Crumbling Walls: A Class of Practical and Efficient Quorum Systems

David Peleg* Avishai Wool†

August 14, 1996

Abstract

A *quorum system* is a collection of sets (quorums) every two of which intersect. Quorum systems have been used for many applications in the area of distributed systems, including mutual exclusion, data replication and dissemination of information

In this paper we introduce a general class of quorum systems called *Crumbling Walls* and study its properties. The elements (processors) of a wall are logically arranged in *rows* of varying *widths*. A quorum in a wall is the union of one full row and a representative from every row below the full row. This class considerably generalizes a number of known quorum system constructions.

The best crumbling wall is the CWlog quorum system. It has small quorums, of size $O(\lg n)$, and structural simplicity. The CWlog has optimal availability and optimal load among systems with such small quorum size. It manifests its high quality for *all* universe sizes, so it is a good choice not only for systems with thousands or millions of processors but also for systems with as few as 3 or 5 processors. Moreover, our analysis shows that the availability will increase and the load will decrease at the optimal rates as the system increases in size.

Keywords: availability, coteries, distributed computing, fault tolerance, load, quorum systems.

*Department of Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot 76100, Israel. Supported in part by a Walter and Elise Haas Career Development Award and by a grant from the Israel Science Foundation. E-mail: peleg@wisdom.weizmann.ac.il.

†Department of Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot 76100, Israel. E-mail: yash@wisdom.weizmann.ac.il.

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 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 *mutual exclusion* (cf. [34]), *data replication protocols* (cf. [8, 14]), *name servers* (cf. [24]), *selective dissemination of information* (cf. [37]), and distributed access control and signatures (cf. [26]).

A protocol template based on quorum systems works as follows. In order to perform some action (e.g., update the database, enter a critical section), the user selects a quorum and *accesses all 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 refused permission by at least one member of any quorum it chooses to access.

We consider three criteria of measuring the quality of a quorum system:

1. *Quorum size* - having small quorums has obvious advantages such as a low message complexity of the protocol using the system or a low number of replicas kept.

2. *Availability* - assuming that each element fails with probability p, what is the probability, F_p, that the surviving elements do not contain any quorum? This failure probability measures how resilient the system is, and we would like F_p to be as small as possible. A desirable asymptotic behavior of F_p is that $F_p \to 0$ when $n \to \infty$ for *all* $p < \frac{1}{2}$, and such an F_p is called Condorcet.

3. *Load* - A strategy is a rule giving each quorum an access probability (so that the probabilities sum up to 1). A strategy induces a load on each element, which is the sum of the probabilities of all quorums it belongs to. This represents the fraction of the time an element is used. For a given quorum system $\mathcal{S}$, the load $\mathcal{L}(\mathcal{S})$ is the minimal load on the busiest element, minimizing over the strategies. The load measures the quality of a quorum system in the following sense. If the load is low, then each element is accessed rarely, thus it is free to perform other unrelated tasks.

These criteria are conflicting, so there can be no quorum system construction that is optimal with respect to all of them. The quorum systems which have optimal availability or optimal load (or achieve a tight tradeoff between these two criteria) have relatively large quorums, of size $\Omega(\sqrt{n})$. Additionally some of the best systems are asymptotic in nature, manifesting their optimality only in very large systems. This situation leads to a quest for new quorum system constructions that combine small quorum sizes with high availability and low load, both asymptotically *and* for practical system sizes.

1.2 Related Work

The first distributed control protocols using quorum systems [36, 12] use *voting* to define the quorums. Each processor has a number of votes, and a quorum is any set of processors with a combined number of votes exceeding half of the system's total number of votes. The simple majority system is the most obvious voting system.

The availability of voting systems is studied in [5]. It is shown that in terms of availability, the majority is the best quorum system when $p < \frac{1}{2}$. In [28, 9] the failure probability function F_p is characterized, and among other things it is shown that the singleton has the best availability when $p > \frac{1}{2}$. The case when the elements fail with different probabilities p_i is addressed in [35] and extended in [4].

The first paper to explicitly consider mutual exclusion protocols in the context of intersecting set systems is [11]. In this work the term *coterie* and the concept of *domination* are introduced. Several basic properties of dominated and non-dominated coteries are proved.

Alternative protocols based on quorum systems (rather than on voting) appear in [22] (using finite projective planes), [1] (the Tree system), [6, 20] (using a grid), [18, 19, 33, 32] (hierarchical systems). The triangular system is due to [21, 10]. A generalization of the triangular system appears in [27] under the name Lovász coteries. The Wheel system appears in [23].

In [15], the question of how evenly balanced the work load can be is studied. Tradeoffs between the potential load balancing of a system and its average load are obtained. The notion of load is studied further in [25]. Lower bounds on the load and tradeoffs between the load and availability are shown. Four quorum system constructions are shown, featuring optimal load and high availability. The question of how many probes are needed for a live quorum to be found is addressed in [31].

While the majority quorum system is the best in terms of availability, and the finite projective planes (FPP) construction has excellent load, they fail according to the other criteria: the load of majority is 1/2 and the failure probability of the FPP tends to 1 as the number of elements grows. The constructions of [25] have both optimal load and high availability, however the availability becomes high only for large values of n. Additionally, all the existing constructions have quorum sizes larger than $\sqrt{n}$ (except for the Tree construction of [1]).

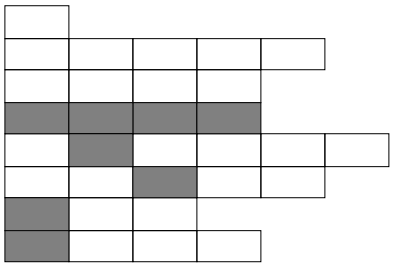

Figure 1: The crumbling wall $\mathrm{CW}\langle 1,5,4,4,6,5,3,4\rangle$, with one quorum shaded.

1.3 New Results

This paper introduces a new class of quorum system constructions, which we call *Crumbling Walls* (or simply walls). The crumbling walls are a generalization of the triangular construction of [21, 10], the Grid of [6], the hollow grids of [20], the Wheel of [23] and the Lovász coteries of [27]. The elements are arranged in *rows*, and a quorum is the union of one full row and a single representative from every row below the full row. However, unlike the triangular system, we do not require that row i have exactly i elements, and allow the "wall" to crumble at its edge (see Figure 1). A crumbling wall with a sequence of row widths $\boldsymbol{n} = (n_1, n_2, \ldots, n_d)$ is denoted by $\mathrm{CW}\langle \boldsymbol{n} \rangle$.

We first discuss some general properties of the crumbling wall construction. We show that a wall is a non-dominated (ND) coterie iff the first row is of width 1 and rows $2, \ldots, d$ are of width ≥ 2. It follows that the number of ND walls over a universe of size n elements is exponential in n (in fact it is exactly a Fibonacci number). Then we show that for any element failure probability $0 < p < 1$, the availability of a wall is improved if the widths form a monotone increasing sequence. We also consider the load of crumbling walls. We prove a lower bound on the load, and show access strategies that achieve near optimal load.

Next we introduce what we consider to be the best crumbling wall, the CWlog system, with quorums of size $\lg n - \lg \lg n$.[1] We show that it has optimal load among the quorum systems with logarithmic size quorums, namely $\mathcal{L}(\mathrm{CWlog}) = O(1/\log n)$. In [30] it is shown that CWlog also has optimal availability among quorum systems in that class, namely $F_p(\mathrm{CWlog}) = O(n^{-\varepsilon})$ for some constant $\varepsilon(p) > 0$. We show that CWlog has high availability for small universe sizes as well; its availability is much better than the Grid and slightly better than the Tree, beginning from universe size $n = 5$. We present two simple procedures to pick quorums, designed to minimize different criteria. The first always picks the smallest live quorum but induces a high load. The second induces a near optimal load but occasionally picks larger quorums. We show that the asymptotic load of the CWlog system remains low even when failures may occur. Specifically, as long as the elements' failure probability is below 0.432 then with high probability the CWlog still has load of $O(1/\log n)$. We conclude that CWlog is a good candidate to be

[1] We use lg to denote $\log_2$.

the construction of choice in practice, featuring high availability, low load, small quorums, and structural simplicity.

The organization of this paper is as follows. In Section 2 we introduce the definitions and notation, and list some useful theorems. Section 3 contains proofs of the basic properties of crumbling walls. In Section 4 we introduce the CWlog system and discuss its properties.

An extended abstract of this paper can be found in [29].

2 Preliminaries

2.1 Definitions and Notation

Let us first define the basic terminology used later on.

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. *A* Quorum System *is a set system* $\mathcal{S}$ *that has the* Intersection property: $S \cap R \neq \varnothing$ *for all* $S, R \in \mathcal{S}$.

Alternatively, quorum systems are known as *intersecting set systems* or as *intersecting hypergraphs*. The sets of the system are called *quorums*. The number of elements in the underlying universe is denoted by $n = |U|$. The cardinality of the smallest quorum in $\mathcal{S}$ is denoted by $c(\mathcal{S}) = \min\{|S| : S \in \mathcal{S}\}$.

Definition 2.2 *A* Coterie *is a quorum system* $\mathcal{S}$ *that has the* Minimality property: *there are no* $S, R \in \mathcal{S}$, *s.t.* $S \subset R$.

Definition 2.3 *Let* $\mathcal{R}, \mathcal{S}$ *be coteries (over the same universe* U*). Then* $\mathcal{R}$ dominates $\mathcal{S}$, *denoted* $\mathcal{R} \succ \mathcal{S}$, *if* $\mathcal{R} \neq \mathcal{S}$ *and for each* $S \in \mathcal{S}$ *there is* $R \in \mathcal{R}$ *such that* $R \subseteq S$. *A coterie* $\mathcal{S}$ *is called* dominated *if there exists a coterie* $\mathcal{R}$ *such that* $\mathcal{R} \succ \mathcal{S}$. *If no such coterie exists then* $\mathcal{S}$ *is* non-dominated (ND). *Let* NDC *denote the class of all* ND *coteries.*

2.2 Examples

Let us illustrate the concept of quorum systems by giving some examples, that play an important role in the results of this paper. The following constructions are known to be non-dominated coteries, except for the Grid system.

The singleton system, denoted by Sngl, is the set system $\text{Sngl} = \{\{u\}\}$.

The majority system [36], denoted by Maj, is the collection of all sets of $\frac{n+1}{2}$ elements over a universe U, when $n = |U|$ is odd.

The Wheel [23, 28] contains $n-1$ "spoke" quorums of the form $\{1, i\}$ for $i = 2, \ldots, n$, and one "rim" quorum, $\{2, \ldots, n\}$.

In the triangular system [21, 10], denoted by Triang, the elements are arranged in d rows, with row i containing i elements. A quorum is any set composed of one complete row i, and a representative from every row $j > i$.

The Lovász coteries of [27] are a generalization of the Triang, in which the first row contains a single element, and all the other rows contain at least 2 element each. A quorum in the system is defined as in the Triang.

In the Tree system [1] the elements are organized in a complete rooted binary tree. A quorum in the system is defined recursively to be either (i) the union of the root and a quorum in one of the two subtrees, or (ii) the union of two quorums, one in each subtree.

In the Grid [6] the $n = d^2$ elements are arranged in a $d \times d$ grid, and a quorum in the system consists of one complete row and a representative element from all the other rows.

2.3 The Probabilistic Failure Model

We use a simple probabilistic model of the failures in the system. We assume that the elements (processors) fail independently with a fixed uniform probability p. We assume that the failures are *transient*, that the failures are *crash* failures (i.e., a failed element stops to function rather than functions incorrectly), and that they are *detectable*.

Note that this model implicitly assumes that the communication links are perfect, and that the network is fully connected, hence the network never partitions. In general this is an over-simplification of real communication networks (see [3] for an empirical evaluation of network connectivity). However we believe that such a model is reasonable for some important cases, and especially for a well maintained local area network (LAN).

Notation: We use $q = 1 - p$ to denote the probability of an element survival.

In this failure model with probability p, the following events can be defined.

Definition 2.4 (Quorum failure) *For every quorum $S \in \mathcal{S}$ let $\mathcal{E}_S$ be the event that S is* hit*, i.e., at least one element $i \in S$ has failed. Let $fail(\mathcal{S})$ be the event that all the quorums $S \in \mathcal{S}$ were hit, i.e., $fail(\mathcal{S}) = \bigwedge_{S \in \mathcal{S}} \mathcal{E}_S$.*

Now we can define the global system failure probability of a quorum system $\mathcal{S}$ (cf. [28]), as follows.

Definition 2.5 $F_p(\mathcal{S}) = \mathbb{P}(fail(\mathcal{S})) = \mathbb{P}\left(\bigwedge_{S \in \mathcal{S}} \mathcal{E}_S\right)$.

The following theorems of [28] describe some properties of the failure probability F_p.

Theorem 2.6 (Symmetry) [28] *For any $\mathcal{S} \in$ NDC, $F_p(\mathcal{S}) + F_{1-p}(\mathcal{S}) = 1$.*

When we consider the asymptotic behavior of $F_p(\mathcal{S}_n)$ for a sequence $\mathcal{S}_n$ of quorum systems over a universe with an increasing size n, we find that for many constructions it is similar to the behavior described by the Condorcet Jury Theorem [7]. Hence, the following definition of

[28].

Definition 2.7 [28] *A parameterized family of functions* $g_p(n) : \mathbb{N} \to [0,1]$, *for* $p \in [0,1]$, *is said to be* Condorcet *if* $\lim_{n\to\infty} g_p(n) = \begin{cases} 0, & p < \frac{1}{2}, \\ 1, & p > \frac{1}{2}, \end{cases}$ *and* $g_{1/2}(n) = \frac{1}{2}$ *for all* n.

In [28] it is shown that the Maj and Tree quorum systems have Condorcet failure probability functions, while the Sngl, Wheel, Triang and Grid systems do not.

2.4 The Load

In this section we list some definitions and theorems from [25] regarding the load of a quorum system.

A protocol using a quorum system (for mutual exclusion, say) occasionally needs to access quorums during its run. A strategy is a probabilistic rule that governs which quorum is chosen each time. In other words, a strategy gives the probability that a quorum S_j will be picked.

Definition 2.8 *Let a quorum system* $\mathcal{S} = (S_1, \ldots, S_m)$ *be given over a universe* U. *Then* $w \in [0,1]^m$ *is a* strategy *for* $\mathcal{S}$ *if it is a probability distribution over the quorums* $S_j \in \mathcal{S}$, *i.e.,* $\sum_{j=1}^{m} w_j = 1$.

For every element $i \in U$, a strategy w of picking quorums induces a probability that the element i is accessed, which we call the *load* on i. The *system load*, $\mathcal{L}(\mathcal{S})$, is the load on the *busiest* element induced by the *best* possible strategy.

Definition 2.9 *Let a strategy* w *be given for a quorum system* $\mathcal{S} = (S_1, \ldots, S_m)$ *over a universe* U. *For an element* $i \in U$, *the* load *induced by* w *on* i *is* $\ell_w(i) = \sum_{S_j \ni i} w_j$. *The* load *induced by a strategy* w *on a quorum system* $\mathcal{S}$ *is*

$$\mathcal{L}_w(\mathcal{S}) = \max_{i \in U} \ell_w(i).$$

The system load *on a quorum system* $\mathcal{S}$ *is* $\mathcal{L}(\mathcal{S}) = \min_w \{\mathcal{L}_w(\mathcal{S})\}$, *where the minimum is taken over all strategies* w.

Following are lower bounds of [25] on the load $\mathcal{L}(\mathcal{S})$ and the failure probability F_p in terms of the smallest quorum size $c(\mathcal{S})$.

Proposition 2.10 [25] $\mathcal{L}(\mathcal{S}) \geq \frac{1}{c(\mathcal{S})}$ *for any quorum system* $\mathcal{S}$.

Proposition 2.11 [25] $F_p(\mathcal{S}) \geq p^{c(\mathcal{S})}$ *for any quorum system* $\mathcal{S}$ *and any* $p \in [0,1]$.

In [25] it is shown that the Maj and Wheel have a load of $> \frac{1}{2}$, while the Tree has a load of $O(1/\lg n)$ and the Grid and Triang have a load of $O(1/\sqrt{n})$ (which is optimal up to constants).

3 Basic Properties of Crumbling Walls

3.1 What Are Crumbling Walls

Definition 3.1 (Crumbling Wall) *Let* $\boldsymbol{n} = (n_1, \ldots, n_d)$ *be such that* $\sum_{i=1}^{d} n_i = n$. *Let* $U_1, \ldots, U_d$ *be nonempty disjoint subsets of the universe* U *with* $|U_i| = n_i$. *Then*

$$\mathrm{CW}\langle \boldsymbol{n} \rangle = \Big\{ U_i \cup \{u_{i+1}, \ldots, u_d\} : u_j \in U_j \ \text{ for } \ j = i+1, \ldots, d \Big\}$$

is the crumbling wall defined by $\boldsymbol{n}$. *The set* U_i *is called the i'th* row *and* n_i *is its* width. *A quorum that uses row i as the full row is called* based on row i.

The class of crumbling walls encompasses a number of other coterie classes as special cases: the Sngl, Triang, Wheel, Grid and Lovász coteries. The Sngl coterie is a trivial wall with $\boldsymbol{n} = (1)$, the Triang with d rows is a wall defined by $\boldsymbol{n} = (1, 2, \ldots, d)$, the Wheel over n elements is a wall defined by $\boldsymbol{n} = (1, n-1)$, and a $d \times d$ Grid is a wall defined by $\boldsymbol{n} = (d, d, \ldots, d)$.[2] A Lovász coterie is a wall with $n_1 = 1$ and $n_i \geq 2$ for all $i \geq 2$.

The following proposition of [27] shows that Lovász coteries are ND.

Proposition 3.2 [27] *If* $n_1 = 1$ *and* $n_i \geq 2$ *for all* $i \geq 2$ *then* $\mathrm{CW}\langle \boldsymbol{n} \rangle \in \mathrm{NDC}$.

In Proposition 3.5 we extend this result, showing that these are in fact the *only* ND walls. We do this via two simple lemmas.

Lemma 3.3 *If* $n_i = 1$ *for some* $i \geq 2$ *then* $\mathrm{CW}\langle \boldsymbol{n} \rangle$ *is not a coterie.*

Proof: Assume that there exists some $i \geq 2$ such that $n_i = 1$. Then any quorum $S \in \mathrm{CW}\langle \boldsymbol{n} \rangle$ that is based on row 1 contains the single element in row i, i.e., the whole U_i. But then S contains some other quorum $R \in \mathrm{CW}\langle \boldsymbol{n} \rangle$ (that is based or row i), violating the Minimality property, so $\mathrm{CW}\langle \boldsymbol{n} \rangle$ is not a coterie. ∎

Lemma 3.4 *If* $n_i \geq 2$ *for all* i *then* $\mathrm{CW}\langle \boldsymbol{n} \rangle$ *is dominated.*

Proof: Any set $T = \{u_1, \ldots, u_d\}$ with $u_i \in U_i$ for $1 \leq i \leq d$ intersects all the quorums, but $T \notin \mathrm{CW}\langle \boldsymbol{n} \rangle$. Therefore $\mathrm{CW}\langle \boldsymbol{n} \rangle$ is dominated. ∎

Proposition 3.5 $\mathrm{CW}\langle \boldsymbol{n} \rangle \in \mathrm{NDC}$ *iff* $n_1 = 1$ *and* $n_i \geq 2$ *for all* $2 \leq i \leq d$.

Proof: Immediate from Proposition 3.2, Lemmas 3.3 and 3.4. ∎

[2] Usually a quorum in a Grid is one full row and a representative in *every* other row. Our somewhat improved variant, in which representatives are required only *below* the full row, has smaller quorums and dominates the regular Grid.

3.2 The Number of ND Walls

The number of ND coteries over a universe of size n is $2^{2^{cn}}$ for some constant c (Yannakakis, cf. [11]). Of these, roughly 2^{n^2} are voting coteries ([11, 16] and the references therein).

The following proposition shows that the number of ND walls is exponential in n (in fact, it is exactly a Fibonacci number). Note however that here we count *non-isomorphic* walls, i.e., the number of different ND wall *shapes*.

Proposition 3.6 *The number of non-dominated walls over a universe of size $n \geq 3$ is $Fib(n-3)$, where $Fib(i)$ is the i'th Fibonacci number, $Fib(0) = 1, Fib(1) = 1$.*

Proof: Following Proposition 3.5, the first row of an ND wall is of width 1, and all the other rows are of width ≥ 2. If there are d rows in the wall, then we need to distribute $n - 2d + 1$ identical elements among $d - 1$ distinct rows (excluding the first row). There are

$$\binom{(n-2d+1)+(d-1)-1}{(d-1)-1} = \binom{n-d-1}{d-2}$$

ways to do so. Therefore

$$\#\text{Walls} = \sum_d \binom{n-d-1}{d-2} = \sum_j \binom{n-3-j}{j}$$

where the summations are over all the values giving nonzero binomial coefficients. Using a combinatorial identity [17, p. 84] we get $\#\text{Walls} = Fib(n-3)$. ∎

Remark: In order for this result to be comparable to the numbers of ND coteries and voting systems, we must also take into account the number of ways of mapping n elements onto a wall. But even if we ignore the fact that elements in the same row are equivalent, and we multiply the result of the proposition by $n!$, then $\#\text{Walls} \leq 2^{O(n \log n)}$, which is still very small in comparison to both voting and general ND coterie numbers.

3.3 The Failure Probability of Crumbling Walls

To calculate the failure probability of a given crumbling wall, consider the following procedure to search the wall for either a complete quorum or a failure configuration. We go over the rows from the bottom up, starting with row d. At row i we have three options:

1. If $i = 0$ or all n_i elements in the row have failed, stop; the system has failed.
2. If all n_i elements in the row are alive, stop; there is a live quorum in the system.
3. Otherwise, continue to row $i - 1$.

A moment's reflection reveals that the procedure considers row $i-1$ only if row i has both a failed element and a live one. Therefore if a fully live row is found, its union with all the live elements in rows below it gives a live quorum. On the other hand, if a fully failed row is found, then it is pointless to search rows above it and we know that all rows below it contain a failed element, so no live quorum exists. If no row is fully live then obviously no live quorum exists. Thus both stopping decisions are correct.

Note that if row 1 consists of a single element, then there is no need to check if i reaches zero since the procedure must fall into one of the stopping cases.

Notation: Let $F_p(i)$ denote F_p of the sub-wall of the top i rows.

Fact 3.7 *The failure probability $F_p(i)$ obeys recurrence*

$$\begin{cases} F_p(1) = 1 - q^{n_1}, \\ F_p(i) = p^{n_i} + (1 - p^{n_i} - q^{n_i})F_p(i-1), \quad i > 1. \end{cases}$$

When $n_1 = 1$ then $1 - q^{n_1} = p$, so we can expand the recurrence to get

Fact 3.8 *The failure probability of a wall* $\mathrm{CW}\langle \boldsymbol{n} \rangle$ *on d rows with $n_1 = 1$ is*

$$F_p(\mathrm{CW}\langle \boldsymbol{n} \rangle) = \sum_{i=1}^{d} p^{n_i} \prod_{j=i+1}^{d} \left(1 - p^{n_j} - q^{n_j}\right).$$

3.4 The Advantage of Monotone Increasing Walls

In this section we prove that walls with monotone increasing row widths have the best availability among all the row permutations.

Lemma 3.9 *Let $\mathcal{S} = \mathrm{CW}\langle s_1, \ldots, s_d \rangle$ be an* ND *wall, and let i be such that $s_{i+1} < s_i$. Consider the wall with rows i and $i+1$ switched, namely, $\mathcal{R} = \mathrm{CW}\langle r_1, \ldots, r_d \rangle$ such that $r_i = s_{i+1}$, $r_{i+1} = s_i$, and $r_j = s_j$ for all other j's. If $p < \frac{1}{2}$ then $F_p(\mathcal{R}) < F_p(\mathcal{S})$.*

Proof: Since $\mathcal{S} \in \mathrm{NDC}$ then by Proposition 3.5 $s_1 = 1$, therefore $i \neq 1$ (otherwise $s_2 < 1$ which is impossible), and then $r_1 = 1$ as well. Therefore we can use Fact 3.8 and write

$$F_p(\mathcal{S}) = \sum_{k=1}^{d} p^{s_k} \prod_{j=k+1}^{d} \left(1 - p^{s_j} - q^{s_j}\right),$$

and similarly for $\mathcal{R}$. Consider the difference $F_p(\mathcal{S}) - F_p(\mathcal{R})$, term by term according to the index k. If $k > i+1$ then $s_j = r_j$ for all $j \geq k$, so this term contributes 0 to the difference. If $k < i$ then the products are of the same values (reordered), so again this term contributes nothing. Therefore

$$F_p(\mathcal{S}) - F_p(\mathcal{R}) =$$
$$\prod_{j>i+1} (1 - p^{s_j} - q^{s_j}) \Big[p^{s_i}(1 - p^{s_{i+1}} - q^{s_{i+1}}) + p^{s_{i+1}} - p^{r_i}(1 - p^{r_{i+1}} - q^{r_{i+1}}) - p^{r_{i+1}} \Big].$$

Since we only care about the sign of the expression, we can drop the product and plug r_i to get

$$p^{s_i}(1-p^{s_{i+1}}-q^{s_{i+1}})+p^{s_{i+1}}-p^{s_{i+1}}(1-p^{s_i}-q^{s_i})-p^{s_i}=$$
$$p^{s_{i+1}}q^{s_{i+1}}(q^{s_i-s_{i+1}}-p^{s_i-s_{i+1}}),$$

and when $s_{i+1} < s_i$ and $p < \frac{1}{2} < q$ the last expression is strictly positive. ∎

Remark: Lemma 3.9 holds when $\mathcal{S} \notin$ NDC as well, i.e., when $s_1 \neq 1$. However the proof becomes somewhat more cumbersome, so for clarity it is omitted.

By applying Lemma 3.9 repeatedly to any given wall system with non-monotone row widths we conclude:

Corollary 3.10 *Out of all the walls defined by some permutation of* $(n_1, \ldots, n_d)$*, the wall with the minimal failure probability when* $0 < p < \frac{1}{2}$ *has its rows in a monotone non-decreasing order of widths.* ∎

3.5 The Load of Crumbling Walls

In this section we consider the load $\mathcal{L}(\mathrm{CW}\langle \boldsymbol{n} \rangle)$ of a crumbling wall. We first show a lower bound on the load. Then we classify a wall as either *normal* or *truncated*, and describe a simple access strategy for each kind of wall. We prove that in both cases the induced load is at most twice the optimum.

Proposition 3.11 *Let* $c = c(\mathrm{CW}\langle \boldsymbol{n} \rangle)$ *be the size of the smallest quorum in a wall* $\mathrm{CW}\langle \boldsymbol{n} \rangle$ *with* d *rows. Then* $\mathcal{L}(\mathrm{CW}\langle \boldsymbol{n} \rangle) \geq \max\left\{\frac{1}{c}, \frac{1}{d}\right\}$.

Proof: The first term in the maximum is just a re-statement of Proposition 2.10. For the second term, consider some collection $\{u_1, \ldots, u_d\}$ of elements, one from each row. Since every quorum contains at least one such u_j, any strategy must access some u_j with probability $\geq 1/d$, hence $\mathcal{L}(\mathrm{CW}\langle \boldsymbol{n} \rangle) \geq 1/d$. ∎

Note that any quorum based on row i has size $n_i + d - i$. We are interested in the *critical* row, on which the smallest quorums are based.

Definition 3.12 *Let the* critical row *be the row* r *on which* $\min_i\{n_i - i\}$ *is achieved. Call a wall* normal *if* $n_r \leq r$*, and* truncated *otherwise.*

Remarks:

- A wall is truncated if its "top rows are missing." Below we show that for such a wall the number of rows d is smaller than the minimal quorum cardinality c. Moreover, an ND wall is never truncated; if $n_1 = 1$ then for the critical row r we have $n_r - r \leq n_1 - 1 = 0$ so $n_r \leq r$ and the wall is normal.
- There may be more than one row on which the minimum is achieved. In such a case define r arbitrarily to be one such row.

1. The rows are $U_1, \ldots, U_d$.
2. Pick a row i in the range $d - t + 1 \leq i \leq d$ at random with probability $1/t$.
3. Set $Q \leftarrow \varnothing$. For all $j > i$, pick an element $u_j \in U_j$ at random with probability $1/n_j$, and add it to Q.
4. return $U_i \cup Q$.

Figure 2: Procedure Pick(t) to pick a quorum based on one of the bottom t rows.

Procedure Pick(t) (given in Figure 2) is a simple strategy template of choosing which quorum to access, depending on the value of the parameter t. It only picks quorums which are based on one of the t bottom rows.

A natural way of using procedure Pick is to randomize over all d rows, i.e., to use Pick(d). However this may induce a high load in some cases. For instance, consider a wall W whose $n/4$ top rows are of width 2 and whose bottom $\sqrt{n}/2$ rows are of width $\sqrt{n}$. Note that $c(W) = \sqrt{n}/2 + 2$ but $d = n/4 + \sqrt{n}/2$. For this W, randomizing over all d rows would induce a load of $\approx 1/2$ on the two elements in row $n/4$, instead of the $O(1/\sqrt{n})$ we could hope for.

The solution is to randomize only over a certain number of the bottom rows. The next proposition shows that for normal walls, using Pick(c) where $c = c(\mathrm{CW}\langle \boldsymbol{n} \rangle)$ achieves almost optimal load.

Proposition 3.13 *Let r be the critical row of $\mathrm{CW}\langle \boldsymbol{n} \rangle$ and let $c = n_r + d - r$ be the size of the smallest quorum. If $n_r \leq r$ then strategy $w_1 \equiv \mathrm{Pick}(c)$ induces a load of*

$$\mathcal{L}_{w_1}(\mathrm{CW}\langle \boldsymbol{n} \rangle) \leq \frac{2}{c} < 2\mathcal{L}(\mathrm{CW}\langle \boldsymbol{n} \rangle).$$

Proof: Since $n_r \leq r$, the number of rows d satisfies $d \geq n_r + d - r = c$. Therefore we can speak of using Pick on the bottom c rows (starting from row $r - n_r + 1$) and strategy w_1 is well defined. An element u on row i among the bottom c rows is used either if row i is picked to be the full row, or if the full row is some row $k < i$ and u is chosen as a representative. Therefore the load that w_1 induces on such a u is

$$\ell(u) = \frac{1}{c} + \frac{i - (r - n_r + 1)}{c} \cdot \frac{1}{n_i} = \frac{1}{c}\left(1 + \frac{i - 1 + n_r - r}{n_i}\right),$$

but $n_r - r \leq n_i - i$ so

$$\ell(u) \leq \frac{1}{c}\left(1 + \frac{n_i - 1}{n_i}\right) < \frac{2}{c}.$$

By Proposition 3.11, w_1 induces a load which is at most twice the optimum. Note that for normal walls the tighter lower bound of Proposition 3.11 is $1/c$. ∎

Remark: Most of the known wall constructions are normal, so strategy w_1 induces the following loads: $\mathcal{L}_{w_1}(\text{Grid}) \leq \frac{2}{\sqrt{n}}$, $\mathcal{L}_{w_1}(\text{Triang}) \lessapprox \frac{\sqrt{2}}{\sqrt{n}}$ and $\mathcal{L}_{w_1}(\text{Wheel}) \leq \frac{1}{2}(1 + \frac{1}{n-1})$.

In truncated walls ($n_r > r$) we cannot apply Pick on the bottom c rows, since there are too few rows ($d < c$). However the next proposition shows that in this case using Pick on *all* d rows is again almost optimal.

Proposition 3.14 *Let r be the critical row of* $\text{CW}\langle \boldsymbol{n} \rangle$. *If $n_r > r$ then strategy $w_2 \equiv \text{Pick}(d)$ induces a load of*

$$\mathcal{L}_{w_2}(\text{CW}\langle \boldsymbol{n} \rangle) \leq \frac{2}{d} < 2\mathcal{L}(\text{CW}\langle \boldsymbol{n} \rangle).$$

Proof: By a similar argument to the one in Proposition 3.13, the load induced by w_2 on an element u in row i is

$$\ell(u) = \frac{1}{d}\left(1 + \frac{i-1}{n_i}\right).$$

By the definition of r and the fact that the wall is truncated it follows that $n_i - i \geq n_r - r > 0$, so $\ell(u) < 2/d$. By Proposition 3.11, w_2 induces a load which is at most twice the optimum. Note that for truncated walls the tighter lower bound is $1/d$. ∎

4 The CWlog System

4.1 The Construction

In this section we focus our attention to a specific crumbling wall which we call the CWlog. The width of row i in the CWlog is $n_i = \lfloor \lg 2i \rfloor$ (see Figure 3). We wish to demonstrate that aside from the theoretic interest, the CWlog wall has merit as a practical construction of a quorum system.

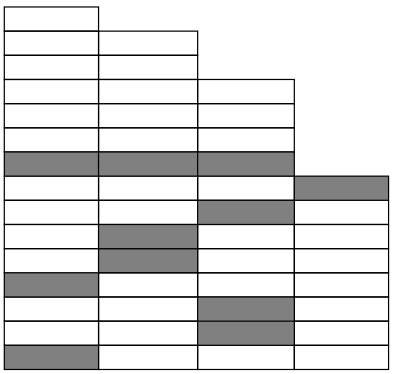

Figure 3: A CWlog with $n = 49$ elements and $d = 15$ rows, with one quorum shaded.

In a CWlog with d rows, the width of the bottom row (which in itself is the smallest quorum in the system) is $\lfloor \lg 2d \rfloor$. It is easy to observe that every integer $k \geq 1$ appears precisely 2^{k-1}

times in the sequence $n_i = \lfloor \lg 2i \rfloor$. It follows that in terms of the universe size $n = \sum_{i=1}^{d} n_i$, the smallest quorum is of size $c(\text{CWlog}) \approx \lg n - \lg\lg n$. The largest quorums are based on row 1, and have a size of $d \approx n/\lg n$. Clearly CWlog is a Lovász coterie, so by Proposition 3.5 it follows that CWlog $\in$ NDC.

Let us point out that the CWlog wall is a very simple construction, and is easy to implement. The elements need to be logically arranged in rows of widths n_i, and then a procedure is needed to produce a quorum on demand. In the sequel we suggest two alternative procedures to pick a quorum, with slightly different properties.

4.2 The Availability of CWlog

In [30] we analyze the asymptotic failure probability of general crumbling walls, and show that CWlog is essentially the only high-availability wall. As a part of this analysis we obtain the following theorem, which describes the asymptotic behavior of $F_p(\text{CWlog})$.

Theorem 4.1 [30] *Consider the* CWlog *system on d rows, with $n_i = \lfloor \lg 2i \rfloor$, let $q = 1-p$, and let α be such that $\alpha + \lg(1/\alpha) = 2$ ($\alpha \approx 0.3099$). Then*

$$F_p(\text{CWlog}) \leq \begin{cases} C_1 \left(\frac{1}{d}\right)^q, & 0 < p < \alpha, \\ C_2 \frac{\lg d}{d^q}, & p = \alpha, \\ C_3 \left(\frac{1}{d}\right)^{(\lg \frac{1}{p} - 1)}, & \alpha < p < \frac{1}{2}, \end{cases}$$

for some C_1, C_2, C_3 that depend only on p. Therefore $F_p(\text{CWlog}) \underset{d\to\infty}{\longrightarrow} 0$ for all $0 < p < \frac{1}{2}$, thus $F_p(\text{CWlog})$ is Condorcet.

Theorem 4.1 shows that the CWlog has high availability, with $F_p(\text{CWlog}) = O((\frac{\lg n}{n})^{\varepsilon})$ for all $0 < p < \frac{1}{2}$, for some $\varepsilon(p) > 0$, i.e., a Condorcet failure probability. By Proposition 2.11 we have:

Theorem 4.2 *The availability of CWlog is optimal up to a constant factor for quorum systems with $c(\mathcal{S}) = O(\lg n)$.*

In particular, this means that CWlog is asymptotically superior to the FPP [22] and the Grid [6], both of which have failure probabilities tending to 1 (see [19, 32]). The CWlog has asymptotic availability similar to that of the Tree system of [1] (as analyzed in [28]).

The CWlog has worse asymptotic availability than the constructions of [18, 19, 25] and than the Maj system [36], which has the optimal availability [5]. However all these construction have relatively large quorums, of size $\Omega(\sqrt{n})$ or $\lceil (n+1)/2 \rceil$ for the Maj.

Unlike the constructions of [18, 25], the availability of CWlog is high not only for very large n. In Figure 4 we show $F_p(\text{CWlog})$ as a function of the universe size, in the range $1 \leq n \leq 100$, for $p = 0.1$ and $p = 0.3$. For comparison we show $F_p(\text{Tree})$ and $F_p(\text{Grid})$ alongside. The comparison with the Tree system is relevant because it is the only alternative

to CWlog when log-sized quorums are required. Comparison with the Grid is relevant since the Grid is sometimes proposed (cf. [20]) as a viable choice for small systems with reliable elements (small p), despite its poor asymptotic availability. Note that the figure shows the behavior of F_p itself (for all systems), not that of the bounds from Theorem 4.1.

Figure 4 reveals that the CWlog has excellent availability starting from $n = 1$. Both the CWlog and Tree systems have similar availability on comparable universe sizes, with a small advantage to the CWlog. For small values of p (e.g., $p \leq 0.1$) the failure probabilities are almost indistinguishable. However for $p = 0.3$ the CWlog has a better failure probability, especially when $n \geq 20$. The availability of the Grid system is much worse. For $p = 0.3$, the failure probability's increase towards 1 starts from $n = 2$. For $p = 0.1$, $F_p(\text{Grid})$ starts to increase beyond the range of the figure. However even in the shown range, there is virtually no gain in the Grid's availability when n passes $n = 16$, and $F_p(\text{CWlog})$ is always much better. We conclude that there is no reason to use the Grid system for practical systems, since its availability is inferior to both the CWlog and Tree systems for all n.

Note that the universe sizes required by the constructions rarely match. The Tree construction requires a universe size of $n = 2^h - 1$ for some h, and the Grid requires $n = d^2$ for some d. Therefore in the range $1 \leq n \leq 100$ there are only 6 fitting Tree sizes (and $\lfloor \lg n \rfloor$ sizes in general) and 10 fitting Grid sizes ($\lfloor \sqrt{n} \rfloor$ in general). In comparison the CWlog wall is more flexible, requiring $n = \sum_{i=1}^{d} \lfloor \lg 2i \rfloor$ for some d, so there are 25 fitting sizes in the range $1 \leq n \leq 100$ ($\approx n/\lg n$ universe sizes).

4.3 The Load of the CWlog

In this section we show that the load is $\mathcal{L}(\text{CWlog}) \approx \frac{1}{\lg n - \lg\lg n}$, which is optimal for a quorum system with such small quorums by Proposition 2.10. The upper bound is achieved by using strategy Pick(d) of Figure 2 (namely, using all the rows).

Proposition 4.3 $\frac{1}{\lfloor \lg 2d \rfloor} \leq \mathcal{L}(\text{CWlog}) < \frac{1}{\lfloor \lg 2d \rfloor} + \frac{1}{d}$.

Proof: The lower bound follows from Proposition 2.10 since $c = c(\text{CWlog}) = \lfloor \lg 2d \rfloor$. For the upper bound, note that Proposition 3.13 guarantees a bound of $2/c$ using the strategy $w_1 \equiv \text{Pick}(c)$, since CWlog is a normal wall (the critical row is $r = d$). However we can do better, by using strategy $w_2 \equiv \text{Pick}(d)$ (using all the rows). Following the same analysis of Proposition 3.14 we get that the load on an element u in row i is

$$\ell_u = \frac{1}{d}\left(1 + \frac{i-1}{n_i}\right).$$

For the CWlog this expression is maximal when $i = d$, and since n_d is the size of the smallest quorum c we obtain that

$$\mathcal{L}_{w_2}(\text{CWlog}) = \frac{1}{d}\left(1 + \frac{d-1}{c}\right) < \frac{1}{d} + \frac{1}{c}. \qquad \blacksquare$$

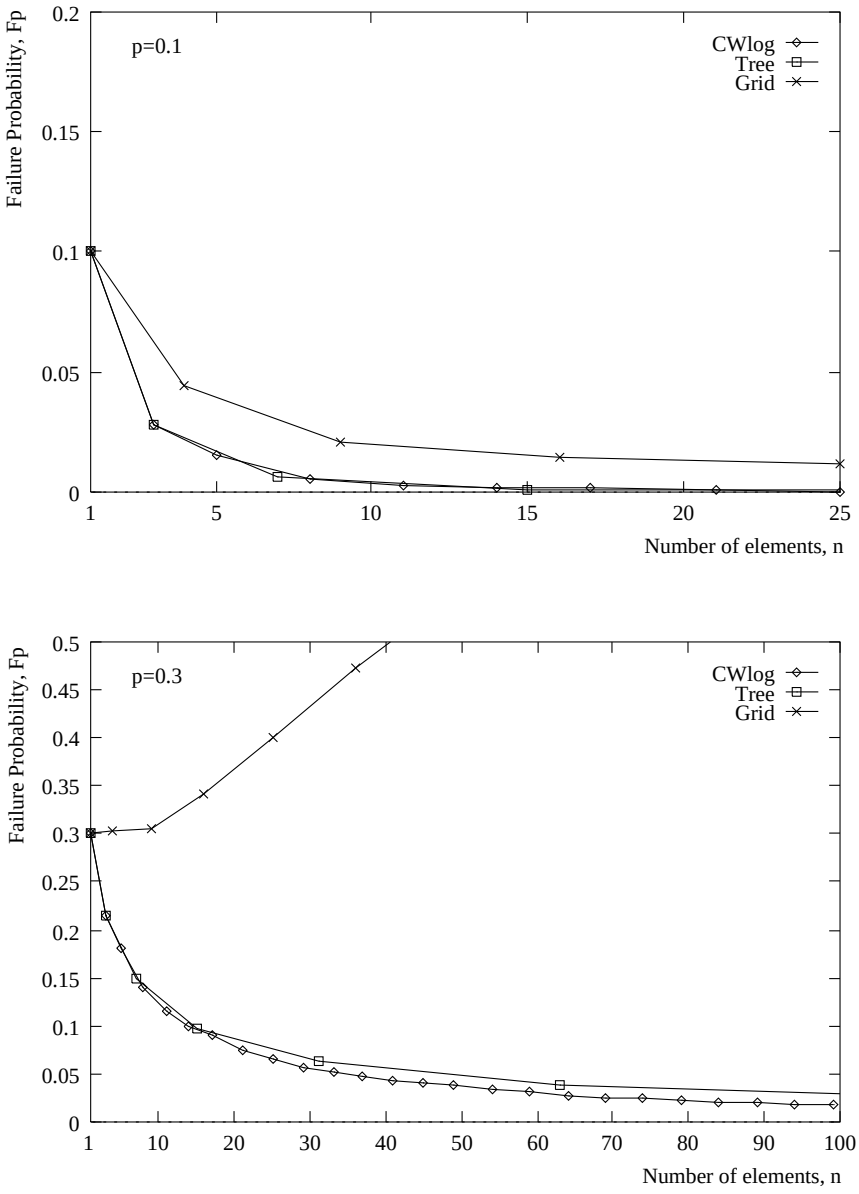

Figure 4: Comparison between the failure probabilities of the CWlog, Tree and Grid quorum systems as functions of the size of the universe n.

Remark: The strategy used in the proof is still not the best possible. For instance, using row 1 as a full row implies that one element from row d will also be used, but the reverse is not true, so the elements in row d are accessed at a higher rate than the element in row 1. This imbalance can be fixed using a more complicated strategy, that would slightly increase the probability of choosing top rows and decrease that of choosing bottom rows. Since the gap between our bounds is $1/d$ it is clear that such a modification would not change the load significantly.

4.4 Selecting a Quorum in CWlog

In this section we consider the question of which CWlog quorum to select whenever the protocol needs to access one. Two important (and conflicting) parameters that depend on the strategy we use are the size of the selected quorums and the load that is induced on the elements.

If the elements are fail free then the question is easy. If the quorum size is more important, then the trivial strategy that only uses the last row (the smallest quorum) is the best possible, but it induces a load of 1. If the load is more important, then the strategy of Proposition 4.3 is almost optimal, but it may return quorums of size $\Omega(n/\log n)$. A reasonable tradeoff is to use strategy Pick(c) (of Proposition 3.13), which induces a near optimal load of at most $2/c = 2/\lfloor \lg 2d \rfloor$ and returns quorums of size no larger than $2\lfloor \lg 2d \rfloor - 1$.

In the sequel we discuss the case where elements may fail. Then the question becomes more interesting for two reasons. First, the smaller quorums may be hit, so our goal becomes picking the smallest *live* quorum. Second, a tacit assumption in the definition of the load is that the structure of the system is *known* to the strategy and its choices are based on this structure. However when failures occur the system structure effectively changes, and this needs to be addressed by the strategy.

```
The rows are U_1, ..., U_d with |U_i| = n_i = ⌊lg 2i⌋.
Q ← ∅
for i = d to 1                    (* bottom to top *)
    if all n_i elements in U_i have failed then
        return ∅                  (* system failure *)
    else if all n_i elements in U_i are alive then
        return U_i ∪ Q            (* success *)
    else                          (* element u_i ∈ U_i is alive *)
        Q ← Q ∪ {u_i}
end-for
```

Figure 5: Procedure PickSmall

4.4.1 Minimizing the Quorum Size Under Failures

Procedure PickSmall (given in Figure 5) is designed to minimize the quorum size. It is an algorithmic version of the argument used in the calculation of the failure probability in Section 3.3.

Lemma 4.4 *Procedure* PickSmall *returns a valid quorum iff one exists in the current configuration.*

Proof: The procedure considers row $i-1$ only if row i has both a failed element and a live one. It collects a live representative of each row into the set Q until either all the rows were examined, or a fully live row was found. Note that since row 1 has a single element, the procedure will surely stop when $i = 1$; a row containing a single element must fall into one of the stopping cases. ∎

The following claim shows that the procedure manifests *graceful degradation.*

Proposition 4.5 *Procedure* PickSmall *returns a minimal sized quorum which is alive in the current configuration.*

Proof: In any configuration in which a live quorum exists, the size of the quorum is only dependent on the index of the full row. A quorum based on row i has a size of $n_i + (d - i) = \lfloor \lg 2i \rfloor + d - i$. This is clearly decreasing with i, therefore the smallest live quorums are based on the full row with largest index, which is precisely the choice made by PickSmall. ∎

Remark: PickSmall always accesses the elements of the bottom row, so if all the elements are alive then the induced load is 1.

1. The rows are $U_1, \ldots, U_d$ with $|U_i| = n_i = \lfloor \lg 2i \rfloor$.
2. Find i_f, the largest i such that all the elements in U_i have failed (set $i_f \leftarrow 0$ if no such U_i exists).
3. Find $i_1, \ldots, i_t$ such that $i_f < i_j \leq d$ and all the elements of U_{i_j} are alive for $j = 1, \ldots, t$. If no such U_{i_j} exists, then return $\varnothing$ (system failure).
4. Choose r uniformly at random in the range $\{1, \ldots, t\}$.
5. Set $Q \leftarrow \varnothing$. For $i = i_r + 1$ to d, pick an element at random from the live elements of U_i and add it to Q.
6. Return $U_{i_r} \cup Q$.

Figure 6: Procedure PickBalanced

4.4.2 Minimizing the Load Under Failures

Procedure PickBalanced (given in Figure 6) chooses quorums in a random fashion, so that the elements will be accessed at roughly the same rate. The procedure follows the proof of Proposition 4.3, taking failures into account.

Lemma 4.6 *Procedure* PickBalanced *returns a valid quorum iff one exists in the current configuration.*

Proof: A row i containing only failed elements disables the use of any quorum with a full row $j < i$. Therefore row i_f of step 2 in the procedure is the "roof" of the interesting rows of the current configuration. Thus the rows $i_1, \ldots, i_t$ of step 3 are the only candidates to be a full row in a quorum. Clearly, in a failure configuration the procedure will find no full rows $i_j > i_f$ (either $i_f = d$ and there are no rows to consider under the roof, or all the rows under the roof are hit). Hence the condition recognizing a system failure is correct. The actual choice of the quorum in steps 4 and 5 is trivially correct. ∎

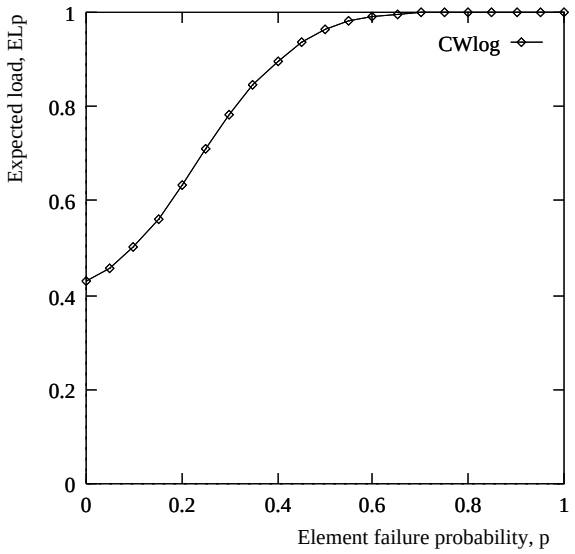

Figure 7: The expected load induced by procedure PickBalanced on the wall $\mathcal{S} = \mathrm{CW}\langle 1,2,2,3,3,3,3\rangle$ with 17 elements and 7 rows, as a function of the element failure probability p.

The expected load induced by PickBalanced is shown in Figure 7. When all the elements are alive, PickBalanced identifies with the strategy described in Proposition 4.3 so it induces an almost optimal load of $1/d + (d-1)/d\lfloor \lg 2d \rfloor$ (which is approximately 0.428 for the wall of Figure 7). When $p > \frac{1}{2}$ then with high probability there is no live quorum, since the CWlog has a Condorcet failure probability (Theorem 4.1 and Theorem 2.6). This is manifested by the load being ≈ 1 in this range. In Proposition 4.7 we show that the load is $O(1/\log n)$ as long as $p < 0.432$, and this behavior is achieved by a procedure that is essentially equivalent to

PickBalanced.

Note that the procedure requires knowledge of the global configuration before deciding which quorum to return. Therefore this approach is more useful in distributed systems in which the configuration of failed elements is known to the processor that is requesting the quorum. This knowledge means that all the computation described in the procedure can take place locally, without sending exploration messages to test the status of each element. Thus PickBalanced is appropriate in systems with broadcast communication capabilities in which the current configuration is available to the processors (e.g., the Transis system [2]), or in point-to-point systems in which the configuration changes are infrequent, where we can assume that the configuration is known for long periods of time.

Remark: The average quorum size that PickBalanced returns is $\approx d/2 + \lg d = O(n/\lg n)$.

4.5 The Load of CWlog in the Presence of Failures

In this section we consider the load in the presence of failures. The following proposition shows that asymptotically, with high probability the load is still $O(\frac{1}{\log n})$ as long as the failure probability is $0 \leq p < 0.432$. The strategy that achieves this performance is essentially procedure PickBalanced of Section 4.4.2, with minor modifications that simplify the analysis. Therefore CWlog can provably tolerate up to 43% failures, without degrading the load significantly. We believe that the true behavior is even better than proved, since in the proof we make several large over-estimates.

Proposition 4.7 *If* $0 \leq p < 0.432$ *then the load of* CWlog *is* $O(\frac{1}{\log n})$ *with probability* $\geq 1 - \left(\frac{\lg n}{n}\right)^{\varepsilon}$ *for some* $\varepsilon > 0$.

Proof: Let k denote the width of the bottom row, and assume that the last block of rows is full, i.e., the bottom 2^{k-1} rows all have width k. Let $a > 2$ be some constant (to be determined later).

Consider a row of width k and let $\#good$ count the number of live elements in it. Then $\mathbb{E}[\#good] = kq > k/2$ when $q > \frac{1}{2}$. Using the strong type of Chernoff bound for the binomial distribution (see [13]),

$$\mathbb{P}(\#good \geq \frac{k}{a}) \geq 1 - \left(\frac{kq}{k/a}\right)^{k/a} e^{k/a - kq} = 1 - \left[(qa)^{1/a} e^{1/a - q}\right]^k.$$

Let $\beta = (qa)^{1/a} e^{1/a-q}$. Then $\mathbb{P}(\#good \geq \frac{k}{a}) \geq 1 - \beta^k$.

Let $1 < r < \frac{1}{\beta}$. Let $\mathcal{E}_1$ be the event that the bottom $\lfloor r^k \rfloor$ rows have at least $\frac{k}{a}$ live elements in each. Then

$$\mathbb{P}(\mathcal{E}_1) \geq (1 - \beta^k)^{\lfloor r^k \rfloor} \geq 1 - (\beta r)^k. \tag{1}$$

Now let $\#full$ count the number of fully live rows among the bottom $\lfloor r^k \rfloor$ ones. Then $\mathbb{E}[\#full] = \lfloor r^k \rfloor q^k$. Let $\mathcal{E}_2$ be the event that $\#full \geq k$. If $q > 1/r$ then $\mathbb{E}[\#full]$ is exponential in k so

there certainly exists $\gamma > 0$ such that $\mathbb{P}(\mathcal{E}_2) \geq 1 - \gamma^k$. Combining with (1) we get

$$\mathbb{P}(\mathcal{E}_1 \wedge \mathcal{E}_2) \geq 1 - (\beta r)^k - \gamma^k \geq 1 - \left(\frac{\lg n}{n}\right)^{\varepsilon}$$

for some $\varepsilon > 0$, since $k \geq \lg(\frac{n}{\lg n})$.

So with high probability we have a configuration in which all $\lfloor r^k \rfloor$ bottom rows have at least $\frac{k}{a}$ live elements, and at least k of these rows have all their elements alive. In such a configuration, we can use the following strategy w: pick one of the available full rows with uniform probability of $\leq \frac{1}{k}$, and in each row below the full one pick a representative with uniform probability among its live elements. The maximal load is induced on the elements of the bottom row, when it is one of the partial rows. Let u be an element of the last row, then $\ell_w(u) \leq \frac{a}{k}$. We are finished, as long as there exist values q, a and r that fill the requirements that

$$\frac{1}{q} < r < \frac{1}{\beta} = (qa)^{-1/a} e^{-1/a+q}.$$

Taking $q > .568$ and $r = 1.762$ ensures the existence of a valid constant a. For example, if we consider only $q \geq 0.7$ and take $r = 1.429$, then $a = 8$ is valid. ∎

5 Conclusion

In the previous sections and in [30] we have analyzed the availability and load of general crumbling walls. We have also identified what we consider to be the best system within this class of quorum systems, the CWlog system, and analyzed it in detail. The CWlog system enjoys the following properties:

- Small (logarithmic) quorum size.
- High availability both for practical universe size and asymptotically.
- Flexible, fits many universe sizes.
- Provably optimal load and availability among systems with log-sized quorums.
- Both the returned quorum size and expected load degrade gracefully as failures occur.

Therefore we believe that the CWlog is a good candidate to be the system of choice when designing a distributed protocol which requires quorum systems.

Acknowledgment

We are grateful to Moni Naor for his contributions to our analysis of the load. We thank the anonymous referees for their remarks, which improved the presentation of the paper.

References

[1] 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.

[2] Y. Amir, D. Dolev, S. Kramer, and D. Malki. Transis: A communication subsystem for high availability. In *Proc. 22nd IEEE Symp. Fault-Tolerant Computing (FTCS)*, pages 76–84, 1992.

[3] 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.

[4] Y. Amir and A. Wool. Optimal availability quorum systems: Theory and practice. Technical Report CS96-02, The Weizmann Institute of Science, Rehovot, Israel, 1996.

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

[6] S. Y. Cheung, M. H. Ammar, and M. Ahamad. The grid protocol: A high performance scheme for maintaining replicated data. *IEEE Trans. Knowledge and Data Eng.*, 4(6):582–592, 1992.

[7] N. Condorcet. Essai sur l'application de l'analyse à la probabilité des decisions rendues à la pluralite des voix. Paris, 1785.

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

[9] K. Diks, E. Kranakis, D. Krizanc, B. Mans, and A. Pelc. Optimal coteries and voting schemes. *Inf. Proc. Letters*, 51:1–6, 1994.

[10] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and Finite Sets*, pages 609–627. Colloq. Math. Soc. János Bolyai 10, 1975.

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

[12] D. K. Gifford. Weighted voting for replicated data. In *Proc. 7th Symp. Oper. Sys. Princip.*, pages 150–159, 1979.

[13] T. Hagerup and C. Rüb. A guided tour of Chernoff bounds. *Inf. Proc. Letters*, 33:305–308, 1990.

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

[15] R. Holzman, Y. Marcus, and D. Peleg. Load balancing in quorum systems. In *Proc. 4th Workshop on Algorithms and Data Structures*, pages 38–49, Kingston, Ont., Canada, 1995. To appear in SIAM J. Discrete Math.

[16] T. Ibaraki and T. Kameda. A theory of coteries: Mutual exclusion in distributed systems. *IEEE Trans. Par. Dist. Sys.*, 4(7):779–794, 1993.

[17] D. E. Knuth. *The Art of Computer Programming, Vol. 1—Fundamental Algorithms.* Addison-Wesley, 1968.

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

[19] A. Kumar and S. Y. Cheung. A high availability $\sqrt{n}$ hierarchical grid algorithm for replicated data. *Inf. Proc. Letters*, 40:311–316, 1991.

[20] A. Kumar, M. Rabinovich, and R. K. Sinha. A performance study of general grid structures for replicated data. In *Proc. 13th Inter. Conf. Dist. Comp. Sys.*, pages 178–185, 1993.

[21] L. Lovász. Coverings and colorings of hypergraphs. In *Proc. 4th Southeastern Conf. Combinatorics, Graph Theory and Computing*, pages 3–12, 1973.

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

[23] Y. Marcus and D. Peleg. Construction methods for quorum systems. Technical Report CS92–33, The Weizmann Institute of Science, Rehovot, Israel, 1992.

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

[25] 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.

[26] 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`.

[27] M. L. Neilsen. *Quorum Structures in Distributed Systems.* PhD thesis, Dept. Computing and Information Sciences, Kansas State University, 1992.

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

[29] D. Peleg and A. Wool. Crumbling walls: A class of practical and efficient quorum systems. In *Proc. 14th ACM Symp. Princip. Distributed Computing (PODC)*, pages 120–129, Ottawa, Canada, 1995.

[30] D. Peleg and A. Wool. The availability of crumbling wall quorum systems. *Discrete Applied Math.*, 1996. To appear.

[31] D. Peleg and A. Wool. How to be an efficient snoop, or the probe complexity of quorum systems. In *Proc. 15th ACM Symp. Princip. Distributed Computing (PODC)*, pages 290–299, Philadelphia, 1996.

[32] S. Rangarajan, S. Setia, and S. K. Tripathi. A fault-tolerant algorithm for replicated data management. In *Proc. 8th IEEE Int. Conf. Data Engineering*, pages 230–237, 1992.

[33] S. Rangarajan and S. K. Tripathi. A robust distributed mutual exclusion algorithm. In *Proc. 5th Inter. Workshop on Dist. Algorithms (WDAG), LNCS 579*, pages 295–308. Springer-Verlag, 1991.

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

[35] 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.

[36] R. H. Thomas. A majority concensus approach to concurrency control for multiple copy databases. *ACM Trans. Database Sys.*, 4(2):180–209, 1979.

[37] 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.