BIBLIOGRAPHY

[1] K. Zhereb, "Improving performance of Python code using rewriting rules technique," *Problemy Programmirovaniâ*, vol. 2–3, pp. 115–125, Aug. 2024. Available at: DOI: 10.15407/pp2020.02-03.115. [Accessed: Aug. 2024].

[2] J. Zhong, M. Hort, and F. Sarro, "Py2Cy," in *Proc. Genetic and Evolutionary Computation Conference Companion*, Aug. 2024. Available at: DOI: 10.1145/3520304.3534037. [Accessed: Aug. 2024].

[3] T. Betcke and M. W. Scroggs, "Bempp-cl: A fast Python based just-in-time compiling boundary element library," *Journal of Open Source Software*, vol. 6, no. 59, p. 2879, Aug. 2024. Available at: DOI: 10.21105/joss.02879. [Accessed: Aug. 2024].

[4] S. Jo Dixon, "Instagram - statistics & facts," *Statista*, Apr. 24, 2024. Available at: Statista.com. [Accessed: Aug. 2024].

[5] S. G. Shorokhov and V. V. Khaptakhanova, "Web based application for operational loss collection and value-at-risk and expected shortfall calculation," *ITTMM (Selected Papers)*, 2019.

[6] T. Sotiropoulos, S. Chaliasos, V. Atlidakis, D. Mitropoulos, and D. Spinellis, "Data-oriented differential testing of object-relational mapping systems," in *Proc. Int. Conf. Software Engineering*, Aug. 2024, pp. 1–11. Available at: DOI: 10.1109/ICSE43902.2021.00137. [Accessed: Aug. 2024].

[7] K.S. Duisebekova, R. Khabirov, and A. Zholzhan, "Django as a secure web framework in practice," *Vestnik KazATK*, vol. 116, no. 1, pp. 1–10, Aug. 2024. [Accessed: Aug. 2024].

[8] "Features of PyPy," *PyPy*. [Accessed: Aug. 8, 2024].

[9] Y. Izawa and H. Masuhara, "Amalgamating different JIT compilations in a meta-tracing JIT compiler framework," in *DLS 2020 - Proc. 16th ACM SIGPLAN Int. Symp. Dynamic Languages - Co-located with SPLASH 2020*, Aug. 2024, pp. 1–10. Available at: DOI: 10.1145/3426422.3426977. [Accessed: Aug. 2024].

[10] A. Roghult, "Benchmarking Python Interpreters: Measuring Performance of CPython, Cython, Jython and PyPy," Master's thesis, School of Computer Science and Communication (CSC), Royal Institute of Technology, Stockholm, 2016.

[11] K. Hule and R. Ranawat, "Analysis of different ORM tools for Data Access Object tier generation: A brief study," *International Journal of Membrane Science and Technology*, vol. 10, no. 1, pp. 1277–1291, 2023.

[12] SQLAlchemy, "The Database Toolkit for Python," Available at: SQLAlchemy.org. [Accessed: Oct. 1, 2024].

[13] Pony ORM. Available at: PonyORM.org. [Accessed: Aug. 2024].

[14] Tortoise ORM. Available at: TortoiseGitHub.io. [Accessed: Aug. 2024].

[15] FLASK. Available at: Flask.palletsprojects.com. [Accessed: Aug. 2024].

[16] FASTAPI. Available at: FastAPI.tiangolo.com. [Accessed: Aug. 2024].

[17] "Immortal Objects for Python," Meta Engineering, Aug. 15, 2023. Available at: Engineering.fb.com. [Accessed: Aug. 2024].

[18] S. Batra, "Instagram scales Python for 2 billion daily users," LinkedIn, Aug. 2024. Available at: LinkedIn.com. [Accessed: Aug. 2024].

[19] Singh, R., Bezemer, C.-P., Shang, W., & Hassan, A. E. (2016). "Optimizing the performance-related configurations of object-relational mapping frameworks using a multi-objective genetic algorithm," in *Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering*, pp. 309–320. ACM. Available at: DOI: 10.1145/2851553.2851576.

[20] "Benchmarking Performance Metrics of Python ORM Frameworks," *Peerdh Programming Insights*, [Accessed: Dec. 19, 2024].

[21] M. Ma, Z. Yang, K. Hao, L. Chen, C. Wang, and Y. Jin, "An Empirical Analysis of Just-in-Time Compilation in Modern Databases," in *Databases Theory and Applications*, Z. Bao, R. Borovica-Gajic, R. Qiu, F. Choudhury, and Z. Yang, Eds., Cham: Springer Nature Switzerland, 2024, pp. 227–240. Available at: DOI: 10.48550/arXiv.2311.04692.

[22] Nuno P. Lopes, "Torchy: A Tracing JIT Compiler for PyTorch," *Proc. of the 32nd ACM SIGPLAN International Conference on Compiler Construction (CC '23)*, Montréal, QC, Canada, pp. 1-12, Feb. 2023. DOI: 10.1145/3578360.3580266.

[23] Saumya Verma, "Efficient In-Memory Processing of SQL Queries with JIT Compilation," Master's thesis, Graduate Academic Unit of Computer Science, University of New Brunswick, Dec. 2023.

[24] Dayton J. Allen, "High Performance Python Through Workload Acceleration with OMR JitBuilder," Master's thesis, Graduate Academic Unit of Computer Science, University of New Brunswick, Jun. 2020.

[25] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: A LLVM-Based Python JIT Compiler," *Proc. of LLVM-HPC2015*, Austin, TX, USA, Nov. 2015. DOI: 10.1145/2833157.2833162.

[26] J. Ansel et al., "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation," *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '24)*, La Jolla, CA, USA, Apr. 2024. DOI: 10.1145/3620665.3640366.

[27] Q. Zhang, L. Xu, and B. Xu, "RegCPython: A Register-Based Python Interpreter for Better Performance," *ACM Transactions on Architecture and Code Optimization*, vol. 20, no. 1, pp. 14:1-14:25, Dec. 2022. DOI: 10.1145/3568973.

[28] J. Juneau et al., *The Definitive Guide to Jython: Python for the Java™Platform*, Apress, 2010. ISBN: 978-1-4302-2527-0.

[29] Q. Zhang, L. Xu, and B. Xu, "Python meets JIT compilers: A simple implementation and a comparative evaluation," *Software: Practice and Experience*, Wiley, 2024.

[30] S. Ramesh, B. N. Sukanth, and S. S. Jaswanth, "ThriveJIT: Dynamic Just-In-Time Compilation for Efficient Execution of Arithmetic Expressions," in *Proc. IEEE Conf.*, 2024.

[31] J. Nanjekye, "Memory management techniques for dynamic languages," Master's thesis, Univ. of New Brunswick, 2024.

[32] M. Ma, "Just-in-Time Compilation in Modern Databases," Master's thesis, UNSW Sydney, 2024.

[33] M. Ma, Z. Yang, K. Hao, L. Chen, C. Wang, and Y. Jin, "An empirical analysis of Just-in-Time compilation in modern databases," *arXiv preprint arXiv:2311.04692*, 2024. [Online]. Available: `https://doi.org/10.48550/arXiv.2311.04692`