

Fault Tolerance Bibliography for River

Patricia Shanahan

March 25, 2011

1 Introduction

2 Transactions

2.1 A survey on the history of transaction management: from flat to grid transactions

In “A survey on the history of transaction management: from flat to grid transactions”, Wang et. al. discuss the development of transaction systems from the “Stone Age” before the use of transactions through early 21st. century grid computing [WVKG08]. This article is useful for high level background, terminology, and citations. It does not discuss implementation, algorithms, or protocols.

3 Fault Tolerant State Machines

A fault tolerant state machine could be used as the basis for either transactions or JavaSpace. It provides a general mechanism for having multiple servers collectively carry out client requests with the ability to continue functioning after some subset has failed, and introducing or reintroducing computers to bring the system back to full strength.

3.1 From Viewstamped Replication to Byzantine Fault Tolerance

Barbar Liskov describes two implementations in “From Viewstamped Replication to Byzantine Fault Tolerance” [Lis10]. The earlier implementation, Viewstamped Replication, was designed to protect against simple failures in which a server stops responding. It was later extended to handle Byzantine failures, in which a server or its communication paths may be under the control of a malicious agent attempting to subvert or disrupt the application.

The last paragraph of the article mentions several subsequent developments with references:

- Techniques for heterogeneous replication to avoid the problem of correlated failures causing many replicas to fail simultaneously [CRL03, VBLM07].
- Study of system properties when more than f replicas fail simultaneously [LM07].
- Avoiding the use of a primary, either entirely or during normal case processing [AEMGG⁺05, CML⁺06].
- Reducing the latency of normal case processing [KAD⁺07, WCN⁺09].[12, 33]

References

- [AEMGG⁺05] Michael Abd-El-Malek, Gregory R. Ganger, Garth R. Goodson, Michael K. Reiter, and Jay J. Wylie. Fault-scalable byzantine fault-tolerant services. *SIGOPS Oper. Syst. Rev.*, 39:59–74, October 2005.
- [CML⁺06] James A. Cowling, Daniel S. Myers, Barbara Liskov, Rodrigo Rodrigues, and Liuba Shrira. Hq replication: A hybrid quorum protocol for byzantine fault tolerance. In *OSDI*, pages 177–190. USENIX Association, 2006.
- [CRL03] Miguel Castro, Rodrigo Rodrigues, and Barbara Liskov. Base: Using abstraction to improve fault tolerance. *ACM Trans. Comput. Syst.*, 21:236–269, August 2003.
- [KAD⁺07] Ramakrishna Kotla, Lorenzo Alvisi, Michael Dahlin, Allen Clement, and Edmund L. Wong. Zyzzyva: speculative byzantine fault tolerance. In Thomas C. Bressoud and M. Frans Kaashoek, editors, *SOSP*, pages 45–58. ACM, 2007.
- [Lis10] Barbara Liskov. From viewstamped replication to byzantine fault tolerance. In Bernadette Charron-Bost, Fernando Pedone, and André Schiper, editors, *Replication*, volume 5959 of *Lecture Notes in Computer Science*, pages 121–149. Springer, 2010.
- [LM07] Jinyuan Li and David Mazières. Beyond one-third faulty replicas in byzantine fault tolerant systems. In *NSDI*. USENIX, 2007.
- [VBLM07] Ben Vandiver, Hari Balakrishnan, Barbara Liskov, and Sam Madden. Tolerating byzantine faults in database systems using commit barrier scheduling. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP)*, Stevenson, Washington, USA, October 2007.

- [WCN⁺09] Benjamin Wester, James A. Cowling, Edmund B. Nightingale, Peter M. Chen, Jason Flinn, and Barbara Liskov. Tolerating latency in replicated state machines through client speculation. In Jennifer Rexford and Emin Gün Sirer, editors, *NSDI*, pages 245–260. USENIX Association, 2009.
- [WVKG08] Ting Wang, Jochem Vonk, Benedikt Kratz, and Paul Grefen. A survey on the history of transaction management: from flat to grid transactions. *Distrib. Parallel Databases*, 23:235–270, June 2008.