Optimal Availability Quorum Systems: Theory and Practice

Yair Amir[*] Avishai Wool[†]

April 3, 1998

Abstract

Quorum systems serve as a basic tool providing a uniform and reliable way to achieve coordination in a distributed system. They are useful for distributed and replicated databases, name servers, mutual exclusion, and distributed access control and signatures.

The un-availability of a quorum system is the probability of the event that no live quorum exists in the system. When such an event occurs the service is completely halted. The un-availability is widely accepted as the measure by which quorum systems are evaluated.

In this paper we characterize the optimal availability quorum system in the general case, when the failure probabilities may take *any* value in the range $0 < p_i < 1$. Then we deal with the practical scenario in which the failure probabilities are unknown, but can be estimated. We give a robust and efficient algorithm that calculates a near optimal quorum system based on the estimated failure probabilities.

Keywords: Quorum systems, distributed computing, fault tolerance, replication.

1 Introduction

1.1 Motivation

Quorum systems serve as a basic tool providing a uniform and reliable way to achieve coordination between processors in a distributed system. Quorum systems are defined as follows. A *set system* is a collection of sets $\mathcal{S} = \{S_1, \ldots, S_m\}$ over an underlying universe $U = \{u_1, \ldots, u_n\}$. A set system is said to satisfy the *intersection property*, if every two sets $S, R \in \mathcal{S}$ have a nonempty

[*]Department of Computer Science, The Johns Hopkins University, Baltimore MD 21218, and the NASA Center of Excellence in Space Data and Information Sciences. E-mail: `yairamir@cs.jhu.edu`.

[†]Bell Laboratories, Lucent Technologies, 700 Mountain Avenue, Murray Hill, NJ 07974, USA. This author's work was completed at the Department of Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot 76100, Israel. E-mail: `yash@research.bell-labs.com`.

To appear in Inf. Proc. Letters, 1998.

intersection. Set systems with the intersection property are known as *quorum systems*, and the sets in such a system are called quorums.

Quorum systems have been used in the study of distributed control and management problems such as *data replication protocols* (cf. [DGS85, Her84, Ami95, Kei94]), *name servers* (cf. [MV88]), *mutual exclusion* (cf. [Ray86]), *selective dissemination of information* (cf. [YG94]), and distributed access control and signatures (cf. [NW96]).

A protocol template based on quorum systems works as follows. In order to perform some action (update the database, say), the user selects a quorum and *accesses all of its elements.* The intersection property then guarantees that the user will have a consistent view of the current state of the system.

For example, if all the members of a certain quorum give the user permission to enter the critical section, then any other user trying to enter the critical section before the first user has exited (and released the permission-granting quorum from its lock) will be denied permission by at least one member of any quorum it chooses to access.

1.2 Overview

It is known that when all of the processors have the same independent failure probability p then the optimal availability quorum system is either the Majority system if $p < 1/2$ [BG87], or the Monarchy if $p > 1/2$ [PW95, DKK+94].

If the failure probabilities p_i are different and all less than $1/2$, then the optimal availability quorum system is known to be defined by voting. If $0 < p_i < \frac{1}{2}$ for all i then [TK88] and [SB94] show that the optimal weights are defined by the formula:

$$w_i = \log_2 \left(\frac{1 - p_i}{p_i} \right), \tag{1}$$

followed by a tie-breaking procedure. An exponential algorithm is suggested in [TK88] to find the optimal tie-breaking, while [SB94] shows that a simple scale-and-truncate rule is near optimal.

In this paper we complete this line of research by handling the most general case, where some or all of the elements may be unreliable, with $p_i \geq \frac{1}{2}$. We prove that any element i with $p_i > \frac{1}{2}$ must be a dummy, i.e., w.l.o.g. we can set $w_i = 0$ for such an element. The only exception is when *all* the elements have $p_i \geq \frac{1}{2}$, in which case we prove that the optimal quorum system is a monarchy with one of the least unreliable processors as the king.

As a by-product, we obtain a new proof to the result of [PW95, DKK+94] that when $p_i = p > \frac{1}{2}$ for all i, the monarchy is optimal. Both original proofs used a strong combinatorial tool, namely the Erdős-Ko-Rado theorem [EKR61]. Our proof gives a more general result (allowing different failure probabilities), and is completely elementary.

When using the weights calculated according to [SB94] and formula (1) for a real system, such as in [AW96], one has to handle the singularity when $p_i \to 0$. The ideal value zero may easily be encountered in practice, since the p_i values are typically calculated by measuring the down time of the processors over some finite period. However the singularity at zero is not a mere technicality. We argue that the fact that some processor i has a fault-less past (giving $p_i = 0$) does not imply that it will never fail in the future, so a measured failure probability of zero should not be used in formula (1) as-is. Instead we suggest a simple method to modify the measured failure probabilities in a meaningful way. The new values are all $\hat{p}_i \geq \varepsilon$, so applying formula (1) to them is never problematic.

The organization of this paper is as follows. In Section 2 we introduce the basic definitions and list some useful results. In Section 3 we prove our theoretical results. In Section 4 we discuss the practical issues, and give a complete description of a computationally robust algorithm to calculate near optimal weights.

2 Basic Definitions

2.1 Quorum Systems

The standard definition of a quorum system is the following.

Definition 2.1 *A* Set System $\mathcal{S} = \{S_1, \ldots, S_m\}$ *is a collection of subsets* $S_i \subseteq U$ *of a finite universe* U *representing the processors. A* Quorum System *is a set system* $\mathcal{S}$ *that has the* Intersection property: $S \cap R \neq \emptyset$ *for all* $S, R \in \mathcal{S}$. *The sets of the system are called* quorums.

Many quorum systems which are based on combinatorial constructions appear in the literature, such as [Mae85, GB85, AE91, CAA90, Kum91, KC91, NW94, PW97]. However all the systems with optimal availability turn out to be defined by voting.

Definition 2.2 *Let* v_i *be a positive vote assigned to element* i, *and let* $V = \sum_i v_i$. *The* voting system *defined by the votes* v_i *is the collection of all the sets which have more than half the total vote, i.e., all* $S \subseteq U$ *s.t.* $\sum_{i \in S} v_i > V/2$.

An equivalent definition of a quorum system, which we find more convenient in our proofs, is via the following definition of an acceptance set.

Definition 2.3 *A collection* $\mathcal{A}$ *of sets over a universe* U *is called an* acceptance set *if*

1. $S \cap T \neq \emptyset$ *for all* $S, T \in \mathcal{A}$. *(Intersection)*

2. *If* $S \in \mathcal{A}$ *then* $T \in \mathcal{A}$ *for all* $T \supseteq S$. *(Monotonicity)*

The collection of minimal quorums is $\mathcal{S} = \mathcal{S}(\mathcal{A}) = \{S \in \mathcal{A} : S \setminus \{u\} \notin \mathcal{A} \text{ for all } u \in S\}$.

Note that such an $\mathcal{S}$, which is the collection of minimal sets of an acceptance set, is precisely a coterie (cf. [GB85]).

Definition 2.4 *A* monarchy of element i *is an acceptance set consisting of all the sets containing i. The element i is called the king.*

Definition 2.5 *Let $\mathcal{A}$ be an acceptance set. An element $i \in U$ is a* dummy *if it does not belong to any minimal set, or formally, $i \notin \cup\{S : S \in \mathcal{S}(\mathcal{A})\}$.*

2.2 Availability

For the definition of the availability, we assume that the communication network is fault-free and fully connected. Therefore the network is never partitioned into disconnected components. We only consider fail-stop failures in the system elements (processors), and we assume that the failures are *independent*. Let $\mathbf{p} = (p_1, \ldots, p_n)$ denote the failure probabilities of the processors.

Let $\mathcal{A}$ be an acceptance set. For every set $S \in \mathcal{A}$ we define (with slight abuse of notation) the *event* S, in which the elements of S are alive and the rest are dead. Let $\bar{S}$ denote $U \setminus S$. Then

$$\Pr(S) = \prod_{i \in S}(1 - p_i) \prod_{j \in \bar{S}} p_j.$$

Since the events S are disjoint for different sets, the failure probability (the "un-availability") of $\mathcal{A}$ is

$$F_{\mathbf{p}}(\mathcal{A}) = 1 - \sum_{S \in \mathcal{A}} \Pr(S).$$

We say that a $\mathcal{A}$ has optimal availability for a given vector $\mathbf{p}$ of failure probabilities if $F_{\mathbf{p}}(\mathcal{A}) \leq F_{\mathbf{p}}(\mathcal{B})$ for all acceptance sets $\mathcal{B}$ over the same universe U.

In our proofs we use the following condition, due to [SB94], characterizing a quorum system with optimal availability.

Proposition 2.6 [SB94] *An acceptance set $\mathcal{A}$ has optimal availability iff the following two conditions hold:*

1. *For every $S \subseteq U$, either $S \in \mathcal{A}$ or $\bar{S} \in \mathcal{A}$.*
2. *If $S \in \mathcal{A}$ then $\Pr(S) \geq \Pr(\bar{S})$.*

It is easy to see that condition 1 implies that $\mathcal{S}(\mathcal{A})$ is a non-dominated coterie [GB85].

The special case in which all the failure probabilities are $\frac{1}{2}$ is captured by the following result.

Proposition 2.7 [PW95] *Let $p_i = \frac{1}{2}$ for all $i \in U$. Then $F_{\mathbf{p}}(\mathcal{S}) = \frac{1}{2}$ for any optimal-availability system $\mathcal{S}$.*

3 The Theory

The next proposition is the heart of our proofs. Essentially it shows that unreliable elements, with $p_i > \frac{1}{2}$, are almost always dummies in an optimal availability quorum system.

Proposition 3.1 *Let $0 < p_i < 1$ for all $i \in U$. Let $\mathcal{A}$ be an optimal availability acceptance set and let $\mathcal{S}$ be its collection of minimal quorums. If $p_k > \frac{1}{2}$ for some $k \in U$ then $k \notin S$ for every $S \in \mathcal{S}$ with $|S| > 1$.*

Proof: Let $\gamma_i = \frac{1-p_i}{p_i}$. For any set S

$$\Pr(S) = \prod_{i \in S}(1-p_i) \prod_{j \in \bar{S}} p_j = \prod_{i \in S} \frac{1-p_i}{p_i} \prod_{j \in U} p_j = \prod_{i \in S} \gamma_i \cdot C \tag{2}$$

for $C = \prod_{j \in U} p_j$ which is independent of the set S (indeed, of the whole acceptance set $\mathcal{A}$). Therefore

$$\Pr(S) > \Pr(\bar{S}) \text{ iff } \prod_{i \in S} \gamma_i > \prod_{i \in \bar{S}} \gamma_i. \tag{3}$$

To obtain a contradiction, assume that $k \in S$ for some minimal quorum $S \in \mathcal{S}$ with $|S| > 1$. Let $T = S \setminus \{k\}$, and $\bar{T} = \bar{S} \cup \{k\}$. Note that $T \neq \emptyset$ since $|S| > 1$. Now from the minimality of S it follows that $T \notin \mathcal{A}$. But $\mathcal{A}$ has optimal availability, so by condition 1 of Proposition 2.6 we have that $\bar{T} \in \mathcal{A}$, and by condition 2 we have that $\Pr(T) \leq \Pr(\bar{T})$.

By the premise $p_k > \frac{1}{2}$, so $\gamma_k < 1$, and using condition 2 of Proposition 2.6 for the set S we obtain

$$\prod_{i \in T} \gamma_i > \gamma_k \prod_{i \in T} \gamma_i = \prod_{i \in S} \gamma_i \geq \prod_{i \in \bar{S}} \gamma_i > \gamma_k \prod_{i \in \bar{S}} \gamma_i = \prod_{i \in \bar{T}} \gamma_i,$$

hence by (3) we get $\Pr(T) > \Pr(\bar{T})$, contradiction. ∎

The next corollary shows the only case when an unreliable element k is not a dummy in an optimal availability quorum system: when the acceptance set is in fact a monarchy with k as the king.

Corollary 3.2 *Let $\mathcal{A}$ be an optimal availability acceptance set, let $\mathcal{S}$ be its collection of minimal quorums and let $p_k > \frac{1}{2}$ for some $k \in U$. If $k \in S$ for some $S \in \mathcal{S}$ then $\mathcal{A}$ is a monarchy with $\mathcal{S} = \{\{k\}\}$.*

Proof: By Proposition 3.1 it must be that $|S| = 1$, i.e., $S = \{k\}$. But from the intersection and monotonicity properties it follows that if $\mathcal{A}$ contains a quorum of size 1 then it must be a monarchy. ∎

The next theorem gives a complete characterization of the optimal availability quorum system.

Theorem 3.3 *Let $R = \{i \in U : p_i < \frac{1}{2}\}$ be the set of reliable elements, let $M = \{i \in U : p_i = \frac{1}{2}\}$, and let $B = \{i \in U : p_i > \frac{1}{2}\}$ be the set of unreliable elements. Let $\mathcal{S}_R$ be an optimal* *availability quorum system over R, and let ℓ be some element attaining* $\min_{i\in U}\{p_i\}$. *If $R \neq \emptyset$ then $\mathcal{S}_R$ is optimal over U, otherwise the monarchy $\{\{\ell\}\}$ is optimal.*

Proof: If $R \cup M = \emptyset$, i.e., $U = B$ and all the elements are unreliable, then by Corollary 3.2 the only candidates to have optimal availability are all the monarchies, none of which is better than $\{\{\ell\}\}$.

Otherwise there exists an element $j \notin B$, so the monarchy $\mathcal{Q} = \{\{j\}\}$ has $F_{\mathbf{p}}(\mathcal{Q}) = p_j \leq \frac{1}{2}$, which is strictly better than that of any monarchy based on some $k \in B$. Therefore monarchies in B have suboptimal availability, and we need only consider the systems over $R \cup M$.

By (2) we see that for an element $j \in M$ with $p_j = \frac{1}{2}$ and any set S, it makes no difference to the availability whether $j \in S$ or $j \in \bar{S}$ since $\gamma_j = 1$. So if $R \neq \emptyset$, we may assume that all the elements of M are dummies, and therefore $\mathcal{S}_R$ (which is optimal over R) is optimal over U.

The only remaining case is when $M = U$, i.e., $p_j = \frac{1}{2}$ for all j. But then by Proposition 2.7 we have that $F_{\mathbf{p}}(\mathcal{S}) = \frac{1}{2}$ for *any* optimal-availability system, hence the monarchy $\{\{\ell\}\}$, which clearly has $F_{\mathbf{p}} = \frac{1}{2}$, is optimal. ∎

Remark: Inside the set R we can apply the results of [TK88, SB94] so formula (1) gives the optimal weights.

4 Practical Issues

The weight formula (1) is undefined when $p_i = 0$ (giving $w_i = +\infty$) and when $p_i = 1$ ($w_i = -\infty$). The case $p_i = 1$ is less interesting since by Theorem 3.3 such an element would get $w_i = 0$ and would never be used. However, in addition to the singularity at $p_i = 0$, the case where $p_i \to 0$ needs to be addressed because very small non-zero values of p_i would generate large and unmanageable weights.

In a practical scenario this problem has another aspect, which is related to the source of the probabilities p_i. The p_i values should be ideal quantities representing the steady state when the time period is infinite. However the actual values available to us are based on measurements of time-to-failure and time-to-repair over a finite period of time, so a value of 0 may actually turn up. Moreover, these measured p_i's represent the failure distribution in the *past*, but the computed weights w_i should give optimal availability in the *future*.

With this in mind, we argue that even if element i never failed during the measured period (and got $p_i = 0$) this does not mean it will never fail in the future. The existence of such a "perfect" element should not automatically imply that we should use a monarchy quorum system (which is what $w_i = +\infty$ in fact gives). Likewise, if an element was out of order throughout the measured period and got $p_i = 1$, we would still like to give some small probability ε to the event that it will eventually be repaired.

Therefore we suggest that before calculating the weights, the measured p_i's need to be

1. Let $p_1 \ldots, p_n$ be the measured failures probabilities. Fix values for M and ε.
2. $\hat{p}_i \leftarrow (1-2\varepsilon)p_i + \varepsilon$ for all $i \in U$.
3. For all $i \in U$, if $\hat{p}_i < \frac{1}{2}$ then $v_i \leftarrow \lfloor M \cdot \log_2(\frac{1-\hat{p}_i}{\hat{p}_i}) \rfloor$,
   else $v_i \leftarrow 0$.
4. If $v_i = 0$ for all i then $v_k \leftarrow 1$ for k with the minimal $\hat{p}_i$.
5. If $\sum_i v_i$ is even then $v_1 \leftarrow v_1 + 1$.

Figure 1: Calculating the Optimal Weights.

modified to $\hat{p}_i = h(p_i)$ in a way that would evade the technical problem of $p_i \to 0$, and capture our limited confidence in the measured values. Formula (1) should be then applied to the resultant $\hat{p}_i$ values. Using some small value of ε, the requirements from the function h are:

1. $h(0) = \varepsilon$ and $h(1) = 1 - \varepsilon$.
2. h is monotone increasing.
3. $h(\frac{1}{2}) = \frac{1}{2}$.

Requirement 1 represents our assumption that a perfect past behavior may change in the future. Requirement 2 guarantees that the correction does not alter the relative ranking of failure probabilities. Together with requirement 3 it also ensures that reliable elements ($p_i < \frac{1}{2}$) are not considered to be unreliable ($p \geq \frac{1}{2}$) after the correction, and vice versa. A simple function h fulfilling all the above requirements is the following linear function:

$$h(p) = (1-2\varepsilon)p + \varepsilon. \tag{4}$$

Using this function h, let $\hat{p}_i = h(p_i)$ for any element i. Then if i is reliable, with $0 \leq p_i < \frac{1}{2}$, it will have a corrected probability in the range $\varepsilon \leq \hat{p}_i < \frac{1}{2}$. Therefore its weight calculated by formula (1) using $\hat{p}$ will obey $0 < w_i \leq \log_2\left(\frac{1-\varepsilon}{\varepsilon}\right)$. Since the probabilities $\hat{p}_i$ are inherently imprecise, we lose no accuracy if we break ties using the method of [SB94]: pick a suitably large constant M, calculate $v_i = \lfloor M \cdot w_i \rfloor$, and if $\sum_{i \in U} v_i$ is even[1] then set $v_1 \leftarrow v_1 + 1$. This guarantees that the final weights calculated for reliable elements will be in the range

$$v_i \in \left\{0, \ldots, \lfloor M \cdot \log_2\left(\frac{1-\varepsilon}{\varepsilon}\right) \rfloor + 1\right\}. \tag{5}$$

[1] If the total weight is even then the votes may define a dominated system. An odd total weight guarantees that the system is non-dominated [GB85].

Based on Theorem 3.3, the results of [TK88, SB94], and the above discussion, we construct a practical algorithm to calculate near-optimal weights. This algorithm, presented in Figure 1, calculates the integer weights v_i that define a quorum system with near-optimal availability, for all possible values of p_i.

We have used this algorithm to calculate a quorum system with optimal weights for a system that includes 14 machines, based on measurements that were taken over a period of six months (see [AW96]). We picked $\varepsilon = 0.0001$, and we chose the scaling factor M using (5) so that the computed weights are all in the range $\{0, \ldots, 10000\}$. Some experimentation indicated that choosing larger values for M would not change the resultant quorum system that the computed weights define.

References

[AE91] D. Agrawal and A. El-Abbadi. An efficient and fault-tolerant solution for distributed mutual exclusion. *ACM Trans. Comp. Sys.*, 9(1):1–20, 1991.

[Ami95] Y. Amir. *Replication Using Group Communication Over a Dynamic Network*. PhD thesis, Institute of Computer Science, The Hebrew University of Jerusalem, Israel. Also available at `http://www.cs.jhu.edu/yairamir`, 1995.

[AW96] Y. Amir and A. Wool. Evaluating quorum systems over the Internet. In *Proc. 26'th IEEE Symp. Fault-Tolerant Computing (FTCS)*, pages 26–35, Sendai, Japan, 1996.

[BG87] D. Barbara and H. Garcia-Molina. The reliability of vote mechanisms. *IEEE Trans. Comput.*, C-36:1197–1208, October 1987.

[CAA90] S. Y. Cheung, M. H. Ammar, and M. Ahamad. The grid protocol: A high performance scheme for maintaining replicated data. In *Proc. 6th IEEE Int. Conf. Data Engineering*, pages 438–445, 1990.

[DGS85] S. B. Davidson, H. Garcia-Molina, and D. Skeen. Consistency in partitioned networks. *ACM Computing Surveys*, 17(3):341–370, 1985.

[DKK+94] K. Diks, E. Kranakis, D. Krizanc, B. Mans, and A. Pelc. Optimal coteries and voting schemes. *Inform. Process. Lett.*, 51:1–6, 1994.

[EKR61] P. Erdős, C. Ko, and R. Rado. Intersection theorems for systems of finite sets. *Quart. J. Math. Oxford*, 12(2):313–320, 1961.

[GB85] H. Garcia-Molina and D. Barbara. How to assign votes in a distributed system. *J. ACM*, 32(4):841–860, 1985.

[Her84] M. P. Herlihy. *Replication Methods for Abstract Data Types*. PhD thesis, Massachusetts Institute of Technology, MIT/LCS/TR-319, 1984.

[KC91] A. Kumar and S. Y. Cheung. A high availability $\sqrt{n}$ hierarchical grid algorithm for replicated data. *Inform. Process. Lett.*, 40:311–316, 1991.

[Kei94] I. Keidar. A highly available paradigm for consistent object replication. Master's thesis, Institute of Computer Science, The Hebrew University of Jerusalem, Israel, 1994.

[Kum91] A. Kumar. Hierarchical quorum consensus: A new algorithm for managing replicated data. *IEEE Trans. Comput.*, 40(9):996–1004, 1991.

[Mae85] M. Maekawa. A $\sqrt{n}$ algorithm for mutual exclusion in decentralized systems. *ACM Trans. Comp. Sys.*, 3(2):145–159, 1985.

[MV88] S. J. Mullender and P. M. B. Vitányi. Distributed match-making. *Algorithmica*, 3:367–391, 1988.

[NW94] M. Naor and A. Wool. The load, capacity and availability of quorum systems. In *Proc. 35th IEEE Symp. Foundations of Comp. Sci. (FOCS)*, pages 214–225, 1994. To appear in SIAM J. Computing 1998.

[NW96] M. Naor and A. Wool. Access control and signatures via quorum secret sharing. In *Proc. 3rd ACM Conf. Comp. and Comm. Security*, pages 157–168, New Delhi, India, 1996. Also available as Theory of Cryptography Library record 96-08, `http://theory.lcs.mit.edu/~tcryptol/1996.html`.

[PW95] D. Peleg and A. Wool. The availability of quorum systems. *Information and Computation*, 123(2):210–223, 1995.

[PW97] D. Peleg and A. Wool. Crumbling walls: A class of practical and efficient quorum systems. *Distributed Computing*, 10(2):87–98, 1997.

[Ray86] M. Raynal. *Algorithms for Mutual Exclusion*. MIT press, 1986.

[SB94] M. Spasojevic and P. Berman. Voting as the optimal static pessimistic scheme for managing replicated data. *IEEE Trans. Par. Dist. Sys.*, 5(1):64–73, 1994.

[TK88] Z. Tong and R. Y. Kain. Vote assignments in weighted voting mechanisms. In *Proc. 7th IEEE Symp. on Reliable Dist. Sys.*, pages 138–143, 1988.

[YG94] T. W. Yan and H. Garcia-Molina. Distributed selective dissemination of information. In *Proc. 3rd Inter. Conf. Par. Dist. Info. Sys.*, pages 89–98, 1994.