Understanding the Behavior of the Conflict-Rate Metric in Optimistic Peer Replication

An-I A. Wang, Peter Reiher, and Rajive Bagrodia
Computer Science Department
University of California, Los Angeles
{awang, reiher, rajive}@fmg.cs.ucla.edu

Geoffrey H. Kuenning
Computer Science Department
Harvey Mudd College
geoff@cs.hmc.edu

Abstract

Optimistic replication of data is a widely used tool for mobile environments, but the behavior of concurrent conflicting updates caused by the relaxed consistency model is poorly understood.

Through analytical modeling, we derive an exact bound for conflict rates for the common case of two replicas. The shape of the two-replica analytic curve matches well with simulation results at 50 replicas.

Our result shows that (1) both frequently and infrequently synchronized mobile machines operate in the low regions of the conflict-rate curve; (2) conflict rate is not well suited for comparing systems, since multiple system settings can result in the same conflict rate number; and (3) the conflict rate is highly dependent on the characteristics of data flow paths.

1. Introduction

Optimistic data replication is becoming a popular tool for providing a relaxed form of consistency in the presence of intermittent network connectivity, a common case in mobile computing. Optimistic replication allows multiple users to edit different replicas of the same document simultaneously with intermittent connectivity. Optimistic replication also provides the guarantee of convergence and correctness of data in the case of conflicting updates. With its resiliency to network failures, high data availability, and cooperative data sharing, optimistic replication has become an enabling technology for mobile computing. Oracle 7 [1], Bayou [14], Ingres, Lotus Notes [5], Microsoft Briefcase, and the Concurrent Version System are popular applications that have adopted the optimistic replication concept [8].

Although optimistic replication overcomes the constraint of intermittent connectivity, its relaxed consistency model allows the possibility of incurring concurrent conflicting updates—a phenomenon too little studied for a tool so widely deployed. There is empirical evidence that conflicts occur infrequently in replicated file systems [6], but a theoretical result [3] suggests that the proliferation of conflicts in databases will prevent scaling for optimistic replication. The exact behavior of conflicts in optimistic replication remains murky.

Through analytical modeling, we discovered the exact shape of the conflict rate curve for the common two-replica case. No previous analytic model for conflict rates has appeared in the literature. The curve matches well with our simulation results at 50 replicas, suggesting that the trends shown by the analytic model for two replicas are present at higher replication factors. Our major findings are that (1) both tightly and loosely synchronized computing environments operate in the low regions of the conflict rate curve; (2) conflict rate is an ambiguous metric for comparing systems, since multiple system settings can result in the same conflict rate number; and (3) the conflict rate heavily depends on the characteristics of data flow paths.

2. Background

Replication is a popular technique for providing high availability for data sharing across machine boundaries, because each machine can own a local copy of the data. ***Optimistic replication*** allows immediate access to any available replica of a data item, at the risk of permitting concurrent updates. In many scenarios, this risk is justifiable. First, most files have a single writer during any given short time period. Thus, concurrent updates are rare. Second, many applications (e.g., library database systems) can still provide meaningful service without immediate propagation of new updates. Third, for many applications, the majority of concurrent data modifications can be performed in parallel. With proper handling, the modifications can be later merged automatically or manually without data loss. Directories are an important example of this case. Independent file creations can be applied to two replicas of a directory without causing

problems, because the differing directory replicas can be easily merged into a single directory [4].

Permitting copies of data content to diverge requires a ***reconciliation process*** to bring replicas into synchronization. At some convenient time (e.g., when portable computers are temporarily connected to the network), this process compares replicas and applies the updates. Typically, reconciliation takes place between two replicas, although multi-way reconciliation is possible. Updates are tracked using either logging [13] or scanning [9]. Improper concurrent accesses, or ***conflicts***, occur when different replicas of the same file are updated after the most recent reconciliation. Optimistic systems often provide extensible application-specific libraries to resolve the majority of conflicting updates automatically [7, 10]. The remaining conflicts require user intervention.

There are three common definitions for conflicts. The first is an update that conflicts with existing updates at any replica, assuming global knowledge of all the instantaneous states of all replicas. Although this definition is simple, it does not reflect the actual effort needed to resolve conflicts (which may depend on the reconciliation model), and the global-knowledge requirement makes it impossible to use in a real system.

The second definition is oriented toward the log-based reconciliation approach. At reconciliation time, both replicas replay logs of all updates since the last reconciliation between the same replica pair. Whenever two updates to the same file on different replicas are seen in the logs, it indicates a conflict.

The third definition is related to the scanning approach, in which a reconciliation-time scan detects updates and resolves conflicts. The difference from the second definition is that multiple updates are collapsed into one and will thus result in the report of only a single conflict. Since the storage consumed by logging does not scale well with the growing number of replicas, most log-based systems also optimize out multiple updates, so that conflicts are counted according to the third definition.

For the remainder of this paper, we will use the third definition, since it is a practical approach that is consistent with the behavior of most real systems.

3. Experiments

3.1 Methodology

We developed a two-replica analytic model to gain insight into the conflict-rate curve. Base-case analysis also helps form hypotheses about large-scale conflict behaviors. For more replicas, we created a general simulation framework that can be configured to evaluate large-scale optimistically replicated file systems with heterogeneous configurations. Space limits prevent detailed discussion here, but [15] fully describes the simulation.

3.2 Experimental Assumptions

For simplicity, we assume that each machine contains a full replica of all files, and only one local user accesses each replica. We also assume that at most one reconciliation process is in progress on any given machine, and a site that is participating in a reconciliation process will deny reconciliation requests from other sites.

Each reconciliation process involves only two replicas, and the site initiating replication can choose any other replica as its partner based on a specified ***reconciliation topology***. In our case, we examined ring, star (with a single centralized node), tree (with a fanout of four), and fully connected topologies. Our reconciliation processes are unidirectional; i.e., the initiating replica pulls updates from the target replica.

Accesses to remote replicas and various node and network failures were not modeled for this study.

3.3 Parameter Space

Table 3.1 summarizes the simulation configuration and parameters. Our simulation models the reconciliation

Table 3.1: Major simulation parameters.

Parameters		*Specifications*
Environment configuration	Simulation duration	576 hours
	Replicated system	RRFS (with zero cost reconciliation)
	Physical topology	Single-level Ethernet-connected servers
System configuration	File-sharing pattern	Trace-data based
	User access skewing function	Distribution mapped from the trace data
	Number of files	10150 files with ~220 MB of data (from the trace)
	Reconciliation direction	One-way pulling
	Number of replicas	50 replicas
	Percent of file accesses to shared files	Trace-data based
	File size distribution	Trace-data based
Independent variables	Reconciliation interval	0.5 to 44 hours
	Reconciliation topology	Ring, star, tree, and fully connected topologies

algorithm of the Rumor replicated filesystem [9]. To remove the dependencies on Rumor's implementation, we only report results for a zero reconciliation delay time.

The ***reconciliation interval*** refers to the frequency of (pairwise) reconciliation. In most systems, reconciliation is performed immediately after an update [4], periodically [9], or on demand [13]. The results presented in this paper used periodic reconciliation only, with an interval varying from 0.5 to 44 hours.

4. Results

4.1 Base-Case Analytical Model

The conflict rate (the number of conflicts over time) is defined by its two implicit multiplicands—the number of conflicts per reconciliation, and the number of reconciliation processes over time. Since the number of reconciliation processes over time can be controlled directly as a replication setting, we need to derive only the probabilistic model for the number of conflicts per reconciliation to define the analytical shape of the conflict rate curve.

We start with a simple two-replica model (Figure 4.1), which is also the predominant model of replication for many applications. Our analysis was for a single file with Poisson update and reconciliation rates.

λ_1 = arrival rate for replica 1 (number of updates per unit interval)
λ_2 = arrival rate for replica 2 (number of updates per unit interval)
μ = reconciliation rate (number of reconciliations per unit interval)
$1/\mu$ = reconciliation interval
p_0 - p_3 = equilibrium probability of being in a given state
p_3 = probability of conflict with given λ and μ

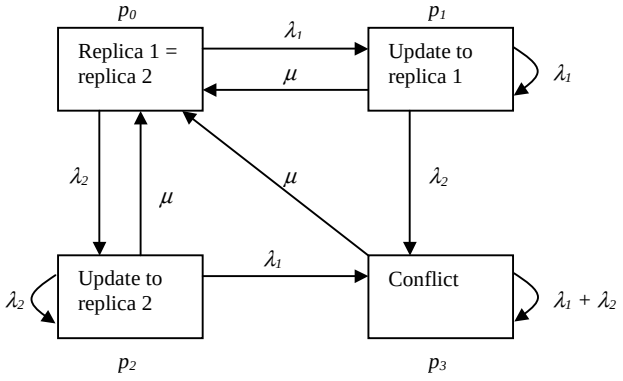

Figure 4.1: System states for two replicas.

Our original analysis used an infinite 2-dimensional Markov chain. An update to the file at replica 1 will contribute to a horizontal transition; an update to the file at replica 2 will contribute to a vertical transition. However, we found that regardless of the number of updates for a given replica, we arrive in the conflict state whenever there are one or more updates to either replica. Therefore, we reduced the infinite chain to four states.

At equilibrium, the inflow transition rate equals the outflow transition rate for each state, and the sum of all probabilities is 1. Thus, we have the following set of equations:

$$p_0 + p_1 + p_2 + p_3 = 1 \quad (1)$$
$$(\lambda_1 + \lambda_2)p_0 = \mu (p_1 + p_2 + p_3) \quad (2)$$
$$\lambda_1 p_0 = (\lambda_2 + \mu)p_1 \quad (3)$$
$$\lambda_2 p_0 = (\lambda_1 + \mu)p_2 \quad (4)$$
$$\lambda_2 p_1 + \lambda_1 p_2 = \mu p_3 \quad (5)$$

After solving these equations and multiplying the resulting equation by the reconciliation rate (μ), we obtain the following closed form (6) for the conflict rate at a given reconciliation interval $C(1/\mu)$:

$$C\left(\frac{1}{\mu}\right) = \mu\left[\left(\frac{\lambda_1}{\lambda_1+\mu}\right)\left(\frac{\lambda_2}{\lambda_1+\lambda_2+\mu}\right)+\left(\frac{\lambda_2}{\lambda_2+\mu}\right)\left(\frac{\lambda_1}{\lambda_1+\lambda_2+\mu}\right)\right] \quad (6)$$

By taking a derivative of the equation (6) with respect to μ, we can locate the maximum conflict rate at the following reconciliation interval ($1/\mu$):

$$\frac{1}{\mu} = \frac{\sqrt{2}}{\lambda_1+\lambda_2} \quad (7)$$

Since we found that the characteristic shape of the curve does not change when values of arrival rates differ, we simplify our remaining analysis by letting $\lambda = \lambda_1 = \lambda_2$, reducing (6) to the following equation:

$$C\left(\frac{1}{\mu}\right) = \mu\left(\frac{\lambda}{\lambda+\mu}\right)\left(\frac{2\lambda}{2\lambda+\mu}\right) \quad (8)$$

Clearly, the shape of the conflict-rate curve is a function of both the update arrival rate and the reconciliation rate, and this equation provides the structure to design and tune optimistically replicated systems in various ways.

Figure 4.2 shows the conflict-rate curve for $\lambda = 1$, with varying $1/\mu$. From the general shape of the curve and equation (8), we gain the following insights:

First, both extremes of the conflict-rate curve are low. If the unit for the reconciliation interval (X axis) is one hour, both tightly synchronized machines (such as servers) and mobile machines that are synchronized infrequently operate in these low regions.

Second, the conflict-rate curve shows that either shortening or lengthening the frequency of reconciliation can reduce conflict rates. By varying the reconciliation interval alone, we can create a given conflict-rate result with more than one unique setting. This conflict-rate behavior

suggests that it would be difficult to use the conflict rate to compare two systems, even with the same parameter settings.

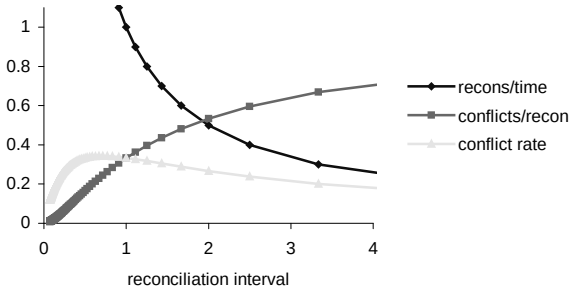

Figure 4.2: Two multiplicands of the conflict rate curve.

Third, the conflict rate is unlikely to grow at a combinatorial rate. Because the number of conflicts per reconciliation is always smaller than 1 for a given file, the family of conflict-rate curves is always smaller than the number of reconciliations over unit time. Therefore, the conflict rate for a given replica cannot exceed the reconciliation rate multiplied by the total number of shared files on a given replica—a worst-case bound for any arrival rate. Also, by changing the reconciliation interval, a system has complete control over the conflict rate and the resources dedicated to resolving conflicts.

4.2 Analytical Modeling vs. Simulation

Figure 4.3 shows the conflict rate curves obtained from simulation, for four different reconciliation topologies at 50 replicas. Except for the ring, all topologies show conflict-rate curves that deviate from our analytical curve.

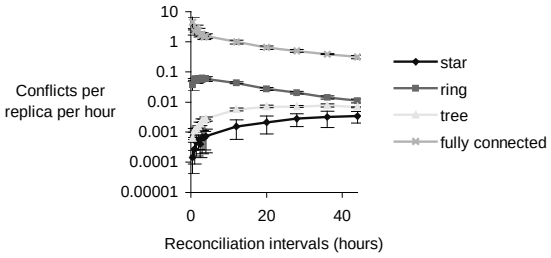

Figure 4.3: Conflict rates for 50 replicas.

We investigated this discrepancy and discovered that the process of resolving conflicts can potentially create intermediate versions that lead to subsequent conflicts, or ***metaconflicts***. This effect cannot occur for two replicas, but can for higher replication factors. Also, topologies such as the star can abort many reconciliation processes because only one reconciliation process can run at a replica at any given time. This effect alters μ directly. Also, if we see conflict resolution as a form of creating updates, the pattern of resolving conflicts can effectively change the ratio of arrival rate to reconciliation rate, or λ/μ. To explore the family of conflict-rate curves derived from our analytical result (6), we varied λ from half to twice the rate of the reconciliation rate μ..

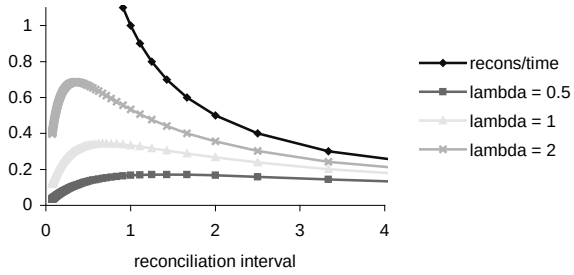

Figure 4.4: Family of conflict rate curves.

Figure 4.4 shows how our family of analytical curves fits to curves obtained from simulation. The simulated curves for the star and tree topologies resemble the analytical curve with λ = 0.5; ring, λ = 1; and fully connected, λ = 2 with the peak very close to the origin.

The simulation result indicates that the relative ordering of conflict-rate curves seems to reflect the freedom of data flow. The fully connected topology imposes the minimum constraint on where an update can propagate, while the star topology controls the conflict resolution at a central point.

We were surprised to discover that conflict-rate curves can vary by many orders of magnitude. Since the fully connected topology potentially allows 25x more reconciliation processes in parallel than the star, the fully connected topology can potentially generate conflicts 25x as fast. However, we still have two orders of magnitude difference, which is explained by the metaconflict phenomenon described in [15].

5. Related Work

Kistler and Satyanarayanan [6] have conducted an empirical study of disconnected operation in the Coda file system, showing a low likelihood of concurrent updates [6, 8]. A study of the Ficus file system [11] showed that optimistic replication used in an office environment achieved an extremely low conflict rate, especially after automatic conflict resolution was considered. Neither study considered large numbers of replicas. Also, being empirical, neither could investigate a wide range of the

parameter space. However, our results are consistent with both empirical studies.

Using an analytical approach with an assumption of uniform access patterns, Gray *et al.,* [3] suggest that the conflict rate grows at a rate that is prohibitive for optimistic replication to achieve scaling. Gray's study was based on certain assumptions about the way replication is used, which but do not hold in many environments, so their pessimistic results about the scalability of optimistic replication are frequently not applicable.

There are also many studies that examine the service quality of optimistic replication [2, 12, 16]; however, the behavior of the conflict-rate curve was not deeply explored in these studies.

6. Lessons and Recommendations

Our study shows that the analytical model of conflict rate is complex, even for the common case of two replicas. Beyond two replicas, enumerating all possible states becomes very difficult. For example, for three replicas, we need to handle corner cases where two identical replicas are in conflict with the third replica.

Our results demonstrate that the conflict rate consists of two implicit multiplicands, and both tightly or loosely synchronized environments operate at low regions of the curve.

However, it is possible to reduce conflict rates by either shortening or lengthening reconciliation intervals. Therefore, the conflict rate may not provide a usable QoS number for the purpose of direct comparison among optimistic replication schemes. Other metrics, such as staleness, have their own problems, such as difficulty of measurement in operational systems [2, 12, 15, 16]. Therefore, conflict rate should be used in combination with these other metrics.

On the other hand, the conflict rate is affected by the freedom of data flow within a given reconciliation topology. Therefore, the conflict rate may indirectly associate with other QoS metrics. It would be interesting to examine the relationship between higher conflict rates and lower staleness of data.

The conflict-rate metric is insufficiently understood, though it is still widely used to characterize the performance of replicated file systems. Other metrics used to characterize the performance of these systems have also not been rigorously examined. As a result, using popular metrics to guide the design and tuning of such systems may lead to results that do not reflect good system behavior from the user's point of view. A more critical evaluation of the metrics used to design and measure replication systems is required.

7. References

[1] Daniels D, Doo LB, Downing A, Elsbernd C, Hallmark G, Jain S, Jenkins B, Lim P, Smith G, Souder B, Stamos J. Oracle's Symmetric Replication Technology and Implications for Application Design. *Proc. of SIGMOD Conference*, p. 467, 1994.

[2] Golding RA. Modeling Replica Divergence in a Weak-Consistency Protocol for Global Scale Distribution Data Bases. Technical report UCSC-CRL-93-09, University of California, Santa Cruz, 1993.

[3] Gray J, Helland P, O'Neil P, Shasha D. The Dangers of Replication and a Solution. *Proc. of the 1996 ACM SIGMOD Conference*, pp.173-182, 1996.

[4] Guy R, Popek G, Page TW. Consistency Algorithms for Optimistic Replication. *Proc. of the 1st International Conference on Network Protocols*, *IEEE*, October 1993.

[5] Kawell LJ, Beckhardt S, Halvorsen T, Ozzie R, Greif I. Replicated Document Management in a Group Communication System. Groupware*: Software for Computer-Supported Cooperative Work*, IEEE Computer Society Press, pp. 226-235, 1992.

[6] Kistler JJ, Satyanarayanan M. Disconnected Operation in the Coda File System. *ACM Transactions on Computer Systems*, 10(1), February 1992.

[7] Kumar P, Satyanarayanan M. Flexible and Safe Resolution of File Conflicts. *Proc. of the 1995 USENIX Technical Conference*, pp. 95-106, January 1995.

[8] Kung HT, Robinson J. On Optimistic Methods for Concurrency Control. *ACM Transactions on Database Systems*, 6(2), June 1981.

[9] Page T, Guy R, Heidemann J, Ratner D, Reiher P, Goel A, Kuenning G, Popek G. Perspectives on Optimistically Replicated, Peer-to-Peer Filing. *Software—Practice and Experience*, December 1997.

[10] Ratner D, Popek GJ, Reiher P. The Ward Model: A Scalable Replication Architecture for Mobility. *Proc. of the OOPSLA'96 Workshop on Object Replication and Mobile Computing (ORMC'96)*, October 1996.

[11] Reiher P, Heidemann J, Ratner D, Skinner G, Popek G. Resolving File Conflicts in the Ficus File System. *Proc. of USENIX Conference*, pp. 183-195, June 1994.

[12] Rowstron AIT, Lawrence N, Bishop CM. Probabilistic Modeling of Replica Divergence. *Proc. of the 8th IEEE Workshop on Hot Topics in Operating Systems*, May 2001.

[13] Satyanarayanan M. Coda: A Highly Available File System for a Disconnected Workstation Environment. *Proc. of the 2nd Workshop on Workstation Operating Systems*, September 1989.

[14] Terry DB, Theimer MM, Petersen K, Demers AJ, Spreitzer MJ, Hauser CH. Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System. *Proc. of the 15th ACM Symposium on Operating Systems Principle*, December 1995.

[15] Wang AI. A Simulation Evaluation for Optimistically Replicated Environment. Master's Thesis. University of California, Los Angeles, 1998.

[16] Yu H, Vahdat A. Design and Evaluation of a Continuous Consistency Model for Replicated Servers. *Proc. of the 4th Symposium on Operating Systems Design and Implementation*, October 2000.