An Analysis of Weakly Consistent Replication Systems in an Active Distributed Network

Amit Chougule[1] **and Pravin Ghewari**[2]

[1]**Department of Computer Science, Bharati Vidyapeeth's college of Engineering, Kolhapur, Maharashhtra, India**

[2]**Department of Electronics & Telecommunication, Bharati Vidyapeeth's College of Engineering, Kolhapur, Maharashtra, India**

Abstract

With the sudden increase in heterogeneity and distribution of data in wide-area networks, more flexible, efficient and autonomous approaches for management and data distribution are needed. In recent years, the proliferation of inter-networks and distributed applications has increased the demand for geographically-distributed replicated databases. The architecture of Bayou provides features that address the needs of database storage of world-wide applications. Key is the use of weak consistency replication among autonomous machines. The protocol carries out pair wise reconciliation between the replicas. It enables replica convergence towards consistency.

The paper presents an analysis of weakly consistent replication system in an active distributed network.

Keywords: *Weakly consistent replication systems, distributed networks, Anti-Entropy protocol, Anti-Entropy time*

1. Introduction

A number of research and commercial systems have used weak consistency replication and propagated updates among replicas. Each of the individual features of Bayou's anti-entropy protocol has almost certainly appeared in previous systems in some form but the interesting differences lie in the implementation details about what information gets exchanged between replicas, what data structures are used to keep track of other replicas and the state of these replicas, what communication patterns are allowed between replicas, and so on [2-6]. Golding's anti-entropy protocol comes closest to Bayou's.

Bayou's anti-entropy protocol for update propagation between weakly consistent storage replicas is based on- pair-wise communication between the servers, the propagation of write operations, and a set of ordering of the writes and closure constraints on the propagation of the writes. It operates over diverse network topologies, including low-bandwidth links / networks. It is incremental.

Bayou can also be designed for a computing environment that includes portable machines with less than ideal network connectivity. Defining a protocol by which the resolution of update conflicts stabilizes, it includes novel methods for conflict detection, called dependency checks, and per-write conflict resolution based on client-provided merge procedures. Bayou servers can rollback the effects of previously executed writes and redo them according to a global serialization order. Furthermore, Bayou permits clients to observe the results of all writes received by a server, including tentative writes whose conflicts have not been ultimately resolved [1].

2. Experimental Setup

The experimental setup consisted of 15 bayou servers connected to each other and a client that submits random requests to random servers in the system. The measurements were taken for a conference reservation application that uses Bayou to store messages. Each experiment measures the time to run the anti-entropy protocol between the bayou servers. In all experiments, both committed and tentative writes are propagated, and each write committed inserts a new message into the database. Results were collected for two message sizes: 512 byte messages and 1024 byte messages; the message sizes include both the names and the seat numbers

2.1 Results and Discussion

The analysis and measurements show that Bayou's anti-entropy protocol performs in the following manner:

- An anti-entropy session propagates only writes unknown to the receiver, and hence performs as a linear function of the number of such writes and the available network bandwidth;
- While traversing its write-log, the sender spends only a minimal amount of time deciding which writes to propagate;
- The mass of the anti-entropy algorithm execution time is spent on the network and applying the newly received writes to the write-log and database of the receiver;
- Version vector storage requirements grow between linearly and quadratically with the number of replicas, depending on the pattern in which servers are created from others;

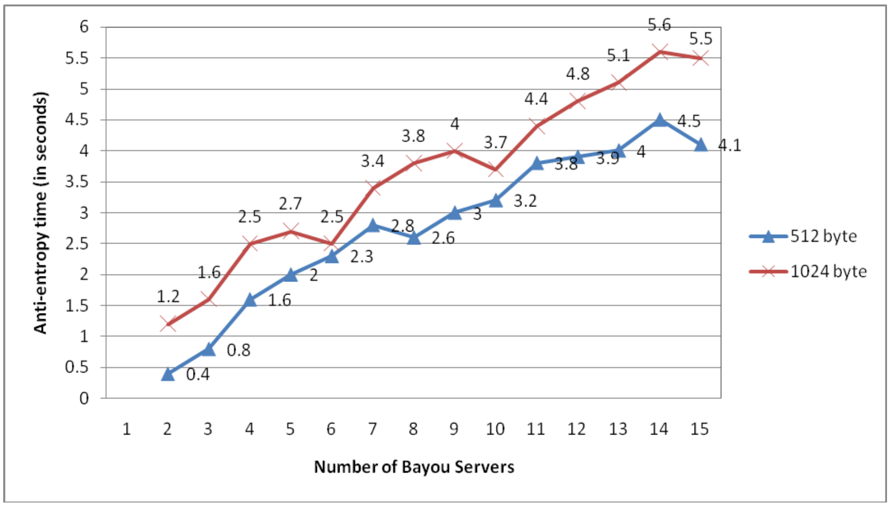

Figure 1: Variation of Anti-Entropy time with number of Bayou Servers (two fixed writes and increasing number of servers)

The data on the variation of Anti-Entropy time with two fixed writes and increasing number of Bayou servers is graphically represented in Figure 1. From the figure, it can be observed that the time taken to execute anti-entropy protocol is roughly linear. The non-linear characteristics however are due to other factors like network speed and the time at which a client submits writes to a random bayou server. Here the number of writes is fixed, a 512 byte write and a 1024 byte write and the number of servers are increased from 2 to 15. The two lines represent the time taken for 2 servers to 15 servers to do anti-entropy with each other in order to reconcile. That is, it takes 0.4 seconds time for a 512 byte write and 1.2 seconds for a 1024 byte write to become consistent on two servers and 4.1 seconds for a 512 byte write and 5.5 seconds for a 1024 byte write on 15 servers.

Anti-entropy time is also governed by the network bandwidth available for reconciliation; five experiments were conducted after a gap of 2 hours and the anti-entropy time was measured. The data on the variation of Anti-Entropy time with a single 512 byte write and fixed 7 Bayou servers is represented in Figure 2. The results are based on a reasonable assumption that network speed varies at various times during the day. This experimental setup consisted of 7 bayou servers and 1 client. Thus, anti-entropy time is affected by the network bandwidth available for reconciliation; Figure 3 is another representation of above data, to help compare the time taken for servers to reconcile in the experiments conducted.

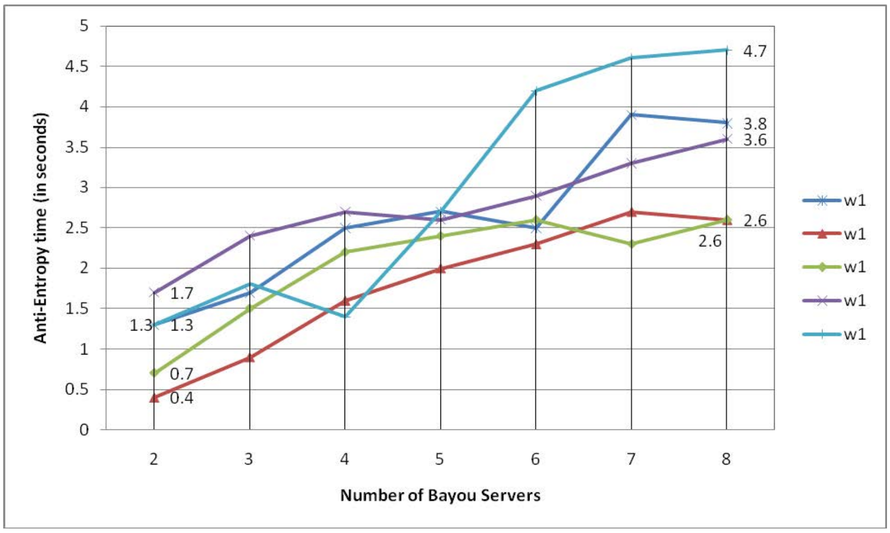

Figure 2: Variation of Anti-Entropy time with number of Bayou Servers (single write and fixed number of servers)

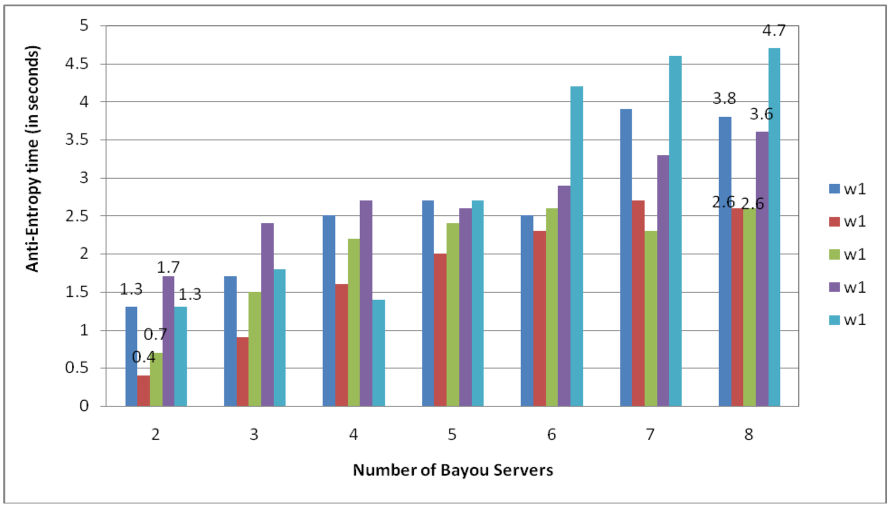

Figure 3: Clustered columns showing variation of Anti-Entropy time with number of Bayou Servers (single write and fixed number of servers)

Finally, the number of writes propagated and the anti-entropy execution time were recorded. The data on the variation of Anti-Entropy time with increasing number of writes propagated and fixed 6 bayou servers is represented in Figure 4.

It can be noticed that, it takes 2.8 seconds to propagate a single 512 byte write on six Bayou servers whereas it takes 7.5 seconds to propagate seven 512 byte writes on six Bayou servers.

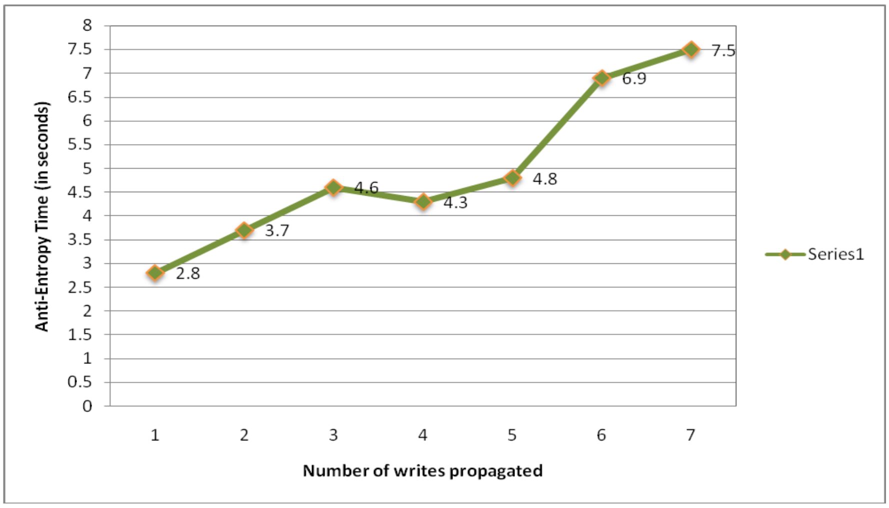

Figure 4: Variation of Anti-Entropy time with number of writes propagated (increasing writes and fixed number of servers)

To further analyze the performance of anti-entropy algorithm, the number of anti-entropy sessions and the number of servers in the system were recorded. To conduct this experiment, fixed five 512 byte writes, 7 bayou servers and a single client were used. The data on the variation of Anti-Entropy sessions with fixed five 512 byte writes and increasing number of bayou servers is graphically represented in Figure 5. The requests were submitted randomly to random servers and the number of anti-entropy sessions taken to reconcile were recorded. It should be noted that a server checking for a new write in the write log of its partner corresponds to a session so also its write propagation in case if it encounters a new write.

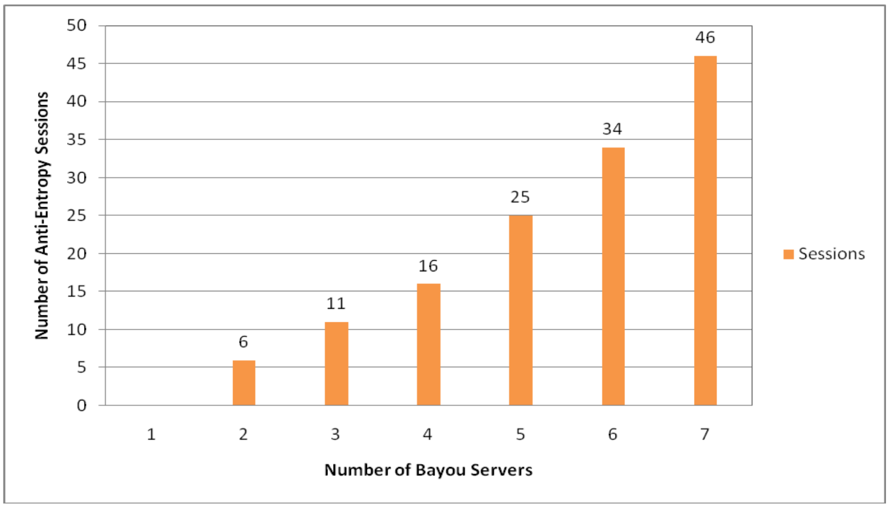

Figure 5: Variation of Anti-Entropy sessions with number of Bayou servers (fixed writes and increasing number of servers)

From Figure 5 it can be seen that, it takes six anti-entropy sessions to reconcile five 512 byte writes on 2 servers and forty six anti-entropy sessions to reconcile five 512 byte writes on seven Bayou servers.

Finally, the performance of anti-entropy algorithm when the writes were submitted from 1 to 50 and fixed number of servers from 2 to 15 is calculated. The data on the variation of Anti-Entropy time with increasing number of 512 byte writes from 1 to 50 and increasing number of bayou servers from 2 to 15 is graphically represented in Figures 6. Fourteen tests were taken, each time keeping the number of servers constant from 2 to 15. Each time 512 byte writes from 1 to 50 were submitted. From Figure 6 it can be seen that a 512 byte write w1 takes 0.2 seconds to converge on 2 servers and 4.1 seconds on 15 servers respectively. Similarly, write w50 takes 2.3 seconds to converge on 2 servers and 6.6 seconds on 15 servers respectively.

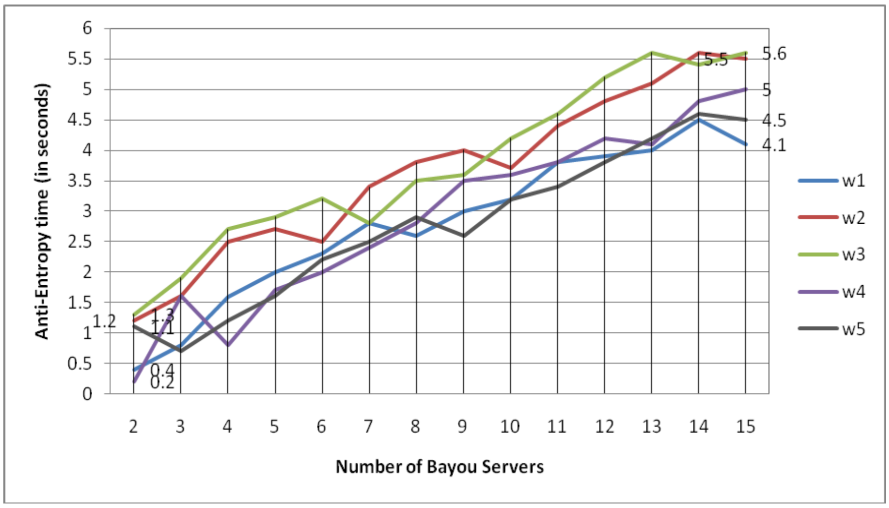

Figure 6: Variation of Anti-Entropy time with increasing number of 512 byte writes from 1 to 5 and increasing number of bayou servers from 2 to 15

3. Conclusion

The variation of anti-Entropy time with number of Bayou servers in the modes such as two fixed writes and increasing number of servers, single write and fixed number of servers, increasing writes and fixed number of servers and fixed writes and increasing number of servers has been studied. The anti-Entropy time has been found to be increasing, approximately linearly, in all the cases studied.

References

1. D. B. Terry, M. M. Theimer. K. Petersen, A. J. Demers, M. J. Spreitzer, and C. H. Hauser. Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System. *Proceedings Fifteenth ACM Symposium on Operating Systems Principles*, Copper Mountain, Colorado, December 1995, pages 172-183.
2. A. Birrell, R. Levin, R.M. Needham and M.D. Schroeder. Grapevine: An Exercise in Distributed Computing. *Communications of the ACM* 25(4): 260-274, April 1982
3. J.J. Kistler and M. Satyanarayanan. Disconnected Operation in the Coda file system. *ACM Transactions on C omputer Systems* 10(1):3-25, February 1992
4. R.G. Guy, J.S. Heidemann, W. Mak, T.W. Page, Jr., G.J. Popek and D. Rothmeier. Implementation of the Ficus Replicated File System. *Proceedings Summer USENIX Conference*, June 1990, pages 63-71.
5. Oracle Corporation. *Oracle 7 Server Distributed System: Replicated Data, Release 7.1*. Part No. A21903-2, 1995
6. J. Gray, P. Helland, P. O'neil and D. Shasha. The Dangers of Replication and a Solution. *Proceedings 1996 ACM SIGMOD Conference*, Montreal, Canada, June 1996 Pages 173-182