An Arbitrary 2D Structured Replica Control Protocol

Robert Basmadjian and Hermann de Meer

University of Passau, Department of Computer Network and Communication
Innstrasse 43, 94032, Passau, Germany
{basmadji,demeer}@fim.uni-passau.de

Abstract

Traditional replication protocols that logically arrange the replicas into a specific structure have reasonable availability, lower communication cost as well as system load than those that do not require any logical organisation of replicas.

We propose in this paper the A2DS protocol: a single protocol that, unlike the existing proposed protocols, can be adapted to any 2D structure. Its read operation is carried out on any replica of every level of the structure whereas write operations are performed on all replicas of a single level of the structure. We present several basic 2D structures and introduce the new idea of obtaining other 2D structures by the composition of several basic ones.

Two structures are proposed that have near optimal performance in terms of the communication cost, availability and system load of their read and write operations. Also, we introduce a new protocol that provides better performance for its write operations than those of *ROWA* protocol while preserving similar read performance.

1998 ACM Subject Classification Fault tolerance, Distributed systems.

Keywords and phrases Replication, Performance attributes, Reliability, Availability, Load.

Digital Object Identifier 10.4230/OASIcs.KiVS.2011.157

1 Introduction

In large distributed systems, data is replicated to provide a high level of *fault-tolerance* and to improve the *system performance*. However when replication is used, data becomes susceptible to inconsistency problems. Therefore, a replica control protocol (RCP) is required to "synchronize"concurrent read (query) and write (update) operations on replicated data. To ensure *one-copy equivalence*[1], a read and a write operation to two different replicas of the same data should not be allowed to execute concurrently. *Quorum systems*[2] are used by these protocols which serve as a basic tool of achieving one-copy equivalence.

Given the importance of the topic, several replication protocols have been described in the literature [1 – 11]. They differ according to various parameters such as the number of replicas involved in a given operation (henceforth referred to as their *communication cost* and ranges between 1 and total number of replicas n), their ability to tolerate replica failures (also termed as their *availability* and ranges between 0 and 1), as well as the *load* (ranges between 0 and 1) they impose on the system. Also, these protocols can be classified into two categories: those that arrange logically replicas of the system into a particular structure

[1] The fact of existing several replicas of a data should be abstracted as if there exists only a single replica.
[2] A quorum system is defined as a set of subsets of replicas called quorums having pair-wise non-empty intersections.

17th GI/ITG Conference on Communication in Distributed Systems (KiVS'11).
Editors: Norbert Luttenberger, Hagen Peters; pp. 157–168
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

[3 – 11] henceforth called structured RCPs, and those that do not impose any structure on the replicas [1, 2] which we call non-structured RCPs. The difference between the two categories is that the former has lower communication cost and system load than the latter, whereas the latter has better availability than the former for read and write operations.

The motivation of this paper is to ask whether it is possible to provide a single protocol that can be applied to any 2D structure and to afford optimal performance in terms of the communication cost, availability and system load of its read and write operations.

We answer with the affirmative by proposing the A2DS protocol: a protocol which assumes that replicas are logically arranged into any 2D structure based on its width w and height h. Read operations are carried out on any single replica at every level of the structure whereas write operations are performed on all replicas of any single level of the structure. We provide several basic 2D structures and give each one's performance in terms of the communication cost, availability and system load of its read and write operations. We also introduce the new idea of obtaining other 2D structures by the composition of several basic ones. For *mostly-read* systems, we propose a structure that has the same performance for its read and write operations as *ROWA* protocol [1]. When read and write operations happen in *equiprobable frequencies*, we introduce two structures that provide near optimal performance for both operations. Among the structured RCPs, the former has the best combined read and write costs of $2\sqrt{n}$ and its operations induce the best combined loads of $\frac{2}{\sqrt{n}}$, whereas the latter has a cost of w for its write operations which is lower than those of the existing structured RCPs and its operations induce near best combined loads of $\mathcal{O}(\frac{2}{\sqrt{n}})$; yet preserving comparable availability for its read and write operations. Finally, we propose a new protocol that we call ***R**ead**T**wo **W**rite**M**ajority* which provides better performance for its write operations than those of the well known *ROWA* protocol [1] while still maintaining similar performance for its read operations. The rest of the paper is organized as follows: Section 2 provides a brief overview of the related work. Our protocol is introduced in section 3. We give a general comparison among the structured RCPs in section 4. Section 5 shows the conclusion.

2 Related Work

For a system of n replicas, the well known *ReadOneWriteAll* (*ROWA*) protocol [1] has a read cost of 1 and a write cost of n. Its read operations are highly fault-tolerant (an availability of 1) and induce on the system a load of $\frac{1}{n}$. On the other hand, write operations have a very low availability (reaches to 0) as an update cannot be performed in the presence of a single replica failure or network partitions; they impose to the system the highest load of 1. The *Majority Quorum Consensus* (*MQC*) protocol [2] has read and write communication costs of $\frac{n+1}{2}$ for an odd-sized number of replicas n and imposes to the system a load of at least 0.5. It tolerates replica failures for read and write operations at the expense of increased read costs with respect to those of *ROWA*. However, both *ROWA* and *MQC* protocols have a high communication cost as well as system load, and are classified as non-structured RCPs.

By arranging replicas of the system into a logical structure, it is possible to reduce the communication cost as well as system load further. Several protocols have been proposed in the literature which make use of quorum systems and assume that the replicas are organized logically into a specific structure: *finite projective plane* [3] , a *grid structure* [4] and [5], or a *tree structure* [6], [7], [8], [9], and [10]. The load of these protocols was studied in [5] and it was proven that for a system of n replicas, the least induced load is $\frac{1}{\sqrt{n}}$ for any operation.

For a system of $n = t^2 + t + 1$ replicas ($t > 0$), the Finite Projective Plane (FPP) protocol

[3] has read and write costs of $\frac{1+\sqrt{4n-3}}{2}$ and induces on the system a load of $\frac{1+\sqrt{4n-3}}{2n}$. The major drawback of this protocol is that when the number of replicas $n > 100$, the availability of its read and write operations deteriorates gradually as it was shown in [12].

The Grid Quorum (GQ) protocol [4], for a system of n replicas, has a read cost of $\sqrt{n}$ and a write cost of $2\sqrt{n}-1$. Its read and write operations are fault-tolerant and induce on the system a load of $\frac{1}{\sqrt{n}}$ and $\frac{2\sqrt{n}-1}{n}$ respectively. Note that all these results are based on the fact that replicas are arranged logically into a square grid. For a system of $n = 2d^2 + 2d + 1$ replicas ($d > 0$) arranged logically into a percolation grid, the read and write operations of the Paths Quorum System (PQS) protocol [5] have a minimum communication cost of $\sqrt{2n-1}$, are highly available, and induce on the system a load of $\frac{\sqrt{2n-1}}{2n}$.

In general, the tree-structured RCPs have a tight trade-off between the communication cost and the load induced by their operations: a low cost results in inducing a high load and vice versa. In [11], the *Arbitrary Tree* protocol was introduced and it was shown that its write operations only induce on the system a load of $\frac{1}{\sqrt{n}}$ with a cost of $\sqrt{n}$, which is lower than state-of-the-art tree-structured RCPs, while preserving comparable write availability. On the other hand, its read operations only induce a cost of $\sqrt{n}$ which is lower than previously proposed tree-structured RCPs with comparable load and availability. In this paper, we adopt load related definitions, notations and propositions of section 2.1 of [11] as well as the system model of section 2.2 of [11].

3 Our Protocol

Given a *replication-based* system of n replicas, we organize them logically into any 2-dimensional structure of height $h > 0$. More precisely, let $R(i,k)$ denote the i^{th} replica of the k^{th} level of the 2D structure where the orientation is taken from left to right and top to bottom respectively such that $i \in [1, m_k]$ and $k \in [0, h]$, where m_k denotes the total number of replicas at level k.

3.1 The operations

We construct the set of read quorums $\boldsymbol{R}$ and write quorums $\boldsymbol{W}$ by respecting the definition 2.3 of [11] on *Bi-coteries*. Furthermore, we assign separate strategies of picking read and write quorums using the definition 2.4 of [11] on *strategies*. More precisely, let w_{read} and w_{write} denote a strategy for picking read quorums of $\boldsymbol{R}$ and write quorums of $\boldsymbol{W}$ respectively where w_{read} and w_{write} are defined in subsequent sections. The availability computations are carried out by taking the assumption that every replica is *independently available* with a probability $p = 1 - q > \frac{1}{2}$, where $q \in [0, \frac{1}{2}[$ denotes the failure probability. In the rest of this section, we use $h > 0$ to denote the height of the structure, whereas we use d and e to denote respectively the *minimal* and *maximal* number of replicas among the levels of the 2D structure such that $d = \min\{m_k \mid \forall k : 0 \leq k \leq h\}$ and $e = \max\{m_k \mid \forall k : 0 \leq k \leq h\}$.

3.1.1 Reads

A read operation takes place by accessing all the members (replicas) of a read quorum $R_j \in \boldsymbol{R}$ and retrieving the data of the replica whose timestamp has the highest *version number*. In case several members of such a quorum R_j have the same highest version number, then the data of the replica (among replicas with highest version number) whose timestamp has the smallest *identifier* (*RID*) is fetched. A read quorum R_j is constructed by having as

its members any single replica of every level of the 2D structure:

$$R_j = \{R(i,k) \mid \forall k : 0 \leq k \leq h \ \wedge \ \exists i : 1 \leq i \leq m_k\}$$

▶ FACT 3.1.1. Let $\boldsymbol{R}$= $\{R_1, R_2, \ldots, R_j\}$ be the set of read quorums such that every read quorum R_j is constructed as explained above. Then the number of read quorums (size) of $\boldsymbol{R}$ is denoted by $m(\boldsymbol{R}) = \prod_{k=0}^{h} m_k$.

In order to compute the load of the system induced by this read operation, a strategy $w_{read} = \sum_{j=1}^{m(\boldsymbol{R})} w_{read,j} = 1$ is taken that picks each read quorum $R_j \in \boldsymbol{R}$ with a probability of $w_{read,j} = \frac{1}{m(\boldsymbol{R})}$ where $j \in \{1, \ldots, m(\boldsymbol{R})\}$. The read operation of our protocol has:

$$\text{A communication cost of } RD_{cost} = 1 + h$$

$$\text{An availability of } RD_{av}\ (p) = \prod_{k=0}^{h}(1-(1-p)^{m_k}) \tag{3.1}$$

$$\text{An optimal system load of } \mathcal{L}_{RD} = \frac{1}{d} \tag{3.2}$$

The proof of the *optimality* of the system load can be found in the *Appendix* of [11].

3.1.2 Writes

A write operation, after retrieving (from the replicas of a read quorum) the highest version number of the data to be modified and then incrementing it by one, accesses all the members of a write quorum $W_j \in \boldsymbol{W}$ in order to update their data with a new value and timestamp (the new *version number* along with the write quorum's first replica's *identifier*).

A write quorum W_j is constructed by taking as its members all replicas of any single level of the 2D structure:

$$W_j = \{R(i,k) \mid \exists k : 0 \leq k \leq h \ \wedge \ \forall i : 1 \leq i \leq m_k\}$$

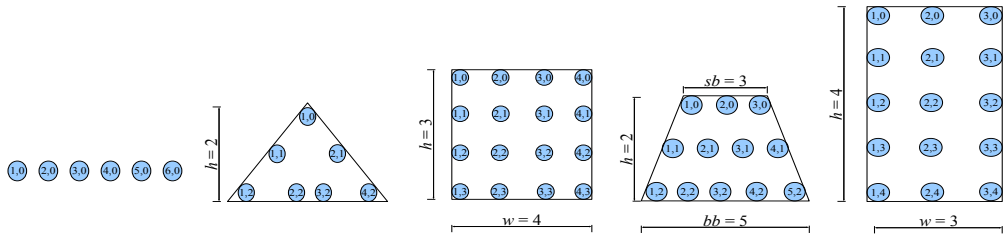
▶ FACT 3.1.2. Let $\boldsymbol{W} = \{W_1, W_2, \ldots, W_j\}$ be the set of write quorums such that every write quorum W_j is constructed as explained above. Then the number of write quorums (size) of $\boldsymbol{W}$ is denoted by $m(\boldsymbol{W}) = 1 + h$.

In order to compute the load of the system induced by this write operation, a strategy $w_{write} = \sum_{j=1}^{m(\boldsymbol{W})} w_{write,j} = 1$ is taken that picks each write quorum $W_j \in \boldsymbol{W}$ with a probability of $w_{write,j} = \frac{1}{m(\boldsymbol{W})}$ where $j \in \{1, \ldots, m(\boldsymbol{W})\}$. The write operation of our protocol has a minimum communication cost of d, a maximum cost of e and an *average* cost of $WR_{cost} = \sum_{k=0}^{h} m_k \times w_{write,k}$. Hence such a strategy w_{write} of picking write quorums induces a communication cost of $\frac{n}{1+h}$. This operation has an availability of:

$$WR_{av}\ (p) = 1 - WR_{fail}\ (p) \tag{3.3}$$

where $WR_{fail}\ (p) = \prod_{k=0}^{h}(1-p^{m_k})$ and it imposes an optimal system load of:

$$\mathcal{L}_{WR} = \frac{1}{1+h} \tag{3.4}$$



(a) Line **(b)** Triangle **(c)** Square **(d)** Trapezoid **(e)** Rectangle

Figure 1 An example of our basic structures for $n = 6, 7, 16, 12$ and 15 replicas respectively.

The proof of the *optimality* of the system load can be found in the *Appendix* of [11].

3.1.3 The expected system load computations

The loads imposed by read and write operations of (3.2) and (3.4) respectively are computed by assuming that all replicas of the system are fail-free. The load of the system induced by these operations becomes higher as replicas of the system start to fail one after another. In order to compute the *expected load* assuming that replicas are available with a probability $p > \frac{1}{2}$, we use the following two equations expressed in terms of (3.1), (3.2), (3.3), and (3.4):

$$\mathbb{E}\mathcal{L}_{RD} = RD_{av}(p) \times (\mathcal{L}_{RD} - 1) + 1 \quad (3.5)$$

$$\mathbb{E}\mathcal{L}_{WR} = WR_{av}(p) \times \mathcal{L}_{WR} + WR_{fail}(p) \times 1 \quad (3.6)$$

We can notice from (3.5) and (3.6) that the expected load computations largely depend on the availability of the operations: as the availability of the operations is high, the *expected load* becomes close to the *fail-free load* and thus the system is classified as *stable*.

3.1.4 The intersection of read and write quorums

In this section, we demonstrate that our system is a *bi-coterie* i.e. any read quorum $R_j \in \boldsymbol{R}$ has a *non empty intersection* with any write quorum $W_j \in \boldsymbol{W}$. The proof is by induction on the height h of the structure:
Basis Step: Trivial for a structure of height $h = 0$, because all replicas are placed at one and only one level.
Induction Hypothesis: Assume that it holds true for a structure of height $h > 0$.
Induction Step: Consider a 2D structure of height $h' = h+1$. Since any read quorum $R_j \in \boldsymbol{R}$ intersects with any write quorum $W_k \in \boldsymbol{W}$ such that $0 \leq k \leq h$ (*induction hypothesis* step), then it holds true because the fact of adding one new level $i = h + 1$ does not prevent any read quorum $R_j \in \boldsymbol{R}$ to have a non-empty intersection with any write quorum $W_k \in \boldsymbol{W}$ such that $0 \leq k \leq h$. On the other hand, since any read quorum $R_j \in \boldsymbol{R}$ contains a replica from the *new level* i and the *new write quorum* W_i contains all replicas of the level i, then any read quorum $R_j \in \boldsymbol{R}$ has a non-empty intersection with this write quorum W_i. Hence by induction, our protocol guarantees *non-empty intersection* of read and write quorums.

3.2 Basic structures

In this section, we introduce various basic 2D structures and give each one's characteristics in terms of the communication cost, availability and load of its read and write operations.

3.2.1 Straight line

This is a special case of our 2D structures ($h = 0$) where n replicas of the system are arranged logically into a *straight line* (see Figure 1(a)). Its read operation has a cost of 1, an availability of $1-(1-p)^n$ and induces on the system a load of $\frac{1}{n}$. The write operation of this structure has a cost of n, an availability of p^n and imposes to the system a load of 1. Note that such a structure has the same characteristics of *ROWA* [1] and therefore is most appropriate for *mostly-read* systems because it favors read operations over write ones.

3.2.2 Triangle

The illustrated structure of Figure 1(b) is constructed by arranging logically $n = 2^{h+1} - 1$ replicas into a *triangle* of height $h > 0$ such that $m_k = 2^k$ at every level $k \in [0, h]$. Its read operation has a cost of $\log(n+1)$, an availability of $\prod_{k=0}^{h}(1-(1-p)^{2^k})$ and imposes on the system a load of 1. The write operation has a cost of $\frac{n}{\log(n+1)}$, an availability of $1-\prod_{k=0}^{h}(1-p^{2^k})$ and induces on the system a load of $\frac{1}{\log(n+1)}$. The major drawback of this structure is that its read operations always induce the highest load of 1 and these operations are poorly available i.e. if the replica at level 0 fails, then no read operations can take place.

3.2.3 Square

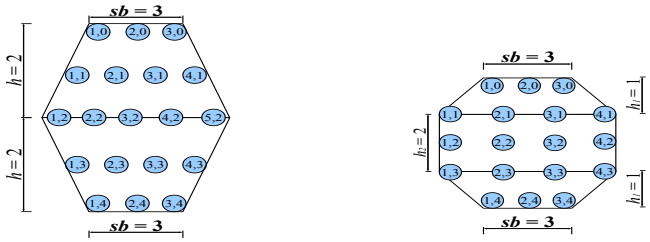
The replicas of this structure (see Figure 1(c)) are arranged logically into a *square* of height $h > 0$ and width $w = h + 1$ such that the number of replicas $n = w \times (h + 1)$. Its operations have a cost of $\sqrt{n}$, and induce on the system a load of $\frac{1}{\sqrt{n}}$. The read and write operations have an availability of $(1-(1-p)^{\sqrt{n}})^{\sqrt{n}}$ and $1-(1-p^{\sqrt{n}})^{\sqrt{n}}$ respectively. The major drawback of this structure is that when $n > 25$, its write operations become poorly available. Thus, such a structure is most appropriate for systems having highly available replicas: $p \geq 0.90$.

3.2.4 Trapezoid

The structure of Figure 1(d) is obtained by arranging logically $n = sb \times (1 + h + \sum_{i=1}^{h} \frac{i}{sb})$ replicas into a trapezoid of height $h > 0$, small base $sb > 1$ and big base $bb = sb + h$ such that $m_k = sb + k$ at every level $k \in [0, h]$. Its read operation has a cost of $\frac{-2sb+1+\sqrt{4sb^2-4sb+1+8n}}{2}$, an availability of $\prod_{k=0}^{h}(1-(1-p)^{(sb+k)})$, and imposes to the system a load of $\frac{1}{sb}$. The write operation of such a structure has a cost of $\frac{2sb-1+\sqrt{4sb^2-4sb+1+8n}}{4}$, an availability of $1-\prod_{k=0}^{h}(1-p^{(sb+k)})$, and induces a system load of $\frac{2sb-1+\sqrt{4sb^2-4sb+1+8n}}{4n}$. In order to obtain satisfactory results for both read and write operations at the same time, we fix $sb = 2$.

3.2.5 Rectangle

The replicas of this structure (see Figure 1(e)) are organized logically into a *rectangle* of height $h > 0$ and width $w > 1$ such that the number of replicas $n = w \times (h + 1)$. The read operation of this structure has a cost of $\frac{n}{w}$, an availability of $(1-(1-p)^w)^{h+1}$, and induces a



(a) Hexagon **(b)** Octagon

Figure 2 An example of our composed structures for $n = 19$ and 18 replicas respectively.

system load of $\frac{1}{w}$. Its write operation has a cost of w, an availability of $1-(1-p^w)^{h+1}$, and imposes to the system a load of $\frac{w}{n}$. To obtain satisfactory results for both operations, we set $h > w$ such that $w = 3$ for $15 \leq n \leq 21$, $w = 4$ for $24 \leq n \leq 48$, $w = 5$ for $50 \leq n \leq 85$, $w = 6$ for $96 \leq n \leq 152$, $w = 7$ for $154 \leq n < 252$ and $w = 8$ for $256 \leq n < 408$.

3.3 Composed structures

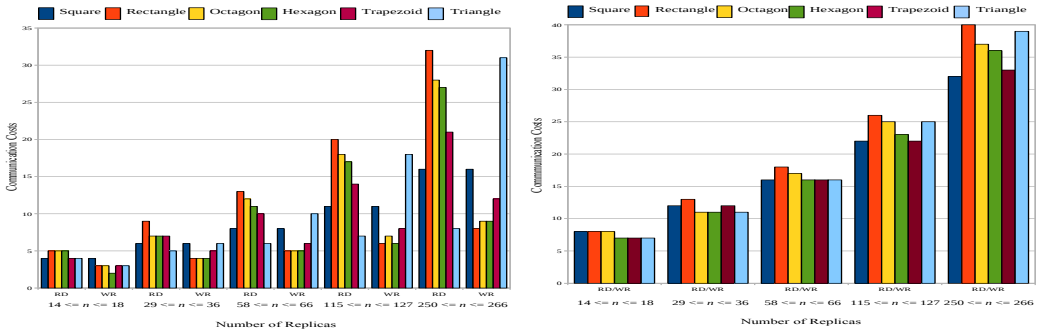
In this section, we introduce two composed structures and give the communication cost, availability and system load of their read and write operations.

3.3.1 Hexagon

The replicas of this structure (see Figure 2(a)) are arranged logically into a *hexagon* composed of two symmetric trapezoids, each of height $h > 0$ and small base $sb > 1$, joined at their corresponding big bases $bb = sb + h$ such that $n = h \times (2sb-1) + sb + 2 \times \sum_{i=1}^{h} i$. The read operation of this structure has a cost of $1-2sb+2\sqrt{sb^2-sb+n}$, an availability of $\left(\prod_{k=0}^{h-1}(1-(1-p)^{(sb+k)})\right)^2 \times (1-(1-p)^{bb})$, and induces on the system a load of $\frac{1}{sb}$. Write operations have a cost of $\frac{n\times(2\sqrt{sb^2-sb+n}+2sb-1)}{4n-1}$, an availability of $1-\left(\prod_{k=0}^{h-1}(1-p^{(sb+k)})\right)^2 \times (1-p^{bb})$, and imposes a system load of $\frac{2\sqrt{sb^2-sb+n}+2sb-1}{4n-1}$. In order to obtain satisfactory results for both operations, we set the small base $sb = 3$ for both trapezoids whenever $n \geq 30$.

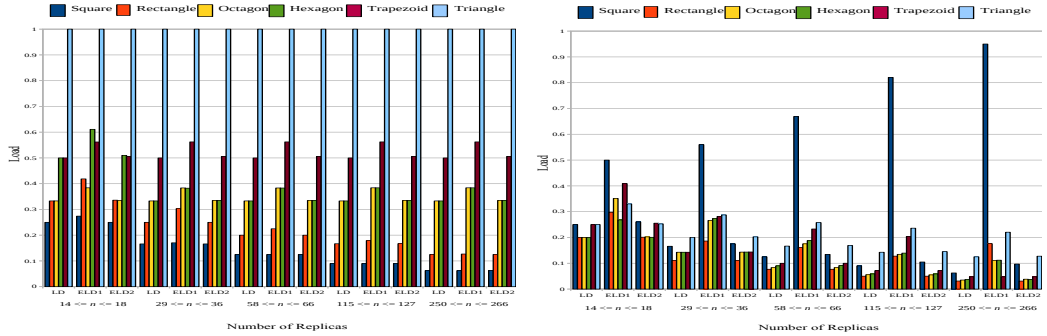
3.3.2 Octagon

This structure is obtained by organizing logically $n = sb\times(1+2h_1+h_2)+h_1\times(h_2-1)+2\times\sum_{i=1}^{h_1} i$ replicas into an *octagon* composed of two symmetric trapezoids, each of height $h_1 > 0$, small base $sb > 1$ and big base $bb = sb+h_1$, which are joined to each other by means of a rectangle of height $h_2 > 0$ and width $w = bb$ (see Figure 2(b)). The read operation of such a structure has a cost of $\frac{n-h_1\times(h_1+h_2)}{sb}$, an availability of $\left(\prod_{k=0}^{h_1-1}(1-(1-p)^{(sb+k)})\right)^2 \times \prod_{k=0}^{h_2}(1-(1-p)^{bb})$, and imposes a system load of $\frac{1}{sb}$. Its write operation has a cost of $\frac{n\times sb}{n-h_1\times(h_1+h_2)}$, an availability



(a) Separate costs **(b)** Combined costs

Figure 3 The communication costs of read (RD) and write (WR) operations.



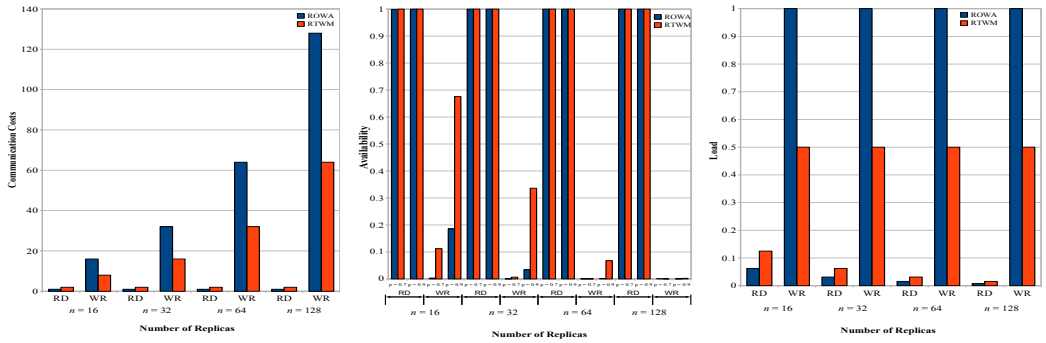
(a) Read operation **(b)** Write operation

Figure 4 The system (LD) and expected system (ELD) loads for $p = 0.7$ and $p = 0.9$.

of $1 - \left(\prod_{k=0}^{h_1-1} (1-p^{(sb+k)}) \right)^2 \times \prod_{k=0}^{h_2} (1-p^{bb})$, and induces on the system a load of $\frac{sb}{n-h_1\times(h_1+h_2)}$. To obtain satisfactory results for both operations, we fix $sb = 3$.

3.4 Comparison

When the number of replicas $n > 20$ (see Figure 3(a)), the *Rectangle* structure of width w has the highest cost of $\frac{n}{w}$ for read operations and the least cost of w for write operations. On the other hand, the *Triangle* structure has the fewest cost of $\log(n+1)$ for read operations and the worst cost of $\frac{n}{\log(n+1)}$ for write operations. Figure 3(b) demonstrates that the structures have quite comparable combined read and write costs. However when $n > 200$, the difference in combined costs of *Rectangle*, *Octagon*, *Hexagon* and *Triangle* structures becomes evident with respect to those of *Square* and *Trapezoid*. The expected system load computations of Figure 4 are carried out by using (3.5) and (3.6) such that we set $p = 0.7$ for *ELD1* and $p = 0.9$ for *ELD2*. We can observe in Figure 4(a) that the *Square* structure has the least system load of $\frac{1}{\sqrt{n}}$ for read operations and such a load diminishes as the number of replicas n increases. Also, the expected system loads of this structure are close to the fail-free system loads due to the high availability of its read operations. The *Triangle* structure has the highest fail-free and expected system loads of 1 due to the fact that the single replica at level 0 participates in every read operation. The *Rectangle* structure (see Figure 4(b)) has



(a) Communication cost **(b)** Availability **(c)** Load

Figure 5 The *RTWM* vs *ROWA* protocols.

the least system load of $\frac{w}{n}$ for write operations. Also, this structure has the least expected system loads either when $p = 0.7$ or $p = 0.9$ except for certain cases. The *Triangle* structure has the highest system load of $\frac{1}{\log(n+1)}$, as well as the highest expected system load when $p = 0.9$ for any number of replicas n. Note that, the *Square* structure has the worst expected system loads when $p = 0.7$ due to the low availability of its write operations.

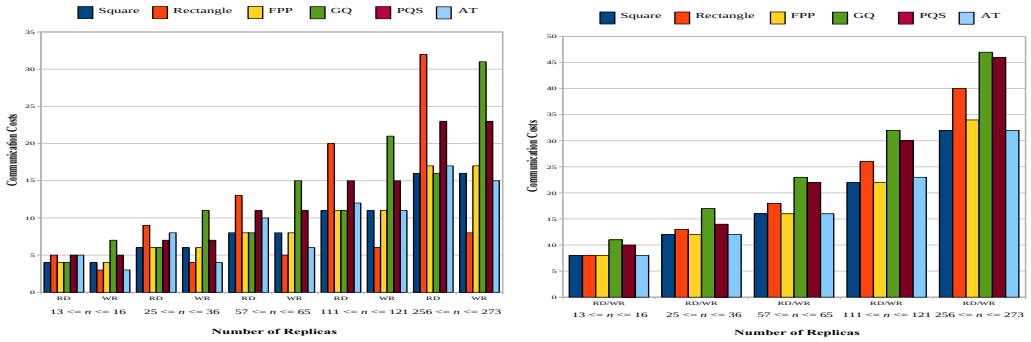
3.5 The RTWM protocol

In order to circumvent the drawbacks (see section 2) of write operations of the *ROWA* [1] protocol and yet to preserve the advantages of its read operations, we propose a protocol which we call *ReadTwoWriteMajority*. More precisely, for an even-sized number of replicas n, we arrange the replicas logically into the *Rectangle* structure of section 3.2.5 such that $w = \frac{n}{2}$ and $h = 1$. For an odd-sized n, we organize the replicas logically into the *Trapezoid* structure of section 3.2.4 such that $sb = \frac{n-1}{2}$, $h = 1$ and $bb = \frac{n+1}{2}$. The read operation of the *RTWM* protocol has a cost of 2, an availability of $\left(1 - (1-p)^{\frac{n}{2}}\right)^2$ if n is even, otherwise $\prod_{k=0}^{1}(1-(1-p)^{(sb+k)})$, and induces a system load of $\frac{2}{n}$ if n is even, otherwise $\frac{2}{n-1}$. On the other hand, its write operation has a cost of $\frac{n}{2}$, an availability of $1 - \left(1 - p^{\frac{n}{2}}\right)^2$ if n is even, otherwise $1 - \prod_{k=0}^{1}(1-p^{(sb+k)})$, and imposes a system load of $\frac{1}{2}$.

The *RTWM* protocol (see Figure 5) has much fewer communication cost (half), better availability and smaller system load (half) for its write operations than those of the *ROWA* [1] protocol, still preserving the advantages of read operations of this latter where our protocol has one communication cost higher than that of *ROWA*, and has comparable availability and system load especially for large number of replicas n.

4 General Comparison

In this section, we provide a general comparison among the *most relevant* existing structured as well as our 2D structured RCPs in terms of the communication cost, availability and system load of their operations. The "Square" configuration is set up based on the structure of section 3.2.3 such that $h = 3, 5, 7, 10$, and 15. The "Rectangle" configuration is studied using



(a) Separate costs **(b)** Combined costs

Figure 6 The communication costs of read (RD) and write (WR) operations.

the structure of section 3.2.5 such that $h = 4, 8, 12, 19$, and 31. The "FPP" configuration is examined by taking the protocol of [3] such that the number of replicas $n = t^2 + t + 1$ where $t = 3, 5, 7, 10$ and 16. The "GQ" configuration is considered by studying the protocol of [4] such that $n = 16, 36, 64, 121$, and 256 replicas. The "PQS" configuration is set up based on the protocol of [5] such that $n = 2d^2 + 2d + 1$ where $d = 2, 3, 5, 7$, and 11. Finally, the "AT" configuration is studied by considering the Algorithm 1 of [11] such that $n = 15, 31, 63, 127$, and 255 replicas.

4.1 Communication cost

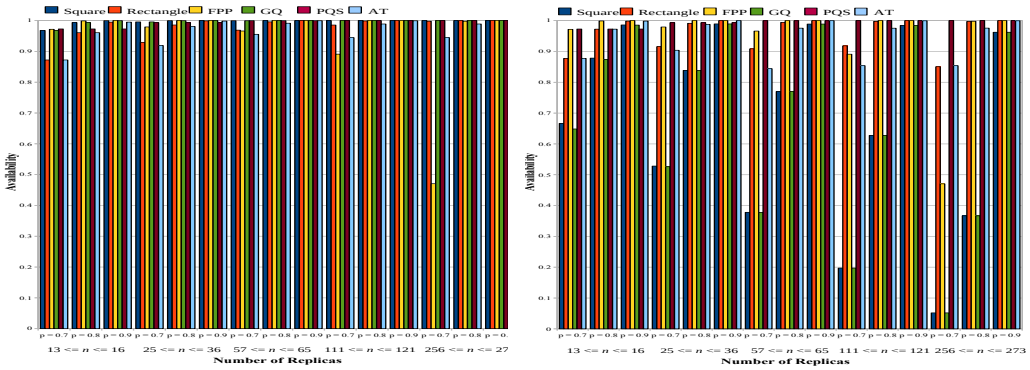
The "Rectangle" of width w has the highest cost of $\frac{n}{w}$ for read operations and the least cost of w for write ones (see Figure 6(a)). The configurations "Square", "FPP", and "GQ" have the fewest cost of $\sqrt{n}$ for read operations whereas "GQ" has the worst cost of $2\sqrt{n} - 1$ for write operations. We can observe in Figure 6(b) that the configurations "Square", "Rectangle", "FPP" and "AT" have quite comparable combined read and write costs. Also, "GQ" and "PQS" have the highest combined costs and that the difference in their combined costs becomes evident with respect to those of the other configurations when $n > 100$.

4.2 Availability

All the configurations have similar availability for read operations when $p = 0.8$ and 0.9 (see Figure 7(a)). When $p = 0.7$, the configurations "Square", "GQ" and "PQS" have quite better read availability than "FPP" and "AT". Note that the read availability of "Rectangle" ameliorates gradually as the number of replicas n increases. Figure 7(b) illustrates that all the configurations have similar write availability when $p = 0.8$ and 0.9 except for "Square" and "GQ". Both of these configurations have the worst write availability which degrades gradually with increasing n. "PQS" has the best write availability when $p = 0.7$ especially for large number of replicas n. Note that the read and write availability of "FPP" degrade gradually when $n > 100$ and $p = 0.7$ due to the non-Condorcet property as shown in [12].

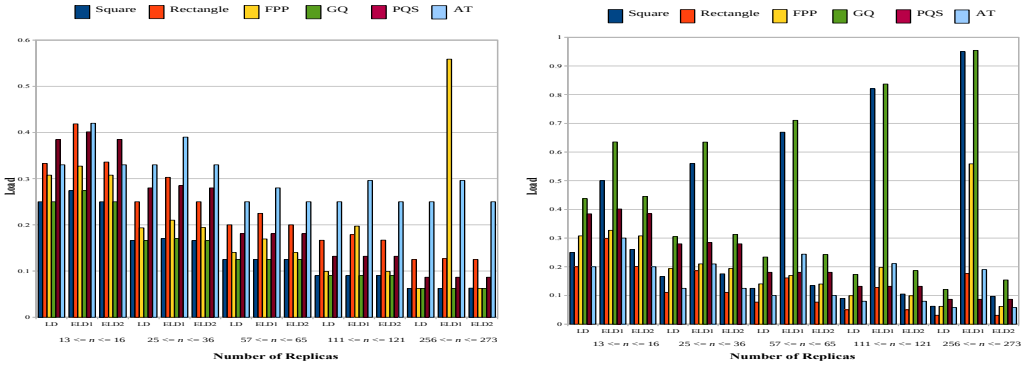
4.3 (Expected) system loads

The expected system load computations of read and write operations are carried out by using (3.5) and (3.6) respectively such that $p = 0.7$ for *ELD1* and $p = 0.9$ for *ELD2*. The "Square" and "GQ" have the least system load of $\frac{1}{\sqrt{n}}$ for read operations and such a load



(a) Read operation **(b)** Write operation

Figure 7 The availability of the operations for $p = 0.7, 0.8$ and 0.9.



(a) Read operation **(b)** Write operation

Figure 8 The system (LD) and expected system (ELD) loads for $p = 0.7$ and $p = 0.9$.

diminishes as the number of replicas n increases (see Figure 8(a)). Also, the expected system loads of both of these configurations are close to the fail-free system loads due to the high availability of their read operations. "AT" has the highest read system load of 0.25 when $n > 24$. Also, this configuration has the highest expected system loads for $24 < n < 200$. Note that when $n > 200$ and $p = 0.7$, the "FPP" configuration induces the highest expected system loads due to the fact that its read availability deteriorates gradually when $n > 100$.

Figure 8(b) illustrates that "Rectangle" of width w has the least system load of $\frac{w}{n}$ for write operations. Also, this configuration has the least expected system loads either when $p = 0.7$ or $p = 0.9$ except for certain cases. The "GQ" configuration has the highest write system load of $\frac{2\sqrt{n}-1}{n}$ which is quite similar to that of "PQS". Also, the former configuration has the highest expected system loads for any number of replicas n. It is worthwhile to note that the configuration "Square" has quite comparable expected system loads with respect to those of "GQ " when $p = 0.7$ due to the low availability of its write operations.

5 Conclusion

In this paper, we proposed a structured replica control protocol that, unlike the previous proposed ones, can be implemented using any logical 2D structure of height $h > 0$. We

presented two structures that provide near optimal performance for their read and write operations. The former has the best combined read and write costs of $2\sqrt{n}$ and induces the best combined read and write loads of $\frac{2}{\sqrt{n}}$, whereas the latter has a cost of w for its write operations which is lower than those of the existing structured RCPs and induces near best combined read and write loads of $\mathcal{O}(\frac{2}{\sqrt{n}})$; yet preserving comparable availability for its read and write operations. We also proposed a new protocol which provides better performance for its write operations than those of the well known *ROWA* [1] protocol while still maintaining comparable performance for its read operations.

6 Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme in the context of the FIT4Green project (grant agreement no. 249020) and the EuroNF Network of Excellence (grant agreement no. 216366).

References

1 Bernstein, P.A., Goodman, N.: An algorithm for concurrency control and recovery in replicated distributed databases. ACM Transactions on Database Systems, 9(4), 596-615 (1984)
2 Thomas, R.H.: A majority consensus approach to concurrency control for multiple copy databases. ACM Transactions on Database Systems, 4(2), 180-209 (1979)
3 Maekawa, M.: A $\sqrt{n}$ algorithm for mutual exclusion in decentralized systems. ACM Transactions on Computer Systems, 3(2), 145-159 (1985)
4 Cheung, S.Y., Ammar, M.H., Ahamad, A.: The grid protocol: a high performance scheme for maintaining replicated data. IEEE Transactions on Knowledge and Data Engineering, 4(6), 438-445 (1990)
5 Naor, M., Wool, A.: The load, capacity, and availability of quorum systems. SIAM Journal on Computing, 27, 214-225 (1998)
6 Agrawal, D., El Abbadi, A.: The tree quorum protocol: an efficient approach for managing replicated data. Proceedings of the sixteenth international conference on Very Large Databases, 243-254 (1990)
7 Agrawal, D., El Abbadi, A.: An efficient and fault-tolerant solution for distributed mutual exclusion. ACM Transactions on Computer Systems, 9(1), 1-20 (1991)
8 Choi, S.C., Youn, H.Y., Choi, J.S.: Symmetric tree replication protocol for efficient distributed storage system. Proceedings of the International Conference on Computational Science, 474-484 (2003)
9 Koch, H.: An efficient replication protocol exploiting logical tree structures. The 23rd annual International Symposium on Fault-Tolerant Computing, 382-391 (1993)
10 Kumar, A.: Hierarchical quorum consensus: a new algorithm for managing replicated data. IEEE Transactions on Computers, 40(9), 996-1004 (1991)
11 Bahsoun, J.P., Basmadjian, R., Guerraoui, R.: An arbitrary tree-structured replica control protocol. The 28th International Conference on Distributed Computing Systems, 502-511 (2008)
12 Peleg, D., Wool, A.: The availability of quorum systems. Inform. and Comput, 210-223 (1995)
13 Pease, M., Shostak, R., Lamport, L.: Reaching agreement in the presence of faults. Journal of ACM, 228-234 (1979)