine simulation: Properties of dark matter haloes before redshift 5.5. *Monthly Notices of the Royal Astronomical Society*, *517*(4), 6004–6012. https://doi.org/10.1093/mnras/stac3072