DAFTAR PUSTAKA

Abhinav, P. Y., Bhat, A., Joseph, C. T., & Chandrasekaran, K. (2020). Concurrency analysis of go and java. *Proceedings of the 2020 International Conference on Computing, Communication and Security, ICCCS 2020*. https://doi.org/10.1109/ICCCS49678.2020.9277498

Akoushideh, A., & Shahbahrami, A. (2022). Performance Evaluation of Matrix-Matrix Multiplication using Parallel Programming Models on CPU Platforms. *Research Square Preprint*, 1–23.

Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. *AFIPS Conference Proceedings - 1967 Spring Joint Computer Conference, AFIPS 1967*, 483–485. https://doi.org/10.1145/1465482.1465560

Caconical. (2024). *Ubuntu 24.04 LTS (Noble Numbat) Release Notes*. Canonical Ltd. https://discourse.ubuntu.com/t/ubuntu-24-04-lts-noble-numbat-release-notes

Castro, D., Hu, R., Jongmans, S. S., Ng, N., & Yoshida, N. (2019). Distributed Programming using Role-Parametric Session Types in Go: Statically-typed endpoint APIs for dynamically-instantiated communication structures. *Proceedings of the ACM on Programming Languages*, *3*(POPL). https://doi.org/10.1145/3290342

Costanza, P., Herzeel, C., & Verachtert, W. (2019). A comparison of three programming languages for a full-fledged next-generation sequencing tool. *BMC Bioinformatics*, *20*(1), 1–10. https://doi.org/10.1186/s12859-019-2903-5

Diehl, P., Brandt, S. R., Morris, M., Gupta, N., & Kaiser, H. (2023). *Benchmarking the Parallel 1D Heat Equation Solver in Chapel, Charm++, C++, HPX, Go, Julia, Python, Rust, Swift, and Java*. *1*, 1–13. https://doi.org/10.1007/978-3-031-48803-0_11

Gross, S. (2023). *PEP 703 – Making the Global Interpreter Lock Optional in CPython*. Python Software Foundation. https://peps.python.org/pep-0703/

Gustafson, J. L. (1988). Reevaluating amdahl's law. *Communications of the ACM*, *31*(5), 532–533. https://doi.org/10.1145/42411.42415

Jung, R., Jourdan, J. H., Krebbers, R., & Dreyer, D. (2021). Safe systems programming in Rust. *Communications of the ACM*, *64*(4), 144–152. https://doi.org/10.1145/3418295

Klabnik, S., & Nichols, C. (2018). *The Rust Programming Language* (2nd ed.). No Starch Press.

Lea, D. (2000). A Java fork/join framework. *ACM 2000 Java Grande Conference*, 36–43. https://doi.org/10.1145/337449.337465

Lee, S. Y., & Aggarwal, J. K. (1987). A Mapping Strategy for Parallel Processing. *IEEE Transactions on Computers*, *C–36*(4), 433–442. https://doi.org/10.1109/TC.1987.1676925

McCool, M., Robison, A. D., & Reinders, J. (2012). Structured Parallel Programming. In *Structured Parallel Programming: Patterns for Efficient Computation*. Elsevier. https://doi.org/10.1016/C2011-0-04251-5

Menard, C., Lohstroh, M., Bateni, S., Chorlian, M., Deng, A., Donovan, P., Fournier, C., Lin, S., Suchert, F., Tanneberger, T., Kim, H., Castrillon, J., & Lee, E. A. (2023). High-performance Deterministic Concurrency Using Lingua Franca. In *ACM Transactions on Architecture and Code Optimization* (Vol. 20, Issue 4). Association for Computing Machinery. https://doi.org/10.1145/3617687

Michailidis, P. D., & Margaritis, K. G. (2012). Performance study of matrix computations using multi-core programming tools. *ACM International Conference Proceeding Series*, 186–192. https://doi.org/10.1145/2371316.2371353

Nurwina Quirante, M., Lincopinis, D., Nurwina Quirante, M. A., Sumagang, E. M., & Lincopinis, D. R. (2023). *Go Programming Language: Overview*. *May*. https://orcid.org/0000-0001-9503-8965,

Roscoe, A. W. (2005). *The Theory and Practice of Concurrency*. Prentice Hall.

Sadish, W. R., Cook, T. D., & Campbell, D. T. (2002). *Experimental and Quasi-Experimental Designs For Generalized Casual Inference*. Houghton Mifflin Company.

Sugiyono. (2020). *Metode penelitian kuantitatif, kualitatif, dan R&D*. Alfabeta.

Van der Walt, S., & Aivazis, M. (2011). The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering. *Computing in Science and Engineering*, *13*(2), 22–30. http://aip.scitation.org/doi/abs/10.1109/MCSE.2011.37

Yuan, X., & Yang, J. (2020). Effective concurrency testing for distributed systems. *International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS*, 1141–1156. https://doi.org/10.1145/3373376.3378484