http://muratbuffalo.blogspot.mx/2013/07/how-i-read-research-paper.html
http://www.eecs.harvard.edu/~michaelm/postscripts/ReadPaper.pdf

1. Gray, J., and Metz, S. (1983). Solving the problems of distributed databases. *Data Communications*, October, pp. 183-192.

2. Dean, J. and Barroso, L.A. (2013). The tail at scale. *Communications of the ACM*, Vol. 56, No. 2, pp. 74-80.

3. Diffie, W. and Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, Vol.22, No. 6, pp. 644-654.

4. Rivest, R.L., Shamir, A. and Adleman, A. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, Vol. 21, No. 2, pp. 120-126.

5. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. bitcoin.org/bitcoin.pdf

6. Cohen, B. (2003). Incentives build robustness in BitTorrent. bittorrent.org/bittorrentecon.pdf

7. Stoica, I., et al. (2001). Chord: A scalable peer-to-peer lockup protocol for Internet applications. *ACM SIGCOMM Computer Communication Review*, Vol. 31, No. 4, pp. 149-160.

8. Alpern, B. and Schneider, F. (1985). Defining liveness. *Information Processing Letters*, Vol. 21, No. 4, pp. 181-185.

9. Lamport, L. (1978). Time, clocks and the ordering of events in a distributed system. *Communications of the ACM*, Vol. 21, No. 7, pp. 558-65.

10. Ricart, G. and Agrawala, A.K. (1981). An optimal algorithm for mutual exclusion in computer networks. *Communications of the ACM*, Vol. 24, No. 1, pp. 9-17.

11. Carvalho, O. and Roucairol, G (1983). On mutual exclusion in computer networks. Technical correspondence. *Communications of the ACM*, Vol. 26, No. 2, pp. 146-147. (With authors' response).

12. Maekawa, M. (1985). A $\sqrt{N}$ algorithm for mutual exclusion in decentralized systems. *ACM Transactions on Computer System* s, Vol. 3, No. 2, pp. 145-159.

13. Chang, E.G. and Roberts, R. (1979). An improved algorithm for decentralized extrema-finding in circular configurations of processes. *Communications of the ACM*, Vol. 22, No. 5, pp. 281-283.

14. Hirschberg, D. and Sinclair, J. (1980). Decentralized extrema-finding in circular configurations of processors*. Communications of the ACM*, Vol. 23, No. 11, pp. 627-628.

15. Peterson, G. (1982). An O(nlogn) unidirectional distributed algorithm for the circular extrema problem*. ACM Transactions on Programming Languages and Systems.* Vol. 4, No. 4, pp. 758-762.

16. Garcia-Molina, H. (1982). Elections in distributed computer systems. *IEEE Transactions on Computers*, Vol. C-31, No. 1, pp. 48-59.

17. Hoare, C.A.R. (1978). Communicating sequential processes. *Communications of the ACM*, Vol. 21, No. 8, pp. 666-677.

18. Dijkstra, E. and Scholten, C. (1980). Termination detection for diffusing computations. *Information Processing Letters*, Vol. 11, No. 1, pp. 1-4.

19. Dijkstra, E., Feijen, W. and van Gasteren, A. (1983). Derivation of a termination detection algorithm for distributed computations. *Information Processing Letters*, Vol. 16, No. 5, pp. 217-219.

20. Topor, R. (1984). Termination detection for distributed computations. *Information Processing Letters*, Vol. 18, No. 1, pp. 33-36.

21. Rana, S. (1983). A distributed solution of the distributed termination problem. *Information Processing Letters*, Vol. 17, No. 1, pp. 43-46.

22. Mitchel, M. and Merrit, M. (1984). A distributed algorithm for deadlock detection and resolution. *Proceedings of the ACM Symposium on Principles of Distributed Computing*, pp. 282-284.

23. Chandy, K., Misra, J. and Haas, L. (1983). Distributed deadlock detection, *ACM Transactions on Computer Systems*, Vol.1, No.2, pp. 143-156.

24. Chandy, K. and Lamport, L. (1985). Distributed snapshots: Determining global states of distributed systems. *ACM Transactions on Computer Systems*, Vol. 3, No. 1, pp. 63-75.

25. Fidge, C. (1991). Logical time in distributed computing systems. *IEEE Computer*, Vol. 24, No. 8, pp. 28-33.

26. Mattern, F. (1989). Virtual time and global states in distributed systems. *Proceedings of the Workshop on Parallel and Distributed Algorithm*s, Amsterdam, North-Holland, pp. 215-226.

27. Schiper, A. and Sandoz, A. (1989). A new algorithm to implement causal ordering. *Distributed Algorithms – Lecture Notes in Computer Science,* Vol. 392, pp. 219–232.

28. Babaoglu, O. and Marzullo, K. (1993). Consistent global states of distributed systems: Fundamental concepts and mechanisms. *Technical Report UBLCS-93-1*, University of Bologna, 40 p.

29. Alagar, S., and Venkatesan, S. (1995). An optimal algorithm for distributed snapshots with causal message ordering. *Information Processing Letters*, Vol. 50, No. 6, pp. 311–316.

30. Dijkstra, E. (1974). Self-stabilizing systems in spite of distributed control. *Communications of the ACM*, Vol. 17, No. 11, pp. 643–644.

31. Dijkstra, E. (1986). A belated proof of self-stabilization. *Distributed Computing*, Vol. 1, No. 1, pp. 5–6.

32. Lampson, B. (1981). Atomic transactions. *Distributed systems: Architecture and Implementation – Lecture Notes in Computer Science,* Vol. 105, pp. 246–265.

33. Skeen, D. (1981). Nonblocking commit protocols. *Proceedings of the ACM SIGMOD international conference on management of data*, pp. 133–142.

34. Wensley, J., Lamport, L., Goldberg, J., Green, M., Levitt, K., Melliar-Smith, P., Shostak, R., and Weinstock, C. (1978) SIFT: Design and analysis of a fault-tolerant computer for aircraft control. *Proceedings of the IEEE*, Vol. 66, No. 10, pp.1240–1255.

35. Pease, M., Shostak, R. and Lamport, L. (1980). Reaching agreement in the presence of faults. *Journal of the ACM*, Vol. 27, No. 2, pp. 228–234.

36. Lamport, L., Shostak, R. and Pease, M. (1982). The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, Vol. 4, No. 3, pp. 382–401.

37. Fischer, M., Lynch, N. and Paterson, M. (1985). Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, Vol. 32, No. 2, pp. 374–382.

38. Ben-Or, M. (1983). Another advantage of free choice: completely asynchronous agreement protocols, *Proceedings of the Second ACM Symposium on Principles of Distributed Computing*, pp. 27–30.

39. Rabin, M. (1983). Randomized byzantine generals, *Proceedings of Twenty-Fourth IEEE Annual Symposium on Foundations of Computer Science*, pp. 403–409.

40. Dwork, C., Lynch, N. and Stockmeyer, L. (1988). Consensus in the presence of partial synchrony, *Journal of the ACM*, Vol. 35, No. 2, pp. 288–323.

41. Chandra, T. and Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, Vol. 43, No. 2, pp. 225–267.

42. Gusella, R. and Zatti, S. (1989). The accuracy of clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD. *IEEE Transactions on Software Engineering*, Vol. 15, No. 7, pp. 847–853.

43. Mills, D. (1995). Improved algorithms for synchronizing computer network clocks. *IEEE Transactions on Networking*, Vol. 3, No. 3, pp. 245–254.

44. Liskov, B. (1991). Practical uses of synchronized clocks in distributed systems, *Distributed Computing.* Vol. 6, No. 4, pp. 211–219.

45. Schneider, F.B. (1990). Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys*, Vol. 22, No. 4, pp. 299–319.

46. Lamport, L. (1998). The part-time parliament. *ACM Transactions on Computer Systems (TOCS)*, Vol. 16, No. 2, pp. 133–169.

47. Lamport, L. (2001). Paxos made simple, *ACM SIGACT News (Distributed Computing Column)*, Vol. 32, No. 4, pp. 51–58.

48. Chandra, T., Griesemer, R. and Redstone, J. (2007). Paxos made live: An engineering perspective. *Proc. of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC),* Portland, Oregon, pp. 398–407.

49. Ongaro, D. and Ousterhout, J. (2014). In Search of an Understandable Consensus Algorithm. *Proceedings of the USENIX Annual Technical Conference*, pp. 305–319.

50. Oki, B., and Liskov, B. (1988). Viewstamped replication: A new primary copy method to support highly-available distributed systems. *Proceedings of the Seventh annual ACM Symposium on Principles of Distributed Computing* (PODC)*,* pp. 8–17.

51. Castro, M. and Liskov, B. (1999). Practical byzantine fault tolerance, *Proceedings of the Third Symposium on Operating Systems Design and Implementation* (OSDI).

52. Liskov, V. (2010). From viewstamped replication to bizantine fault tolerance. *Replication – Lecture Notes in Computer Science,* Vol. 5959, Chapter 7, pp. 121–149.

53. Terry, D., Demers, A., Petersen, K., Spreitzer, M., Theimer, M., and Welch, B. (1994). Session guarantees for weakly consistent replicated data, *Proceedings of 3rd International Conference on Parallel and Distributed Information Systems* (PDIS), pp. 140–149.

54. Vogels, W. (2009). Eventually consistent, *Communications of the ACM*, Vol. 52, No. 1, pp. 40–44.

55. Terry, D. (2011). Replicated data consistency explained through baseball, *MSR Technical Report*, 14p.

56. Brewer, E. (2012). CAP twelve years later: how the "rules" have changed. *IEEE Computer*, Vol. 45, No. 2, pp. 23–29.

57. Golab, W., et al. (2014). Eventually consistent: Not what you were expecting? *Communications of the ACM*, Vol. 57, No. 3, pp. 38–44.

58. Stonebraker, M. and Cattel, R. (2011). 10 rules for scalable performance in 'simple operation' datastores. *Communications of the ACM*, Vol. 54, No. 6, pp. 72–80.

59. Terry, D., et al. (1995). Managing update conflicts in Bayou, a weakly connected replicated storage system, *Proceedings of the 15th ACM symposium on Operating Systems Principles*, pp. 172–182.

60. DeCandia, G., et al. (2007). Dynamo: Amazon's highly available key-value store, *Proceedings of the Symposium on Operating Systems Principles*, pp. 205–220.

61. Lakshman, A. and Malik, Prashant. (2010). Cassandra: a decentralized structured storage system, *ACM SIGOPS Operating Systems Review*, Vol. 44, No. 2, April, pp. 35–40.

62. Corbet, J., et al. (2013). Spanner: Google's globally-distributed database. *ACM Transactions on Computer Systems*, Vol. 31, No. 3, Article 8, 22p.

63. Brewer, E. (2017). Spanner, TrueTime & the CAP theorem, https://research.google.com/pubs/pub45855.html

64. Herlihy, M. (2019). Blockchains from a distributed computing perspective, *Communications of the ACM*, Vol. 62, No. 2, pp. 78–85.

65. Hellerstein, J. & Alvaro, P. (2020). Keeping CALM: when distributed consistency is easy, *Communications of the ACM*, Vol. 63, No. 9, pp. 72–81.