Reliability of Replicated Data Objects

Darrell D. E. Long [1] Jehan-François Pâris
Computer Systems Research Group
Computer Science & Engineering
University of California, San Diego

John L. Carroll
Computer Science Division
Mathematical Sciences
San Diego State University

Abstract

Improved fault tolerance of many applications can be achieved by replicating data at several sites. This data redundancy requires a protocol to maintain the consistency of the data object in the presence of site failures. The most commonly used scheme is *voting*. Voting and its variants are unaffected by network partitions. When network partitions cannot occur, better performance can be achieved with *available copy* protocols.

Common measures of dependability include *reliability*, which is the probability that a replicated object will remain constantly available over a fixed time period. We investigate the reliability of replicated data objects managed by voting, available copy and their variants. Where possible, closed-form expressions for the reliability of the various consistency protocols are derived using standard Markovian assumptions. In other cases, numerical solutions are found and validated with simulation results.

1 Introduction

In a distributed system, data are often replicated for protection against site failures and network partitions. When data are replicated at several sites, a consistency control policy must be chosen to ensure a consistent view of the data and maintain the appearance that there is only a single replica of the data. The view presented to the user must remain consistent even in the presence of site failures and network partitions. Sites recovering from a failure must present the data stored at that site in such a way that it is consistent with the global view of the data. Several consistency control protocols have been proposed, and the most promising are variants of available copy schemes and schemes based on quorum consensus.

The degree of fault tolerance achieved depends on both the degree of data redundancy and on the consistency protocol that is used to manage the object. Two common measures of dependability are *availability* and *reliability*. The *availability* $A_P(n)$ of a data object in an n-site system managed by protocol P is the steady-state probability that at an arbitrary time an access request for that data object will be granted. While the availability of a protocol measures the overall robustness of that protocol over a long period of time, its reliability estimates the probability a replicated object managed by that policy will remain constantly available over a fixed period of time. In general, the reliability $\mathcal{R}_P(n,t)$ of an n-site system managed by protocol P is defined as the probability that the system will operate correctly over a time interval of duration t given that all n units were operating correctly at time $t = 0$ [16,8].

The requirements of the application that manipulates the data object affects the relative importance of the measures. While reliability is perhaps the best indicator of the dynamic behavior of the system, if the objective is to simply minimize the inaccessibility of the data being replicated, then availability is often of primary concern. By contrast, enhanced reliability is usually the main objective for applications that incur disproportionately large costs for *any* failure of the data object.

Most studies of the degree of fault-tolerance provided by consistency protocols have been based on a comparison of data availability. While data availability is an excellent measure of the equilibrium behavior of consistency protocols, it fails to take into account the fact that most contemporary installations never reach equilibrium as network configurations often change at a rate that is less than an order of magnitude over the rate at which individual sites crash and communication subnets experience failures.

This paper employs several methods for comparing the reliability afforded by the consistency protocols discussed in the next section. Section 3 analyzes the relative reliability afforded by several schemes for varying numbers of sites using results from the theory of *k-out-of-n* systems and observations about the corresponding state-transition rate diagrams. Section 4 explores closed-form expressions for the reliability of the data objects managed by the consistency schemes for small numbers of sites. For systems in which the number of sites precludes closed-form solutions, numerical solutions, validated by simulation results, are used to establish a hierarchy of systems ordered by increasing reliability.

[1] Present addresses:
D. Long, Computer and Information Sciences, University of California, Santa Cruz, CA 95064
J.-F. Pâris, Department of Computer Science, University of Houston, Houston, TX 77004
J. Carroll, Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093

CH2713-6/89/0000/0402$01.00 © 1989 IEEE

2 Background

Voting protocols are probably the most widely studied class of consistency protocols for replicated data objects. In their simplest form, voting protocols assume that the correct state of a replicated object is the state of the majority of its replicas. Ascertaining the state of a replicated object requires collecting a *quorum* of the replicas. Should this be prevented by a sufficient number of site failures, the replicated object is considered to be inaccessible.

Majority Consensus Voting is called a *static* protocol because the required quorums of replicas and the number of votes assigned to each replica are fixed [7,5]. *Dynamic* protocols that adjusts quorums, such as dynamic voting and its variants [4,10,11,13], or modify the number of votes assigned to each replica [2], can minimize the impact of site failures and increase availability.

The *Dynamic Voting* protocol [4] instantly adjusts quorums to reflect changes in the state of the network of sites holding the replicas. Simple extensions to this scheme include *Linear-Dynamic Voting* [10] and *Hybrid Dynamic Voting* [11]. Voting protocols can successfully manage a replicated data object in the presence of network partitions. *Available copy* protocols were designed to guarantee consistency of data stored on non-partitionable computer networks, and allow higher data availabilities and reliabilities than voting protocols. Since they discount the possibility of partial communication failures, available copy protocols can allow access to a replicated object so long as a single replica of the data object remains available.

The original *Available Copy* protocol [9] assumed instantaneous detection of failures in order to find the last site to fail. Two protocols that are more realistic in terms of failure detection and are less expensive in terms of message traffic have been proposed by the authors [12]. The simpler of them is a *Naive Available Copy* protocol that does not maintain any state information. The other is an *Optimistic Available Copy* protocol that updates system state information only when a write or a recovery occurs. These protocols differ from available copy only in their behavior after a total failure, and thus have the same reliability as available copy.

3 Reliability Analysis

This section explores the effect that the protocols described in Section 2 have on reliability in systems conforming to standard Markovian assumptions. For each of these schemes, we will assume that the copies of the replicated data object reside on distinct *sites* of a computer network. Sites are subject to failures; these failures may either involve the site itself or its communication interface. When a site fails, a repair process is immediately initiated. We will also assume that the repair process will attempt to bring up to date all the copies that might have become obsolete during the time the site under repair was not operational. Such attempts will not be always successful since they depend on the availability of verifiably up-to-date copies of the replicated object.

We will assume that individual site failures are independent events distributed according to the same Poisson law with *failure rate* λ. Similarly, we will require that site repairs are exponentially distributed with *repair rate* μ. The ratio of the failure rate to the repair rate will be denoted by ρ.

The continuous accessibility of a replicated object managed by any available copy protocol is guaranteed so long as at least one of the n replicas of the object remains. Available copy, naive available copy and optimistic available copy therefore all yield the same reliability. Since no replicated object can ever operate without having at least one replica accessible, it follows that consistency protocols that do not generate new replicas to replace the ones that failed will never provide a higher reliability than available copy protocols [13].

Systems that remain operational as long as one of a set of n parallel subsystems remains operational are known as 1-*out-of-n* systems. They constitute a special case of k-*out-of-n* systems. The evaluation of their reliability requires the solution of $n+1$ differential equations [6,15]. McGregor has shown in particular [15] that the reliability of k-*out-of-n* systems with repairs can be approximated by $\mathrm{R}(n,t) \approx \exp(\frac{-t}{T_M})$, where T_M is the mean time to fail from an initial configuration where all subsystems are operational. The approximation results in negligible errors for $\rho \leq \frac{1}{5n}$ especially when $n > 3$. The reliability of such systems therefore exhibits approximately exponential decay as time increases.

The differential equations describing the behavior of systems managed by the consistency protocols can be derived from the state-transition flow rate diagrams. The states in the Markov chain are labeled to reflect the number of sites that can successfully respond to a request for the replicated object. An n-site system is in state zero if the replicated data object has been inaccessible at some point in the past, while for $1 \leq i \leq n$, the system is in state i if the object has been continuously accessible and if i replicas of the data object are currently accessible. Thus, no transitions are permitted from state zero, as we are interested only in the behavior of the system prior to the first total failure. Flow rates to adjacent states are governed by the number of sites operational and the number of sites under repair. The diagram for an n-site system employing the available copy protocol is given in Figure 1.

The state-transition flow rate diagrams for the other protocols are similar. Unlike most, linear-dynamic voting is not a birth-and-death process due to the transition in Figure 2 between the nonadjacent states 2 and 0.

Definition 1 *The* reliability $\mathcal{R}_P(n,t)$ *of an n-site system managed by protocol P is defined as the probability that the system will operate correctly over a time interval of duration t given that all n units were operating correctly at time $t = 0$. The reliability of an n-site system managed by the available copy, dynamic voting, linear-dynamic voting, and majority consensus protocols will be denoted by*

$$\mathcal{R}_A(n,t), \mathcal{R}_D(n,t), \mathcal{R}_L(n,t), \text{ and } \mathcal{R}_V(n,t),$$

respectively.

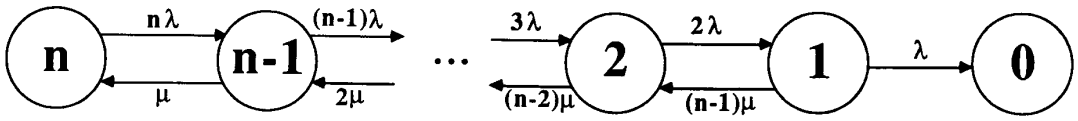

Figure 1: The $n+1$-state flow-rate diagram for available copy with n sites

While majority consensus voting is conceptually simple, it manages resources inefficiently, especially when there is an even number n of sites. In this case, the weights of the replicas can be adjusted in order to break the ties occurring when $n/2$ copies are available. The optimum weighting assignments will allow access in exactly one half of these former ties, and all such optimal strategies will provide identical reliability and availability. One such strategy assigns equal weights to $n-1$ copies, and a smaller weight to the remaining copy. It can easily be shown that this strategy performs as though the remaining copy has zero weight, that is, as though there are only $n-1$ participating sites. Thus, for each value of $k \geq 1$ and $t \geq 0$, $\mathcal{R}_V(2k,t) = \mathcal{R}_V(2k-1,t)$. This is reminiscent of the corresponding availability result, where $\mathcal{A}_V(2k) = \mathcal{A}_V(2k-1)$ [3].

For each of the other protocols, it is interesting to note that $\mathcal{R}_P(n+1,t) > \mathcal{R}_P(n,t)$ for each value of $n \geq 1$ and protocol P representing available copy, dynamic voting, and linear-dynamic voting.

Theorem 1 *For each value of $n \geq 2$ and $t \geq 0$ in systems with identical values of ρ, the reliability $\mathcal{R}_A(n,t)$ of a replicated object with n copies managed by the available copy consistency protocol is greater than the reliability $\mathcal{R}_D(n+1,t)$ of a replicated object with $n+1$ copies managed by the dynamic voting protocol which is in turn greater than the reliability $\mathcal{R}_V(2n-1,t)$ of a replicated object with $2n-1$ identical copies managed by the majority consensus voting protocol. That is,*

$$\mathcal{R}_A(n,t) > \mathcal{R}_D(n+1,t) > \mathcal{R}_V(2n-1,t).$$

Proof: The state-transition flow rate diagrams for n sites managed by available copy and $n+1$ sites managed by dynamic voting each have n states, and corresponding states are labeled with identical rates returning to higher-numbered states. Flow rates to lower-numbered states are larger for the dynamic voting diagram, and hence state zero will be reached sooner than in systems managed by available copy. The proof of the remainder of this theorem and the next similarly follow from an analysis of the state-transition flow rate diagrams.

Theorem 2 *For each value of $n \geq 2$ and $t \geq 0$ in systems with identical values of ρ, the reliability $\mathcal{R}_A(n,t)$ of a replicated object with n identical copies managed by the available copy consistency protocol is greater than the reliability $\mathcal{R}_L(n,t)$ of a replicated object with n copies managed by the linear-dynamic voting protocol which is in turn greater than the reliability $\mathcal{R}_D(n,t)$ of a replicated object with n copies managed by the dynamic voting protocol or the hybrid dynamic voting protocol. That is,*

$$\mathcal{R}_A(n,t) > \mathcal{R}_L(n,t) > \mathcal{R}_D(n,t).$$

4 Analytic Results

For small numbers of sites, closed-form solutions for the reliability of some of the protocols can be obtained from the differential-difference equations. Less tractable systems can be both simulated and solved numerically.

4.1 Closed-Form Solutions

The set of differential-difference equations arising from n sites managed by an available copy protocol is given by

$$\begin{aligned}
\frac{dp_n(t)}{dt} &= \mu p_{n-1}(t) - n\lambda p_n(t) \\
\frac{dp_j(t)}{dt} &= (j+1)\lambda p_{j+1}(t) + \\
&\quad (n+1-j)\mu p_{j-1}(t) - \\
&\quad (j\lambda + (n-j)\mu)p_j(t) \quad \text{for } 1 < j < n \\
\frac{dp_1(t)}{dt} &= 2\lambda p_2(t) - (\lambda + (n-1)\mu)p_1(t) \\
\frac{dp_0(t)}{dt} &= \lambda p_1(t) \quad \text{with initial conditions} \\
p_i(t) &= 0 \quad \text{for } 0 \leq i < n \text{ and} \\
p_n(t) &= 1
\end{aligned}$$

In this system, $p_0(t)$ represents the probability that at time t the system has failed. The reliability of the system is $\mathcal{R}_A(n,t) = 1 - p_0(t)$. When $n = 2$, the time-dependent solution to this birth-and-death process yields

$$\begin{aligned}
\mathcal{R}_A(2,t) &= \frac{(\mu+3\lambda)\sinh(t\frac{\sqrt{\mu^2+6\lambda\mu+\lambda^2}}{2})}{\sqrt{\mu^2+6\lambda\mu+\lambda^2}} e^{-t\frac{\mu+3\lambda}{2}} \\
&\quad + \cosh(t\frac{\sqrt{\mu^2+6\lambda\mu+\lambda^2}}{2})
\end{aligned}$$

A similar system of equations arises from the state-transition rate diagram associated with majority consensus voting. When $n = 3$, a closed-form solution for the corresponding $p_0(t)$ implies

$$\begin{aligned}
\mathcal{R}_V(3,t) &= \frac{(\mu+5\lambda)\sinh(t\frac{\sqrt{\mu^2+10\lambda\mu+\lambda^2}}{2})}{\sqrt{\mu^2+10\lambda\mu+\lambda^2}} e^{-t\frac{\mu+5\lambda}{2}} \\
&\quad + \cosh(t\frac{\sqrt{\mu^2+10\lambda\mu+\lambda^2}}{2})
\end{aligned}$$

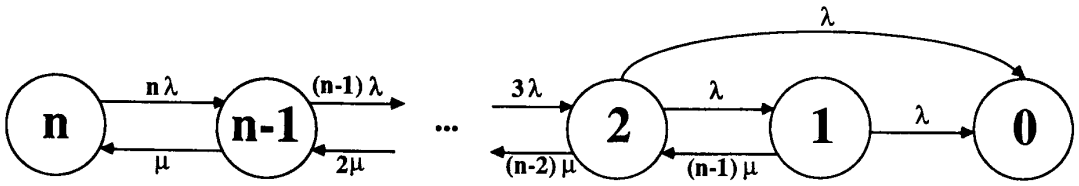

Figure 2: The $n+1$-state flow-rate diagram for linear-dynamic voting with n sites

The marked similarity of the two solutions highlight one aspect of Theorem 1, which predicts that $\mathcal{R}_A(2,t) > \mathcal{R}_V(3,t)$. The set of differential-difference equations arising from the other protocols are similar, but are intractable for meaningful values of n.

4.2 Numerical Solutions

The results in Section 3 do not impose a total ordering on the suite of systems described in this paper. Systems managed by each of the protocols can be solved numerically, affording further comparison of the reliability of each.

Figure 3 highlights the inequalities derived in Section 3 for a typical value of ρ. The time scale is measured in repair-time units: μ is taken to be 1. This figure illustrates that, as with all values of ρ, the hierarchy of reliability protocols is

$$\mathcal{R}_A(4,t) > \mathcal{R}_L(5,t) > \mathcal{R}_D(5,t) >$$

$$\mathcal{R}_A(3,t) > \mathcal{R}_L(4,t) > \mathcal{R}_D(4,t) >$$

$$\mathcal{R}_V(5,t) > \mathcal{R}_A(2,t) > \mathcal{R}_D(3,t)$$

The relative performance of the systems with respect to reliability is thus seen to match the results obtained by measuring availability [3,14]. While the hybrid linear-dynamic voting, naive available copy, and optimistic available copy variants have the same reliability as their progenitors, they do not inherit quite the same availabilities. However, for these variants and the protocols discussed in detail in this paper, it follows that for any two schemes P and S, $\mathcal{R}_P(n) > \mathcal{R}_S(n)$ implies $\mathcal{A}_P(n) > \mathcal{A}_S(n)$ for all n.

While the curves were generated from the numerical solutions of the sets of differential equations, the highlighted data points shown on the graph were obtained from simulating the repairs and failures of a system of n sites until all sites failed, and noting the time at which each of the protocols would first deny access to a replicated data object. The process was repeated 1000 times, and the results were sorted to obtain an approximation of the reliability function. The deciles of these results are shown on the graphs, and they closely agree with the differential equation solutions.

5 Conclusion

Available copy protocols have been shown to provide the highest possible reliability figures for all consistency protocols that do not incorporate new sites to replace the ones that have failed. Various relationships governing the relative reliabilities of the four main classes of consistency protocols were derived for systems with the same number of sites, and for systems with differing numbers of sites.

The numerical solutions of the Markov models, backed by simulation results, established a hierarchy of systems ordered by increasing reliability. Both dynamic and hybrid dynamic voting schemes provide marked and equal improvement over majority consensus voting, while the simple tie-breaking extension used in linear-dynamic voting comes even closer to approaching the reliability of available copy.

These conclusions need to be qualified as they rely on the hypotheses introduced in our Markovian analysis.

First, available copy protocols function correctly only when network partitions and other *partial* communication failures are impossible. This assumption is correct as long as all replicas of the data object are stored on the same CSMA/CD segment or on the same token ring. It nevertheless precludes the usage of available copy protocols in many environments where sites holding replicas are separated by gateways. Voting protocols are not subject to this limitation. In many network topologies, voting protocols suffer severe performance degradation when partitions occur. Thus, the analysis presented here represents an optimistic view of the reliability of voting protocols in environments subject to partial communication failures.

Second, unlike voting protocols, available copy protocols do not guarantee serializability of concurrent accesses. When concurrent updates can happen, available protocols need to be supplemented by a locking protocol if the network does not provide an atomic broadcast mechanism [1].

Finally, simultaneous failures of all sites holding replicas were not considered. Such failures often result from external events such as a power failure or a high-voltage transient. In any case, the consistency protocol chosen is immaterial in the wake of a total simultaneous failure, and hence the relative performances of the various schemes are unchanged.

Acknowledgements

We wish to thank Walter Burkhard, Keith Messer, Susan Hudson, Ernestine McKinney and Robin Fishbaugh for their help and encouragement. Simulation results were obtained with the aid of SIMSCRIPT, a simulation language developed and supported by CACI Products Company of La Jolla, CA. This work has been done with the aid of MACSYMA, a symbolic manipulation program developed at the MIT Laboratory for Computer Science. MACSYMA is a trademark of Symbolics, Inc.

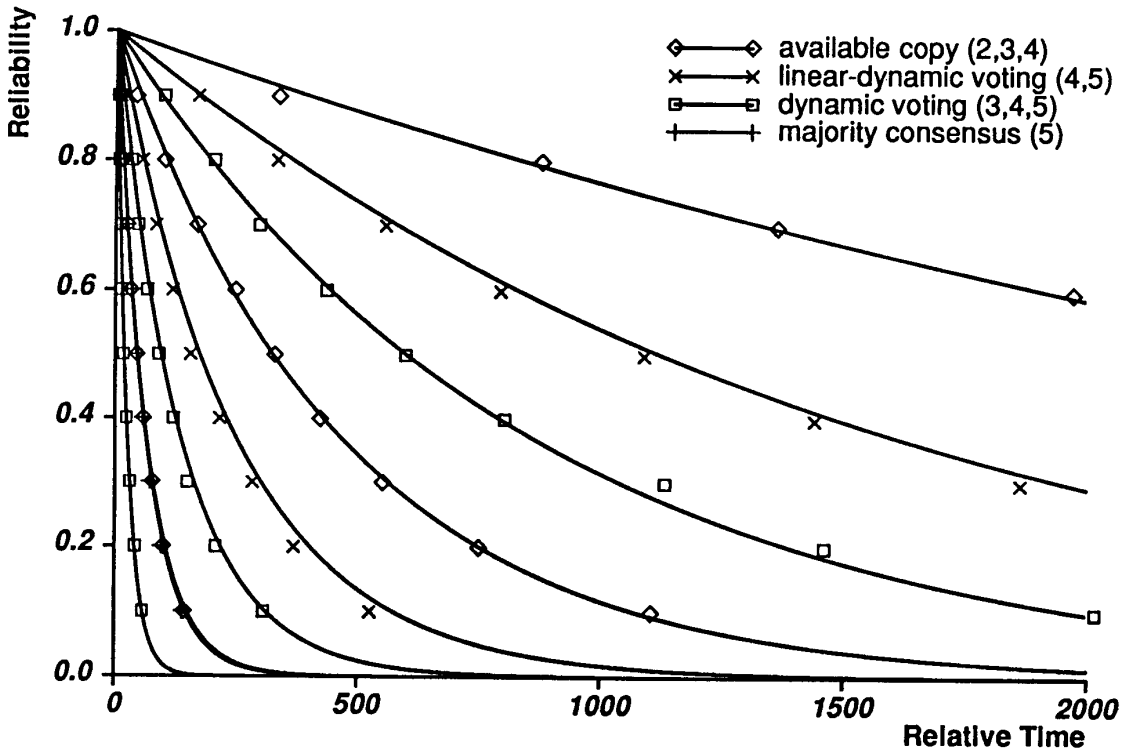

Figure 3: Comparison of protocols for $\rho = 0.1$

References

[1] Birman, K., "Replication and Fault-Tolerance in the ISIS System," *Proc. 10th ACM SOSP*, Orcas Island (1985) pp. 35–50.

[2] Bouricius, W., Carter, W., and Schneider, P., "Reliability Modeling Techniques for Self-Repairing Computer Systems," *Proc. 24th ACM Natl. Conf.*, (August 1969), pp. 295–309.

[3] Carroll, J. L., D. Long and Pâris, J.-F., "Block-Level Consistency of Replicated Files," *Proc. 7th IEEE ICDCS*, Berlin (1987) pp. 146–153.

[4] Davcev, D. and Burkhard, W.A., "Consistency and Recovery Control for Replicated Files," *Proc. 10th ACM SOSP*, (December 1985), pp. 87–96.

[5] Garcia-Molina, H., and D. Barbara, "Optimizing the Reliability Provided by Voting Mechanisms," *Proc. 4th IEEE ICDCS*, San Francisco, (May 1984), pp. 340–346.

[6] Gaver, D.P., "Stochastic Modeling: Ideas and Techniques," In: G. Louchard and G. Latouche, eds., *Probability Theory and Computer Science*. Academic Press, London, (1983).

[7] Gifford, D. K. "Weighted Voting for Replicated Data," *Proc. 7th ACM SOSP*, Pacific Grove, (Dec. 1979), pp. 150–161.

[8] Gnedenko, B. V., *Mathematical Methods in Reliability Theory*, Moscow, English Translation, New York, Academic Press, (1968).

[9] Goodman, N., D. Skeen, A. Chan, U. Dayal, R. Fox and D. Ries, "A Recovery Algorithm for a Distributed Database System," *Proc. 2nd ACM PODS*, Atlanta, (March 1983), pp. 8–15.

[10] Jajodia, S., "Managing Replicated Files in Partitioned Distributed Database Systems," *Proc. 3rd Intl. Conf. on Data Engineering*, Los Angeles (1987) pp. 412–418.

[11] Jajodia, S. and Mutchler, D., "Integrating Static and Dynamic Voting Protocols to Enhance File Availability," *Proc. 4th Intl. Conf. on Data Engineering*, Los Angeles (1988) pp. 144–153.

[12] Long, D.D.E. and Pâris, J.-F., "On Improving the Availability of Replicated Files," *Proc. 6th IEEE SRDSDS*, Williamsburg (1987) pp. 77–83.

[13] Long, D.D.E. and Pâris, J.-F., "Regeneration Protocols for Replicated Files," *Technical Report CS88-126*, CSE Department, University of California, San Diego (1988).

[14] Long, D.D.E. and Pâris, J.-F., "On the Performance of Available Copy Protocols," submitted for publication.

[15] McGregor, M.A., "Approximation Formulas for Reliability with Repair," *IEEE Trans. Reliability, R-12 (1963), pp. 64–92.*

[16] Trivedi, K., *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*, Prentice Hall, Englewood Cliffs, (1982).