















- symposium on Principles and practice of parallel programming (pp. 143-154). ACM.
- [13] Lock-Based vs. Lock-Free Concurrent Algorithms <http://mechanical-sympathy.blogspot.ru/2013/08/lock-based-vs-lock-free-concurrent.html> Retrieved: Jan, 2015
- [14] Michael, M. M. (2003). CAS-based lock-free algorithm for shared deques. In Euro-Par 2003 Parallel Processing (pp. 651-660). Springer Berlin Heidelberg.
- [15] Herlihy, M., Luchangco, V., & Moir, M. (2003, May). Obstruction-free synchronization: Double-ended queues as an example. In Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on (pp. 522-529). IEEE.
- [16] Fraser, K. (2004). Practical lock-freedom (Doctoral dissertation, University of Cambridge).
- [17] John D. Valois. Lock-Free Linked Lists Using Compare-and-Swap. In Proceedings of the 14th Annual ACM Symposium on Principles of Distributed Computing, pp. 214–222, August 1995.
- [18] Barbera, M. V., Epasto, A., Mei, A., Perta, V. C., & Stefa, J. (2013, October). Signals from the crowd: uncovering social relationships through smartphone probes. In Proceedings of the 2013 conference on Internet measurement conference (pp. 265-276). ACM.
- [19] Rajwar, R., & Goodman, J. R. (2002). Transactional lock-free execution of lock-based programs. ACM SIGOPS Operating Systems Review, 36(5), 5-17.
- [20] Herlihy, M., Luchangco, V., Moir, M., & Scherer III, W. N. (2003, July). Software transactional memory for dynamic-sized data structures. In Proceedings of the twenty-second annual symposium on Principles of distributed computing (pp. 92-101). ACM.
- [21] Michael, M. M. (2013). The balancing act of choosing nonblocking features. Communications of the ACM, 56(9), 46-53.
- [22] Namiot, D., & Sneps-Sneppe, M. (2014). On IoT Programming. International Journal of Open Information Technologies, 2(10), 25-28.
- [23] Namiot, D., & Sneps-Sneppe, M. (2014). On M2M Software. International Journal of Open Information Technologies, 2(6), 29-36.
- [24] Shavit, N. (2011). Data structures in the multicore age. Communications of the ACM, 54(3), 76-84.
- [25] Cederman, D., Gidenstam, A., Ha, P., Sundell, H., Papatriantafylou, M., & Tsigas, P. (2013). Lock-free concurrent data structures. arXiv preprint arXiv:1302.2757.
- [26] Li, X., An, H., Liu, G., Han, W., Xu, M., Zhou, W., & Li, Q. (2011, May). A Non-blocking Programming Framework for Pipeline Application on Multi-core Platform. In Parallel and Distributed Processing with Applications (ISPA), 2011 IEEE 9th International Symposium on (pp. 25-30). IEEE.
- [27] Herlihy, Maurice, and Nir Shavit. "The art of multiprocessor programming." PODC. Vol. 6. 2006.
- [28] M. Herlihy and J. Wing. "Linearizability: a Correctness Condition for Concurrent Objects." ACM Transactions on Programming Languages and Systems, 12(3): 463–492, 1990
- [29] Kaur, Ranjeet, and Pushpa Rani Suri. "Concurrent Access Algorithms for Different Data Structures: A Research Review." Global Journal of Computer Science and Technology 14.3 (2014).
- [30] Michael, M. M., & Scott, M. L. (1996, May). Simple, fast, and practical non-blocking and blocking concurrent queue algorithms. In Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing (pp. 267-275). ACM.
- [31] N. Shavit and D. Touitou. Elimination trees and the construction of pools and stacks. Theory of Computing Systems, 30:645–670, 1997
- [32] Ricci, Alessandro, Mirko Viroli, and Giulio Piancastelli. "simpA: An agent-oriented approach for programming concurrent applications on top of Java." Science of Computer Programming 76.1 (2011): 37-62.
- [33] Feo, John T., ed. A comparative study of parallel programming languages: the Salishan problems. Elsevier, 2014.
- [34] Armstrong J. Programming Erlang: software for a concurrent world. – Pragmatic Bookshelf, 2007.
- [35] Jones, S. P., Gordon, A., & Finne, S. (1996, January). Concurrent haskell. In POPL (Vol. 96, pp. 295-308).
- [36] Di Sanzo, P., Ciciani, B., et.al. (2012). On the analytical modeling of concurrency control algorithms for software transactional memories: the case of commit-time-locking. Performance evaluation 69(5), 187-205.
- [37] Dragojevic, A., Harris, T., (2012, April). STM in the small: trading generality for performance in software transactional memory. In Proceedings of the 7<sup>th</sup> ACM European Conference on Computer Systems (pp. 1-14). ACM.