CITI Technical Report 92-9

On Dependability in Distributed Databases

Toby J. Teorey
teorey@citi.umich.edu

ABSTRACT

Distributed database availability, reliability, and mean transaction completion time are derived for repairable database systems in which each component is continuously available for repair. Reliability is the probability that the entire transaction can execute properly without failure. It is computed as a function of mean time to failure (MTTF) and mean time to repair (MTTR). Tradeoffs between distributed database query and update are derived in terms of both performance and reliability.

September 1992

Center for Information Technology Integration
University of Michigan
519 West William Street
Ann Arbor, MI 48103-4943

On Dependability in Distributed Databases

Toby J. Teorey

September 1992

1. Introduction

The increasing availability and importance of distributed databases raises serious concerns about their dependability in a fragile network environment, much more than with centralized databases. Although the major impetus for distributed databases is to increase data availability, it is not always clear whether the many components of a distributed system provide the desired dependability. Performance of a database system is closely related to dependability. A significant amount of research has been reported on the subject of dependability of computer systems, and a large number of analytical models exist to predict reliability for such systems [5]. While these models provide an excellent theoretical foundation for computing dependability, there is still a need to transform the theory to practice for distributed databases running on real platforms in real distributed environments. CITI's goal is to provide that transformation with a realistic set of system parameters, test the model with a simulation tool, and make recommendations for validation testing with actual distributed database management systems.

Dependability of a system encompasses three basic characteristics: availability, reliability, and serviceability [5,10]. These terms are defined as follows:

- *Steady-state* availability is the probability that a system will be operational at any random point of time. It is expressed as the fraction of time a system is expected to be operational during the period it is required to be operational.
- *Reliability* is the conditional probability at a given confidence interval that a system will perform its intended function properly without failure and satisfy specified performance requirements during a given time interval (0,t) when used in the manner intended.
- *Serviceability*, or maintainability, is the probability of successfully performing and completing a corrective maintenance action within a prescribed period of time with the proper maintenance support.

This paper examines the issues of availability and reliability in the context of simple distributed database transactions (and their subtransactions) in a network environment where the steady-state availability is known for individual system components: computers, networks, and the various network interconnection devices (and possibly their subcomponents). A transaction path is considered to be a sequential series of resource acquisitions and executions, with alternate parallel paths allowable. All individual system components (software and hardware) are assumed to be repairable [5]. A nonrepairable distributed database is one in which transactions can be lost and the system is not available for repair. In a repairable distributed database all com-

ponents are assumed to be continuously available for repair, and an aborted transaction can restart from its point of origin.

Serviceability is assumed to be deterministic in our model, but the model could be extended for service not being completed on time.

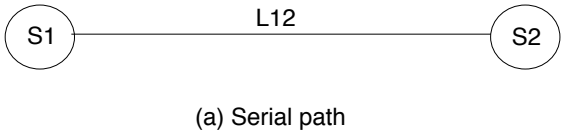
2. Availability in a Distributed Database

This section looks at the computation of steady-state availability in the network underlying the distributed database. In Figure 1a two sites, S1 and S2, are linked with the network link L12. Let A_{S1}, A_{S2}, and A_{L12} be the steady-state availabilities for components S1, S2, and L12, respectively.

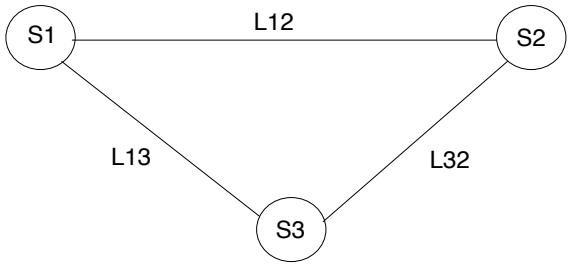
Assuming that the availability of each system component is independent of the availability of all other components, the probability that path S1/L12/S2 is available at any randomly selected time t is

$$A_{S1/L12/S2} = A_{S1} * A_{L12} * A_{S2} \quad (1)$$

which is the probability for independent events in a series.



(a) Serial path



(b) Parallel paths

Figure 1. Simple Network Paths for a Distributed Database

Extending the concept of availability to parallel paths (Figure 1b), we use the well-known relationship for parallel independent events for the availability of the network at any randomly selected time, after factoring out the two components, S1 and S2, that are common to each path:

$$A_{S1//S2} = A_{S1} * A_{S2} * (A_{L12} + A_{L13} * A_{S3} * A_{L32} - A_{L12} * A_{L13} * A_{S3} * A_{L32}) - A_{L12} * A_{L13} * A_{S3} * A_{L32} \quad (2)$$

Note that if query optimizers pick the shortest path without regard to availability, the system could reverse the decision if the selected path is not available. We now have the basic relationships for serial and parallel paths for steady-state availability. This concept can be extended for more complex paths that are needed to satisfy the transaction, due to larger networks or more complex data allocation strategies.

3. Reliability in a Repairable Distributed Database

An estimate of availability is limited to a single point in time. This section estimates the reliability for an entire transaction (including queries and/or updates) and the mean transaction completion time for a repairable distributed database that has automatic restarts (calculated in Section 4). Assume that we are given the steady-state availability of each system component, the mean time to failure (MTTF), and the mean time to repair (MTTR) for each component. We are also given the mean delay experienced by the subtransaction on each component resource, derived from known characteristics of the network and database system. From Siewiorek and Swarz's paper "The Theory and Practice of Reliable System Design" [10], we get the basic relationship for mean time between failures (MTBF):

$$MTBF = MTTF + MTTR \tag{3}$$

Reliability is the probability that the entire transaction can execute properly (over a given time interval) without failure. Compute the estimated mean reliability over a time duration (0,t), where t is the mean delay experienced over the system during the transaction execution. For tractability we assume that the failure rate of each system component has a Poisson distribution of:

$$P_j(k,t) = (mt)^k * e^{-mt} / k! \tag{4}$$

which is the probability that there are exactly k failures of transaction j in time interval t, where m is the mean number of failures in time interval t.

The probability that there are no failures in time interval t is:

$$P_j(0, t) = e^{-mt} \tag{5}$$

Transform Equation 5 using real parameters from a distributed database system. Let

$MTTF_i$ = mean time to failure on component i
$MTTR_i$ = mean time to repair for component i
MD_j = mean delay for transaction j

MD_j is estimated here without contention. In a real system with contention for resources the value of MD_j will increase dramatically as the system nears saturation, and thus the probability of failure will increase significantly. We consider contention in Section 4.

The ratio $MD_j / MTTF_i$ represents the fraction of unit failure on component i for transaction j, where mt is normalized to one, the unit (mean) time for the first failure. Substituting the ratio $MD_j/MTTF_i$ for mt in Equation 5, we obtain for any transaction j, the probability of no failures in the time interval $(0, MD_j)$ on component i. The probability that the transaction has no failures, while actively using component i, is the conditional probability that the component is reliable over the interval, given that it is available at the beginning of the interval:

$$P_{j,i}(0, MD_j) = e^{-MDj/MTTFi} * A_i \tag{6}$$

where the mean time to first failure of component i is $MTTF_i$ and the steady-state availability of a single component i is given by

$$\begin{aligned} A_i &= MTTF_i / (MTTF_i + MTTR_i) \\ &= MTTF_i / MTBF_i \end{aligned} \tag{7}$$

Example: 1 Query Reliability for a Simple Distributed Database

Apply the relationship in Equation 6 to the simple distributed database in Figure 1a. The probability that the whole transaction (query in this case) succeeds, equals the probability that the transaction can be completed without failure, beginning with the initiation at site S1, local processing the data at site S2, and returning with the result to site S1. The transaction is successful only if all components are active during the entire time of the transaction. That is,

$$P(\text{success}) = P_{j,S1}(0, MD_j) * P_{j,L12}(0, MD_j) * P_{j,S2}(0, MD_j) \quad (8)$$

Let
QIT = query initiation time (CPU)
PTT = packet transmission time
PD = packet propagation delay
QPT = query processing time (CPU & I/O)
n = number of packets in the result of the query
QRDT = query result display time (CPU & I/O)

Assume the following reasonable values for the above parameters:

QIT = 1 ms
PTT = 8 ms (T1 link @ 1.544 Mb/s, packet size 1544 bytes)
PD = 10 ms (assumed 2000 km distance, degraded electronic speed 200 km/ms)
QPT = 200 ms
n = 5
QRDT = 60 ms
$MTTF_i$ = 300 sec for each component i
$MTTR_i$ = 5 sec for each component i ($MTBF_i$ = 305 sec)
$A_i = 300/(300 + 5) = 0.984$

Define the mean total (query) delay time as the sum of all the nonoverlapped delays from the site S1, to site S2, and returning with the result to site S1. That is,

MD_j = QIT + PTT + PD + QPT + n * PTT + PD + QRDT
= 1 + 8 + 10 + 200 +5 * 8 + 10 + 60 ms
= 329 ms

Applying Equation 8 we obtain:

P(success for this query) $= [e^{-329/300000}(0.984)]^3$
$= [e^{-0.00109667}(0.984)]^3$
$= [(0.998904)(0.984)]^3$
$= [0.982922]^3$
$= 0.9496$

Example 2: Query/Update Tradeoffs

We now extend Example 1 (Figure 1b) to include updates so we can study the tradeoffs between performance and reliability for multiple copies of data. Assume that the frequency for this query is 10/tu where tu is the standard time unit (hour, day, week, etc.) for this discussion. Also assume that we have an update transaction on the same data, with time UT1 equals 329 ms multiplied by the frequency of 2/tu. The corresponding query and update times, weighed by frequency of occurrence are:

QT1 (query time for a single copy) = 329 ms * 10 = 3290 ms
UT1 (update time for a single copy) = 329 ms * 2 = 658 ms

Assume there are two copies of the data, one at S2 and another at S3, and assuming each update is initiated separately and takes the same time as the first update except with a smaller propagation delay (PD) due to the shorter distance (1200 km), we obtain the following:

QT2 (minimum query time for two copies) = 325 ms $*$ 10 = 3250 ms
UT2 (total update time for two copies) = 654 ms $*$ 2 = 1308 ms

We use the "all beneficial sites" method [1,11] to decide whether to add another copy of the data. The benefit from adding the second copy is the decrease in query time. The cost is the added cost of updating the second copy. That is,

Benefit (2nd copy) = QT1 - QT2 = 3290 - 3250 = 40 ms
Cost (2nd copy) = UT2 - UT1= 1308 - 658 = 650 ms

For this case, the cost exceeds the benefit. Therefore, the decision, based purely on performance (query and update time), is not to replicate the data. However, the query reliability should increase as the second copy is placed in site S3, and the update reliability should decrease.

In general, we see that adding more copies causes the query time to decrease and the query reliability to increase, and causes the update time to increase and the update reliability to decrease. Although the net effect tends to make adding extra copies look less favorable, deciding whether to add the copies depends on some crossover point between costs and benefits. As update frequencies become very small, the benefits dominate the costs. In the next section we will see how mean completion time is estimated.

4. Mean Transaction Completion Time

The mean transaction completion time is a function of the mean delay time, or service time, for the transaction over all components plus queuing delays, for contention, and restart delays. See Figure 2.

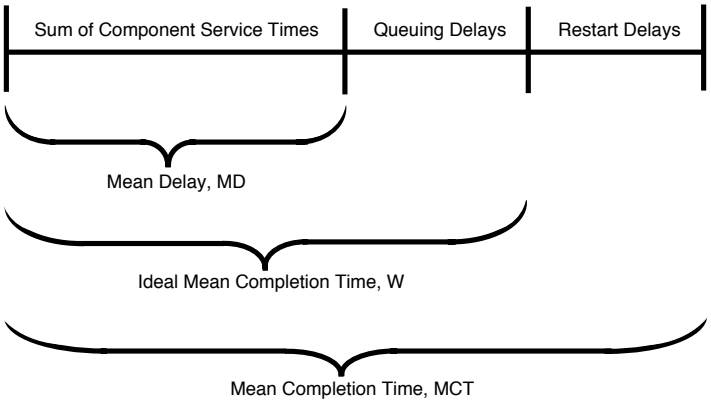


Figure 2. Components of Mean Completion Time

Queuing delays can be estimated for an open Jackson network using the formula for the mean response (completion) time:

$$W = T_S/(1 - \rho) = T_S/(1 - \lambda T_S) \quad (9)$$

where service time T_S equals MD_j, ρ is the utilization of the configuration, and λ is the mean arrival rate of queries. Therefore, W represents the ideal mean completion time when no restarts are required. In Example 1, if we assumed the arrival rate of queries was 2/sec, then

$$W = 329 \text{ ms} \; / \; (1 - 0.002 * 329) = 329/0.342 = 962 \text{ ms} = 0.962 \text{ sec}$$
$$p(\text{success}) = [e^{-962/300000} (0.984)]^3 = 0.9436$$

for the given database.

When we consider both queuing and restart delays, mean completion time is estimated by computing the probability of different completion times possible. For example, if W is the given total time in the system, the mean completion time (MCT) can be easily derived, assuming probability p of a successful transaction, and, for every failure, an average time to failure and recovery (W/2 + MTTR), where W/2 is the mean time to failure of the transaction, given random failures of a collection of components, over which that transaction must successfully execute:

$$\begin{aligned} MCT &= p * W + q * p * (W + W/2 + MTTR) + q^2 * p * (W + 2 * W/2 + 2 * MTTR) + \\ &\quad q^3 * p * (W + 3 * W/2 + 3 * MTTR) + \ldots \\ &= p * W + q * p * W + q^2 * p * W + q^3 * p * W + \ldots + q * p * (W/2 + MTTR) + \\ &\quad 2q^2 * p * (W/2 + MTTR) + 3q^3 * p * (W/2 + MTTR) + \ldots \\ &= p * W * (1 + q + q^2 + q^3 + \ldots) + q * p * (W/2 + MTTR) * (1 + 2q + 3q^2 + 4q^3 + \ldots) \\ &= W + (q/p) * (W/2 + MTTR) \end{aligned} \quad (10)$$

by noting that

$$(1 + q + q^2 + q^3 + \ldots) = 1/(1 - q) = 1/p \quad (11)$$
$$(1 + 2q + 3q^2 + 4q^3 + \ldots) = 1/(1 - q)^2 = 1/p^2 \quad (12)$$

In our example, given W = 0.962 sec, p = 0.9436, MTTR = 5 sec, and MTTF = 300 sec, we derive:

$$q = 1 - p = 0.0564$$
$$MCT = 0.962 + (0.0564/0.9436) * (0.962/2 + 5.0 \text{ sec}) = 0.962 + 0.328 = 1.290 \text{ sec}$$

Thus, the actual mean completion time need not be dramatically longer than the ideal completion time without any restarts.

5. Conclusions

We have derived expressions for availability and reliability in a repairable distributed database system for simple transactions involving both queries and updates. Reliability is derived in terms of the probability of success of an entire transaction and the mean transaction completion time due to both traffic congestion and restarts due to failure. We have also shown that simple decisions about data allocation, as an extension of the all beneficial sites algorithm, can be made from careful analysis of query and update costs.

References

1. S. Ceri and G. Pelagatti, *Distributed Databases: Principles and Systems*, McGraw-Hill, New York (1984).

2. J.B. Dugan, "On Measurement and Modeling of Computer Systems Dependability: A Dialog Among Experts," pp. 506-510 in *IEEE Transactions on Reliability Vol. 39 No. 4* (October 1990).

3. W. Feller, *An Introduction to Probability Theory and Its Applications 3rd edition*, John Wiley & Sons (1968).

4. J. Gray, "A Census of Tandem System Availability Between 1985 and 1990," pp. 409-418 in *IEEE Transactions on Reliability Vol. 39 No. 4* (October 1990).

5. A.M. Johnson, Jr. and M. Malek, "Survey of Software Tools for Evaluating Reliability, Availability, and Serviceability," pp. 227-269 in *ACM Computing Surveys Vol. 20 No. 4* (December 1988).

6. R.A. Maxion and F.E. Feather, "A Case Study of Ethernet Anomalies in a Distributed Computing Environment," pp. 433-443 in *IEEE Transactions on Reliability Vol. 39 No. 4* (October 1990).

7. M.T. Ozsu and P. Valduriez, *Principles of Distributed Database Systems*, Prentice Hall, Englewood Cliffs, NJ, pp. 327-328.

8. R.A. Sahner and K.S. Trivedi, "Reliability Modeling Using SHARPE," pp. 186-193 in *IEEE Transactions on Reliability Vol. 36 No. 2* (June 1987).

9. W.H. Sanders and J.F. Meyer, "METASAN: A Performability Evaluation Tool Based on Stochastic Activity Networks," pp. 807-816 *Proc. 1986 Fall Joint Computer Conference*, AFIPS, New York (November 2-6, 1986).

10. D.P. Siewiorek and R.S. Swarz, *The Theory and Practice of Reliable System Design*, Digital Press, Bedford, MA (1982).

11. T.J. Teorey, *Database Modeling and Design*, Morgan Kaufmann Publishers, Palo Alto, CA (1990).