Selected Distributed Computing Literature

November 18, 2014

Here are some of the key books and papers about distributed computing that are related to the formal approach of the PRL group. Several of the PRL group articles are cited as well.

[6, 2, 3, 5, 11, 12, 8, 9, 10, 47, 46, 45]

[1, 19, 18, 25, 37, 39, 36, 50, 30, 31, 32, 34, 35, 33]

[51, 48, 49, 21, 7, 16, 44, 38]

[26, 43, 41, 42, 17, 4, 22, 23, 14, 40]

[20, 40, 52, 24, 15, 28, 29, 27, 13]

References

[1] H. Attiya and J. Welch. *Distributed Computing: Fundamentals, Simulations, and Advanced Topics.* Wiley Interscience, New York, 2nd edition, 2004.

[2] Mark Bickford. Component specification using event classes. In *Lecture Notes in Computer Science 5582*, pages 140–155. Springer, 2009.

[3] Mark Bickford, Robert Constable, and David Guaspari. Constructing event structures over a general process model. Department of Computer Science TR1813-23562, Cornell University, Ithaca, NY, 2011.

[4] Mark Bickford, Robert Constable, Joseph Y. Halpern, and Sabina Petride. Knowledge-based synthesis of distributed systems using event structures. In Franz Baader and Andreo Voronsky, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, volume 3452 of *Lecture Notes in Computer Science*, pages 449–465, 2005.

[5] Mark Bickford, Robert Constable, and Vincent Rahli. The Logic of Events, a framework to reason about distributed systems. Computing and Information Science Technical Reports http://hdl.handle.net/1813/28695, Cornell University, Ithaca, NY, 2012.

[6] Mark Bickford and Robert L. Constable. Formal foundations of computer security. In *Formal Logical Methods for System Security and Correctness*, volume 14, pages 29–52, 2008.

[7] Mark Bickford, Christoph Kreitz, Robbert van Renesse, and Xiaoming Liu. Proving hybrid protocols correct. In Richard Boulton and Paul Jackson, editors, *14th International Conference on Theorem Proving in Higher Order Logics*, volume 2152 of *Lecture Notes in Computer Science*, pages 105–120, Edinburgh, Scotland, September 2001. Springer-Verlag.

[8] Kenneth P. Birman. The process group approach to reliable distributed computing. *Communications of the ACM*, 36(12):37–53, December 1993.

[9] Kenneth P. Birman. *Building Secure and Reliable Network Applications.* Manning Publishing Company and Prentice Hall, January 1997.

[10] Kenneth P. Birman. *Reliable Distributed Systems.* Springer-Verlag, 2004.

[11] Kenneth P. Birman and Thomas A. Joseph. Exploring virtual synchrony in distributed systems. In *ACM Symp on Operating System Principles*, 1987.

[12] Kenneth P. Birman and Thomas A. Joseph. Reliable communication in the presence of failures. *ACM Transactions on Computer Systems*, 5(1):47–76, February 1987.

[13] S. Bykov, A. Geller, G. Kliot, J. Larus, Rl Pandya, and J. Thelin. Orleans: A framework for cloud computing. Technical Report MSR-TR-2010-159, Microsoft Research, 2010.

[14] Carlos Caleiro, Luca Viganò, and David A. Basin. Metareasoning about security protocols using distributed temporal logic. *Electr. Notes Theor. Comput. Sci.*, 125(1):67–89, 2005.

[15] K. M. Chandy and J. Misra. *Parallel Program Design: A Foundation.* Addison-Wesley, 1988.

[16] Robert Constable, Mark Bickford, and Robbert van Renesse. Correct-by-construction attack-tolerant systems. In *Survivability in Cyberspace Workshop*, 2010.

[17] Edgar W. Dijkstra. A constructive approach to the problem of software correctness. *BIT*, 8:174–186, 1968.

[18] Alan Fekete, Nancy Lynch, and Alex Shvartsman. Specifying and using a partitionable group communication service. In *Proceedings.* PODC, 1997.

[19] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faculty process. *JACM*, 32:374–382, 1985.

[20] Uwe Glaesser, Yuri Gurevich, and Margus Veanes. Abstract communication model for distributed systems. *IEEE Transactions on Software Engineering*, 30(7):458–472, 2004.

[21] David Gries and S. Owicki. An axiomatic proof technique for parallel programs. *Acta Informatica 6*, pages 319–340, 1976.

[22] Joseph Y. Halpern and Ronald Fagin. Modeling knowledge and action in distributed systems. *Distributed Computing*, 3(4):159–177, 1989.

[23] Joseph Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *Journal of the Association for Computing Machinery*, 37(3):549–587, 1990.

[24] D. Harel, O. Kupferman, and M. Y. Vardi. On the complexity of verifying concurrent transition systems. In *Proc. 8th Conference on Concurrency Theory*, volume 1243 of *Lecture Notes in Computer Science*, pages 258–272. Springer-Verlag, July 1997.

[25] Jason J. Hickey, Nancy Lynch, and Robbert van Renesse. Specifications and proofs for Ensemble layers. In W. Rance Cleaveland, editor, *5^{th} International Conference on Tools and Algorithms for the Construction*

and Analysis of Systems, volume 1579 of *Lecture Notes in Computer Science*, pages 119–133. Springer, 1999.

[26] C. A. R. Hoare. *Communicating Sequential Processes.* Prentice Hall, 1985.

[27] Galen C. Hunt and James R. Larus. Singularity: Rethinking the software stack. *ACM SIGOPS Operating Systems Review*, 41:37–49, 2007.

[28] David Kitchin, William R. Cook, and Jayadev Misra. A language for task orchestration and its semantic properties. pages 477–491, 2006.

[29] David Kitchin, Adrian Quark, William R. Cook, and Jayadev Misra. The Orc programming language. pages 1–25, 2009.

[30] Leslie Lamport. A New Solution of Dijkstra's Concurrent Programming Problem. *Communications of the ACM*, 17(8):453–455, 1974.

[31] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–65, 1978.

[32] Leslie Lamport. The "hoare logic" of concurrent programs. *Acta Informatica 14*, pages 21–37, 1980.

[33] Leslie Lamport. The temporal logic of actions. Systems Research Center 79, Digital Equipment Corp., Palo Alto, CA, December 1991.

[34] Leslie Lamport. The part-time parliament. *ACM Trans. Computer Systems*, 16(2):133–169, 1998.

[35] Leslie Lamport. Paxos made simple. *ACM SIGACT News*, 32(4):51–58, 2001.

[36] Leslie Lamport and P. M. Melliar-Smith. Synchronizing clocks in the presence of faults. *Journal of the Association for Computing Machinery*, 32(1):52–78, January 1985.

[37] Butler Lampson. How to build a highly available system using consensus. In *Distributed Algorithms*, LNCS 115, pages 1–17. Springer-Verlag, 1996.

[38] Xiaoming Liu, Christoph Kreitz, Robbert van Renesse, Jason J. Hickey, Mark Hayden, Kenneth Birman, and Robert Constable. Building reliable, high-performance communication systems from components. In

David Kotz and John Wilkes, editors, *17th ACM Symposium on Operating Systems Principles (SOSP'99)*, volume 33(5) of *Operating Systems Review*, pages 80–92. ACM Press, December 1999.

[39] Nancy Lynch. *Distributed Algorithms.* Morgan Kaufmann Publishers, San Mateo, CA, 1996.

[40] Simon Meier, Cas Cremers, and David A. Basin. Efficient construction of machine-checked symbolic protocol security proofs. *Journal of Computer Security*, 21(1):41–87, 2013.

[41] R. Milner. Elements of interaction. *Communications of the Association for Compututing Machinery*, 36(1):78–89, January 1993.

[42] Robin Milner. *Communication and Concurrency.* Prentice-Hall, London, 1989.

[43] Robin Milner. Calculi for interaction. *Acta Informatica*, 33(8):707–737, 1996.

[44] Nicolas Schiper, Vincent Rahli, Robbert Van Renesse, Mark Bickford, and Robert L. Constable. Developing correctly replicated databases using formal tools, 2014. Accepted to DSN 2014.

[45] Fred B. Schneider. Understanding protocols for Bysantine clock synchronization. Department of Computer Science 87–859, Cornell University, Ithaca, NY, August 1987.

[46] Fred B. Schneider. Implementing fault-tolerant services using the state-machine approach. 22, December 1990.

[47] Fred B. Schneider. *On Concurrent Programming.* Springer-Verlag, New York, 1997.

[48] Robbert van Renesse. Paxos made moderately complex. Technical report, Cornell University, 2011.

[49] Robbert van Renesse, Kenneth P. Birman, Mark Hayden, Alexey Vaysburg, and David Karr. Building adaptive systems using Ensemble. *Software: Practice and Experience*, 28(9):963–979, July 1998.

[50] Robbert van Renesse, Kenneth P. Birman, and Silvano Maffeis. Horus: A flexible group communication system. *Communications of the ACM*, 39(4):76–83, April 1996.

[51] Robbert van Renesse, Yaron Minsky, and Mark Hayden. A gossip-based failure detection service. In *Proceedings of Middleware '98*, June 1998.

[52] M. Y. Vardi. An automata-theoretic approach to fair realizability and synthesis. In P. Wolper, editor, *Computer Aided Verification, Proceedings of the 7th International Conference*, volume 939 of *Lecture Notes in Computer Science*, pages 267–292. Springer-Verlag, 1995.