Multi-Dimensional Quorum Sets for Read-Few Write-Many Replica Control Protocols

Bujor Silaghi, Pete Keleher and Bobby Bhattacharjee
Department of Computer Science, University of Maryland, College Park
{bujor, keleher, bobby}@cs.umd.edu

Abstract

We describe d-spaces, a replica control protocol defined in terms of quorum sets on multi-dimensional logical structures. Our work is motivated by asymmetrical access patterns, where the number of read accesses to data are dominant relative to update accesses, i.e. where the protocols should be read-few write-many. D-spaces are optimal with respect to quorum group sizes. The quality of the trade-off between read efficiency and update availability is not matched by existing quorum protocols. We also propose a novel scheme for implementing d-spaces that combines caching and local information to provide a best-effort form of global views. This allows quorum reconfiguration to be lightweight without impacting access latencies, even when the rate of membership changes is very high.

1 Introduction

This paper presents a new quorum protocol based on multi-dimensional logical structures called *d-spaces*. Our work is motivated by two dominant characteristics of presently deployed peer-to-peer (P2P) infrastructures and proposed overlay networks [10, 14, 11]. First, data reads account for the vast majority of accesses to items exported by P2P services. Presently deployed P2P file-sharing utilities fit this description, with the vast majority of data exported by participants unaltered after creation. Second, active participation in such networks is highly transient, with users/sites frequently joining and leaving the service [12].

Use of d-space quorums provides the following advantages:

- They provide a better combination of efficiency and availability than existing methods. In particular, a d-space can be configured to give optimal communication complexity (quorum set sizes) for any given read/write access ratio, without compromising availability.
- Their use of local information allows much more flexibility than approaches based on global views. One implication of this flexibility is that membership changes do not require global reconfiguration phases.

We support one-copy equivalence [4] and serializable executions. Together, these give us one-copy serializability, which is also the correctness criteria supported by the read-one write-all approach (ROWA) [2], the quorum consensus scheme [6], and the available copies method [7].

Supporting one-copy equivalence implies that all copies of an object should appear as a single item to clients. Efficiency requires that only a small fraction of the set of all copies be accessed during any read or write operation. This is especially important for read operations, which tend to represent the dominant write accesses for most application classes. Finally, we would like to achieve both strict consistency and operational efficiency without sacrificing the availability of data, as this is the main reason for data replication. All these requirements (read efficiency, update availability, and strict correctness) hold for applications that belong to the classical distributed systems domain even as they are being migrated or re-engineered to work in a P2P environment. For instance, distributed databases is one such application domain, and the PIER project [8] a notable effort in this direction.

There is a clear trade-off between the efficiency of a quorum protocol and its operational availability. The break-even point depends primarily on the logical structure (or the lack thereof) that arranges participating sites. We argue that protocols based on d-space quorums are superior to existing protocols both in terms of efficiency and in availability.

The remainder of this paper is structured as follows. In Section 2 we briefly describe existing quorum protocols with emphasis on structured quorums. Section 3 defines d-spaces, a quorum consensus replica control protocol on multi-dimensional spaces. In Section 4 we show that the protocol is optimal with respect to communication complexity (efficiency). In Section 5 we analyze the d-space operational availability. Section 6 contrasts the protocol's performance with two well known replica control protocols. A lightweight reconfiguration mechanism that combines local information and global views is discussed in Section 7. We summarize our findings and conclude the paper in Section 8.

2 Related work

Synchronization in quorum-based replica control protocols takes place by defining groups of sites that need to agree before launching an activity, and requiring the intersection of groups defined for conflicting activities. A read operation on a copy conflicts with all write operations on any copy of the object. A write operation on a copy conflicts with all read and write operations. We will refer to the group of sites needed to perform a read (write) operation as the quorum group for that operation. The collection of read (write) quorum groups is called a read (write) quorum set. Thus, any element of a read quorum set must intersect all elements of a write quorum set, which in turn must intersect among themselves in a pairwise fashion.

2.1 Voting

Gifford defines quorum sets in terms of weighted voting [6] for synchronizing concurrent accesses to shared files. If the total number of votes is v , v_r votes are needed to read a file, and v_w votes are needed to write a file, such that (i) $v_r + v_w > v$ and (ii) $2v_w > v$.

Thomas defines majority quorums as quorum sets for which each quorum group contains a majority of copies [15]. Again, this is a special case of weighted voting for which $v_r = v_w = \lfloor v/2 \rfloor + 1$. This assignment provides the best symmetric availability for read and write operations.

2.2 Structured quorums

Quorum sets defined on logical structures use structural information to define intersecting quorum groups. We briefly present the Grid protocol, the tree protocol, and the hierarchical quorum consensus protocol.

Cheung el al. defined a replica control protocol using quorums on a 2-dimensional grid [5]. Quorum groups for read operations consist of one line, and quorum groups for write operations consist of one line and one column. The authors observe that instead of a line, for both read and write quorums, a more relaxed configuration that requires one node in each column can be employed. The Grid protocol has low communication costs ($O(\sqrt{N})$), and is best suited for scenarios where the frequency of read and write operations are on the same order. Many variants and improvements of the Grid protocol have since been proposed by the research community.

The tree protocol proposed by Agrawal and El Abbadi organizes the set of copies in a binary tree with $\log N$ levels [1]. A quorum group is formed by including all the sites in some arbitrary path from the root to a leaf. The tree protocol has the lowest message complexity ($O(\log N)$) among all structured quorum schemes, assuming no site failures. In the presence of failures, the algorithm degrades gracefully as progressively more sites are involved in a quorum group, for a maximum of $N/2$. The approach is less appealing when considering the distribution of accesses over the set of copies. The root site is part of all quorum groups (assuming no failures), while a leaf site is part of $N/2$ times fewer quorum groups. The tree protocol is not truly distributed and employs a weak form of decentralization to ensure exclusion of accesses.

Kumar extends weighted voting to voting on multiple levels of a hierarchy comprising the set of all replicas [9]. In contrast to the tree protocol, physical copies of objects are stored only at the leaves of the tree, while all other levels serve a logical grouping purpose. In effect, the protocol performs a hierarchical partitioning of the replica set. Given a perfectly balanced tree with $m + 1$ levels (with the root on level 0 and replicas on level m) such that a node on level i has l_{i+1} children, the overall number of replicas is $\prod_{i=1}^{m} l_i$. A node assembled on level i must in turn recursively assemble r_{i+1} (w_{i+1}) of its l_{i+1} children nodes on level $i+1$ for a read (write) quorum group. The root node is part of all read (write) quorum groups. The quorum intersection condition is satisfied if (i) $r_i + w_i > l_i$, and (ii) $2w_i > l_i$ for all levels $i = 1...m$.

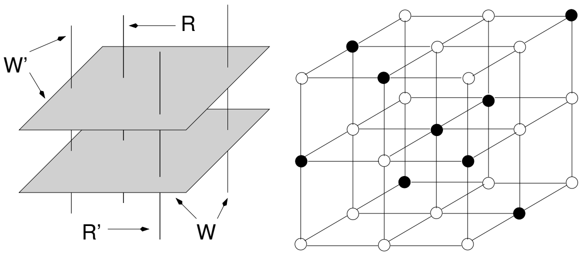
A read quorum group defined by the hierarchical quorum consensus scheme consists of $\prod_{i=1}^{m} r_i$ copies, and a write quorum groups of $\prod_{i=1}^{m} w_i$ copies. Optimal quorum group sizes are obtained for the hierarchical consensus method when each group contains three subgroups, i.e. $l_i = 3$. In this case symmetrical quorum groups consist of $N^{0.63}$ sites. The method allows for imbalanced quorum groups for read and write operations to be specified. For these reasons we have chosen it to contrast the performance and availability of the d-space approach.

3 D-space quorums

We define quorum groups on multi-dimensional spaces and show how read-few write-many replica control protocols can be implemented using our method.

3.1 Definition

Assume we have N replicas of a data item arranged in a d-dimensional discrete space D, and that each dimension i in D is indexed from 1 to n_i, i.e. $N = \prod_{i=1}^{d} n_i$. D is an abstract space, and the N replicas do not necessarily correspond to physical copies. For this reason we will refer to replicas as *nodes* and not *sites*. In Section 7.2 we discuss



a) b)

Figure 1. Example of 3-space read and write quorum groups. a) A read quorum (R or R') intersects all write quorums. A write quorum (W or W') intersects all read quorums and all other write quorums. b) A cover of a plane (filled points) can be used instead of the plane.

how nodes are mapped to sites. Until then we assume that nodes (replicas) are sites (servers).

We choose k of the d dimensions in D, and let $u_{i_1}, u_{i_2}, \ldots, u_{i_k}$ be k arbitrary coordinates on selected dimensions $i_1, i_2, \ldots i_k$. Similarly, let $v_{j_1}, v_{j_2}, \ldots, v_{j_{d-k}}$ be arbitrary coordinates on the rest of the $d-k$ dimensions $(j_1, j_2, \ldots, j_{d-k})$. We define subspaces U and V to be:

$$U = \{(x_1, \ldots, x_d) \in D \wedge (x_{i_t} = u_{i_t})_{t=1 \ldots k}\} \quad (1)$$

$$V = \{(x_1, \ldots, x_d) \in D \wedge (x_{j_t} = v_{j_t})_{t=1 \ldots d-k}\} \quad (2)$$

Subspaces U and V overlap and their intersection is minimal. There exists a single point (node) common to U and V. The intersection point is given by coordinates $u_{i_1}, \ldots, u_{i_k}$, $v_{j_1}, \ldots, v_{j_{d-k}}$ written in canonical order. In a 2-dimensional space, U and V represent intersecting lines parallel to the two axes. Note that U and V factorize space D in the sense that $|D|$=$|U||V|$. Thus each of the two subspaces contains considerably fewer nodes that the original space D. In particular, the sum $|U| + |V|$ is minimized when $|U| = |V|$.

Let a read quorum group be V, and a write quorum group be $V \cup U$. Any two write quorum groups intersect, and any read quorum group intersects any other write quorum group. The quorum intersection property is satisfied and reads and writes are serializable. In particular, one-copy serializability [3] is guaranteed. On the left side of Figure 1 we illustrate read and write quorum groups for a 3-space quorum set. A read quorum requires 3 nodes, and a write quorum requires $9 + 2 = 11$ nodes. For a 3-space there exists another space factorization (0 versus 3 dimensions) which is in effect ROWA.

For clarity of exposition we will assume hereafter that the extension of the replica space is the same along all dimensions, i.e. $n_i = N^{\frac{1}{d}}$ for all $i = 1 \ldots d$. All the results presented can easily be extended to account for the general case formally defined above. Given the new constraints, the replica space becomes a regular d-dimensional space, with space V containing $N^{\frac{k}{d}}$ points and space U containing $N^{\frac{d-k}{d}}$ points. A read quorum group will thus consist of $N^{\frac{k}{d}}$ nodes while a write quorum group will consist of $N^{\frac{k}{d}} + N^{\frac{d-k}{d}} - 1$ nodes.

3.2 The Read-Few Write-Many approach

More interesting for distributed systems are quorums sets that are highly asymmetrical, i.e. for which d is relatively high compared to k. We call this instantiation of d-space quorum sets the *read-few write-many* approach (RFWM). One such instance is given by read quorum groups consisting each of $N^{\frac{1}{d}}$ nodes (a line), and write quorum groups consisting each of $N^{\frac{1}{d}} + N^{\frac{d-1}{d}} - 1$ nodes (a line and a hyper-plane). For ease of presentation we will also refer to $N^{\frac{k}{d}}$ as a line and to $N^{\frac{d-k}{d}}$ as a hyper-plane.

Read-few write-many replica control protocols are amenable to scenarios where the frequency of read operations is orders of magnitude higher than the frequency of write operations. We call the ratio of frequencies for read and write operations the *read/write access ratio*. The communication complexity of a replica control protocol is the expected number of replicas to be contacted per operation. Our goal is to minimize the protocol's communication complexity, irrespective of operation type (i.e. read or write). Therefore, the ratio of quorum group sizes for read and write operations should be inverse to the read/write access ratio. By ρ_{wr} we denote the ratio of the write quorum size to the read quorum size. When ρ_{wr} equals the read/write access ratio, the communication complexity of a read-few write many replica control protocol is minimized. Any proportion of read and write operations can be modeled by choosing appropriate values for k and d (or more generally, for d and dimension extensions n_i).

Although we have defined read and write quorum groups in terms of projective subspaces, the set of $N^{\frac{d-k}{d}}$ nodes in a write quorum group need not strictly conform to the definition. We allow any cover of a hyper-plane to act as the second component of a write quorum group. A *cover* of a hyper-plane is any set of points such that their projection covers the whole hyper-plane. For our discrete space we are interested in the minimal cover of a hyper-plane, i.e. a cover having exactly $N^{\frac{d-k}{d}}$ points. Relaxing a write quorum group to a line combined with the cover of a hyper-plane greatly improves the availability of write operations. On the right side of Figure 1 we show a (minimal) cover for a plane that can be part of a 3-space write quorum group. Note that any of the three planes parallel to the base of the cube is covered by the cover shown.

As an example of a RFWM replica control protocol using quorum consensus on d-spaces, we note the classical approach of using version numbers to identify the latest update, and locking to enforce mutual exclusion.

4 Optimality of communication complexity

As noted above we expect read and write quorum group sizes to be inversely proportional to the access frequency for the corresponding operations. We argue that replica control protocols using d-spaces provide optimal communication complexity, i.e. read and write quorum sizes are minimal for a given access ratio. We require the following constraints on any quorum set defined on d-spaces:

QS1 Each read (write) quorum group in the set has r (w) nodes. The condition ensures that the message complexity of an operation is independent of the quorum group chosen.

QS2 Each node appears in at least one quorum group. The condition ensures that all copies are used effectively.

QS3 Each node is contained in the same number of read (write) quorum groups. The condition ensures uniform load sharing over the set of all copies (assuming quorum groups are selected uniformly at random when performing operations).

Given conditions QS1-3, Theorem 1 states the optimality of the approach. The following lemma will help us prove the theorem.

Lemma 1. *A set $\mathcal{W}$ that intersects all elements of a read quorum set satisfying conditions QS1-3 contains at least $w = N/r$ elements.*

Proof. Assume each node is contained in g distinct read quorum groups (by QS3). We also have that $g > 0$ (by QS2). The total number of nodes, considering all duplicates as distinct elements, is gN. Since there are r nodes in each read quorum group (by QS1), there are gN/r read groups in the read quorum set.

We construct $\mathcal{W}$ starting with the empty set, such that $\mathcal{W}$ intersects all gN/r read quorum groups. Every node added to $\mathcal{W}$ is contained in exactly g of the read quorum groups, and ensures the intersection of $\mathcal{W}$ with the corresponding groups. Thus, with the addition of one node we can cover the intersection of $\mathcal{W}$ with at most g groups in the quorum set. Since there are gN/r quorum groups, at least N/r nodes need be added to $\mathcal{W}$ to cover its intersection with all groups in the read quorum set. □

Theorem 1. *Read quorum sets defined using d-spaces are optimal with respect to quorum group size for any read/write access ratio ρ_{wr}. Write quorum sets are optimal within a factor of 2.*

Proof. Let k and d be such that such that $N^{\frac{d-k}{d}} \approx N^{\frac{k}{d}}\rho_{wr}$. We define d-space read quorum groups of size $N^{\frac{k}{d}}$ and write quorum groups of size $N^{\frac{k}{d}} + N^{\frac{d-k}{d}} - 1$ in the usual manner. We have that $(N^{\frac{k}{d}} + N^{\frac{d-k}{d}} - 1)/N^{\frac{k}{d}} = O(w/r)$, and the quorums satisfy the read/write access ratio.

A read quorum group contains $N^{\frac{k}{d}}$ nodes. Every node appears in exactly one of the read quorum groups ($g = 1$). In fact, the read quorum set defines a partition on the set of all copies, where each member of the partition has the same number of nodes. Conditions QS1-3 are thus satisfied and by Lemma 1 we have that write quorum groups must contain at least $N^{\frac{d-k}{d}}$ elements. Since $N^{\frac{k}{d}} + N^{\frac{d-k}{d}} - 1 \leq 2N^{\frac{d-k}{d}}$ (assuming $k \leq d - k$), we have that write quorum groups are within a factor of 2 from the optimal size.

Read quorum groups cannot contain less than $N^{\frac{k}{d}}$ nodes since that would proportionately increase the size of write quorums (as given by Lemma 1). This would break the read/write access ratio. Thus, read quorum groups are optimal with respect to size. □

We describe how k and d can be chosen such that the access ratio condition is satisfied. Given N nodes and access ratio ρ_{wr}, we are looking for quorum sizes for write and read operations, w and r, such that (i) $w = N/r$, and (ii) $\rho_{wr} = w/r$. Thus we have that $w = \sqrt{N\rho_{wr}}$ and $r = \sqrt{N/\rho_{wr}}$. N can be factorized in the list of its prime factors. The list can then be partitioned in two sublists such that the multiplication of prime factors in one list approximates w, and in the second list approximates r.

Given w and r, values for k and d can easily be identified such that $N^{\frac{k}{d}}$ approximates r, and $N^{\frac{k}{d}} + N^{\frac{d-k}{d}} - 1$ approximates w. For the general case, where the extension of the replica space does not have to be the same along all dimensions, we have more flexibility in choosing the parameters. If N has few prime factors (e.g. N is a prime number itself), a neighboring number of N can be factorized instead of N. In this case, a few holes will be present in the structure, and quorum groups containing the holes will not be operational. In Section 7.2 we discuss how nodes can be mapped onto sites such that N is chosen at will, and any number of physical copies is naturally accommodated.

5 Availability analysis

Fault tolerance is reflected in the availability of the last update (data availability), and the availability of read and write operations. We establish these availabilities next. We assume that the network is reliable, node failures are both independent and fail-stop [13], and all nodes are identical. Let p be the probability of a node being operational, i.e. the node's availability. Given p, the probability of finding m operational nodes among the N nodes is given by the binomial distribution:

$$b(N, m, p) = \left(\begin{array}{c} N \\ m \end{array} \right) p^m (1-p)^{N-m} \qquad (3)$$

5.1 Read availability

The availability of read operations is the probability that the operation concludes successfully, assuming no state changes while it is in progress. Read operations are robust: any line of $N^{\frac{k}{d}}$ nodes needs to be operational for a read to succeed. There are $N^{\frac{d-k}{d}}$ candidate lines to choose from. A line is available with probability $p^{N^{\frac{k}{d}}}$, while m *selected* lines are available with probability:

$$\alpha_{line}(m) = p^{mN^{\frac{k}{d}}} \tag{4}$$

Knowing that there are $N^{\frac{d-k}{d}}$ potential lines to choose from, the availability of read operations is given by:

$$\begin{aligned} \alpha^{R}_{RFWM} &= 1 - (1 - \alpha_{line}(1))^{N^{\frac{d-k}{d}}} \\ &= 1 - (1 - p^{N^{\frac{k}{d}}})^{N^{\frac{d-k}{d}}} \end{aligned} \tag{5}$$

5.2 Write availability

ROWA is deemed unsatisfactory due to its stringent requirement that all copies be available whenever an update occurs. The read-few write-many approach approximates ROWA from the read efficiency standpoint, and dramatically improves over its write availability. The availability of write operations is the probability that the operation concludes successfully, assuming no state changes while it is in progress. A write is successful if one line of $N^{\frac{k}{d}}$ nodes together with a cover of a hyper-plane of $N^{\frac{d-k}{d}}$ nodes can be accessed. Note that since the line already covers one node in the hyper-plane, only $N^{\frac{d-k}{d}} - 1$ nodes need to be further covered. More generally, a write is successful if m lines and a cover of $N^{\frac{d-k}{d}} - m$ nodes are operational.

The cover of a hyper-plane of m nodes is available if there is at least one available node in each of the m corresponding lines. We further require that each such line not be fully available. At least one node, but not all of them, is available in a line with probability:

$$\begin{aligned} \alpha_{pointC} &= \sum_{m=1}^{N^{\frac{k}{d}}-1} b(N^{\frac{k}{d}}, m, p) \\ &= 1 - p^{N^{\frac{k}{d}}} - (1-p)^{N^{\frac{k}{d}}} \end{aligned} \tag{6}$$

The availability of a hyper-plane cover of m *selected* nodes is:

$$\begin{aligned} \alpha_{planeC}(m) &= (\alpha_{pointC})^{m} \\ &= (1 - p^{N^{\frac{k}{d}}} - (1-p)^{N^{\frac{k}{d}}})^{m} \end{aligned} \tag{7}$$

To compute the availability of write operations we add the probabilities for all combinations of m available lines (4) and $N^{\frac{d-1}{d}} - m$ additional nodes available in a cover (7):

$$\begin{aligned} \alpha^{W}_{RFWM} &= \sum_{m=1}^{N^{\frac{d-k}{d}}} \binom{N^{\frac{d-k}{d}}}{m} \alpha_{line}(m) \cdot \alpha_{planeC}(N^{\frac{d-k}{d}} - m) \\ &= (\alpha_{line}(1) + \alpha_{planeC}(1))^{N^{\frac{d-k}{d}}} - \alpha_{planeC}(N^{\frac{d-k}{d}}) \\ &= (1 - (1-p)^{N^{\frac{k}{d}}})^{N^{\frac{d-k}{d}}} \\ &\quad - (1 - p^{N^{\frac{k}{d}}} - (1-p)^{N^{\frac{k}{d}}})^{N^{\frac{d-k}{d}}} \end{aligned} \tag{8}$$

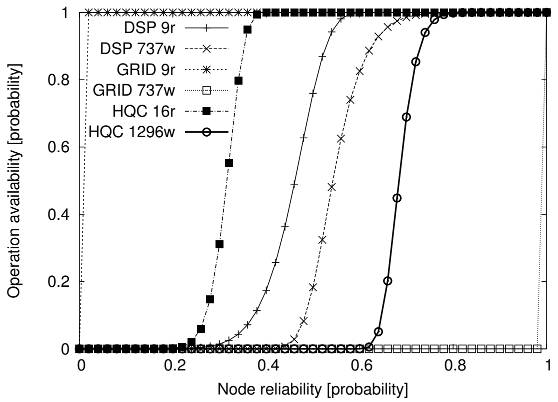


Figure 2. Operation availability for DSP, HQC and GRID with $6,561$ nodes and $\rho_{wr} = 81$.

6 D-spaces, Grid, and HQC

We compare the communication complexity and operation availability of the d-space (DSP), Grid (GRID), and hierarchical quorum consensus (HQC) methods. We only examine skewed scenarios, where read operations dominate accesses to data.

HQC is optimal when each logical group is decomposed in three subgroups, and the resulting hierarchy is perfectly balanced. We choose the number of nodes with respect to such criteria, i.e. $N = 3^m$ ($l_i = 3$). Further, the quorums in HQC were distributed at each level such that write availability is optimized since updates are the critical component with respect to failures (i.e. consistently having lower availability than read operations).

6.1 Fault tolerance: operational availability

Figure 2 shows the read and write availability as a function of node reliability for d-spaces, grids, and hierarchical quorum consensus. $6,561$ nodes are considered and the access ratio modeled is $\rho_{wr} = 81$. The overall availability of a protocol is established by the curve that has poorer availability among the read and write curves. For all three protocols considered reads always have better availability than updates. Thus, DSP can tolerate higher degree of failures than both GRID and HQC.

GRID has the poorest availability for the configuration shown in Figure 2. The Grid protocol targets symmetrical scenarios for which the expected frequency of read and write operations is approximately the same ($\rho_{wr} \approx 1$). The d-space protocol is a generalization of Grid to multiple dimensions. It is also reciprocal to Grid in the sense that it defines inverted quorum groups with respect to subspace covers. These two properties enable it to deliver good update availability even when reads occur orders of magnitude more often than updates. If the frequency of executing read and write operations is substantially different, quorums should be defined using the d-space approach, otherwise they should be defined using the Grid approach.

6.2 Efficiency: communication complexity

We argued that d-space quorum sets have optimal message complexity for any access ratio. We now show how communication costs for the hierarchical quorum consensus relate to d-spaces. Grid has the same communication complexity as d-spaces.

In Figure 2 we labeled the availability curves using the number of nodes in the read and write quorum groups. For the configuration shown, HQC has read and write quorums that are almost double in size as compared to d-spaces (or Grid). More generally, it can be shown that the cost ratio for read operations in HQC versus DSP is given by:

$$\gamma^R = \left(\frac{N}{\rho_{wr}}\right)^{\log_9\left(\frac{4}{3}\right)} \approx \left(\frac{N}{\rho_{wr}}\right)^{0.13} \qquad (9)$$

while for write operations is given by:

$$\gamma^W = \frac{\rho_{wr}}{\rho_{wr}+1}\gamma^R \qquad (10)$$

We show in Figure 3 how the efficiency of HQC and DSP relate to each other as a function of system size, for read and write operations and access ratios 9, 81, and 729. For a given ρ_{wr}, the ratio of communication costs for both read and write operation grows with N. Thus, the d-space approach scales better than HQC. Implementing RFWM using d-spaces results in a message complexity 2–3 times lower than implementing it using hierarchical quorum consensus. Beside saving network resources this also materializes in better load sharing at sites holding copies, and increased system throughput. The expected increase in system throughput is proportional to γ^R for read operations, and γ^W for write operations.

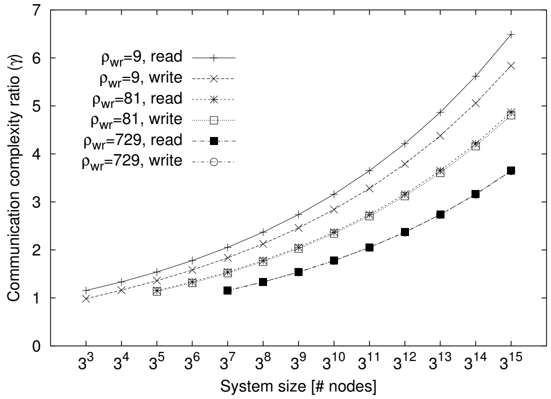


Figure 3. Communication complexity ratio (HQC/DSP) for read and write operations.

7 Quorum reconfiguration

P2P systems are highly volatile networks in the sense that the rate of membership changes (users/sites joining and leaving the system) is very high. A site leaving the network can make at least one quorum unavailable. Mechanisms are needed to reconfigure quorum groups on the d-space structure (or the d-space structure itself) such that deadlock scenarios when quorum groups can no longer be assembled are avoided. Existing approaches use global knowledge to reconfigure quorum groups. In order to consistently establish a new global view, global agreement is required usually in the form of having to gather a write quorum. Global knowledge, even in limited forms is not a feasible option for P2P networks that experience frequent membership changes.

7.1 Global view vs. local information

The fact that logically-structured quorum approaches are not accommodating to mutations (reconfiguring the space, adding or removing nodes individually or in group) is inherent to its global view of a logical structure. Flexibility is sacrificed for the sake of efficiency. At the other extreme, similar structures are maintained by the CAN overlay network [10] to route among participants in peer-to-peer networks. CAN lacks a global view and uses local information (each sites knows its $2d$ immediate neighbors) to arrange the sites in a d-dimensional torus.

Reconfiguration can be localized if only local information is used to route packets. However latency is hurt as an isolated remote contact for a node in CAN networks requires $O(dN^{\frac{1}{d}})$ incremental hops toward the destination. The latency to access a quorum group in d-spaces with local information is given by the size of the corresponding group, i.e. $N^{\frac{k}{d}}$ for reads and $N^{\frac{d-k}{d}} + N^{\frac{k}{d}} - 1$ for updates. Even though quorum sizes are the same irrespective of implementation choice (global view or local information), the high latency of the latter makes it prohibitive as a support mechanism for implementing structured quorums. Further, the message complexity of the local information approach also increases as incremental forwarding is required to reach arbitrary nodes in the overlay network.

Node space virtualization provides a local alternative to global reconfiguration protocols. The method defines the node space independent of the set of sites holding the physical copies. Local information is combined with a best-effort form of global views to achieve seemingly contradicting goals: lightweight local reconfigurations and good global latencies.

7.2 Virtualizing the d-space

Though we have heretofore assumed that they were the same, there are advantages to making a distinction between the replica (node) space, and the set of sites (servers) hosting the physical copies. Which replicas are hosted by servers is given by the mapping procedure.

A server can host more than one replica; a replica is hosted by exactly one server. Replicas form a pre-defined and static d-space. The replica space is virtual in the sense that there is no one-to-one correspondence between nodes and physical copies of a data item. Instead, all nodes mapped to a site correspond to one physical copy. Servers form a dynamic (with respect to membership changes) and unstructured space. Thus, the server space is the same as the space of physical copies, and there is a many-to-one hosting relationship among nodes and servers. Data values and version numbers of replicas mapped to a server are automatically kept consistent at all times.

The replica space is very large, such that the number of sites will never match the number of nodes, and will feature high dimensionality. For instance, $N = 2^d$ with d fixed (32 for instance) and the d-space becomes a hypercube. Each node has exactly d neighbors in the replica space, and the shortest path between any two nodes is bounded by d.

7.3 Joining and leaving the structure

A site has as many neighboring sites as defined by the mapping of nodes to sites. Two sites are neighboring each other in the server space if and only if they host neighboring nodes in the hypercube replica space. When sites join or leave the structure, we use hypercube routing to forward requests in the network, and a protocol similar to CAN's for splitting and merging the zones (subspaces) assigned to affected sites. Given the number of sites in the system S, the assignment protocol can perform an almost perfectly balanced mapping of nodes to sites: 90% of sites will hold N/S nodes, while the rest will hold half or twice of that [10].

A virtual node space gives us the flexibility to make localized adjustments upon joining and leaving. Upon joining, the newly joined site will initialize the version numbers and the data items to those of the site splitting its volume. Upon leaving, data transfer is necessary only if the recipient's versions are smaller than the leaving site's versions. When a server joins or leaves the network only its neighbors need to be informed. Thus, even though the replica space is static, the mapping of nodes to sites allows arbitrary mutations of the network through purely local reconfigurations.

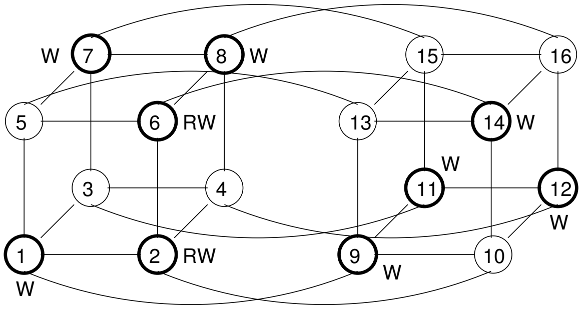


Figure 4. Example of cached read and write quorum groups for a 4-dim hypercube.

7.4 Caching quorum groups

The resulting system uses caching to match the low latencies achieved with global views. Every site caches a write quorum. Since a write quorum includes a read component, sites implicitly cache a read quorum as well. Caching a quorum translates to caching a list of mappings from subspace ranges to site identities (a site's identity is the tuple of its network address and port number). Note that the amount of state cached for a write quorum is smaller than with global views, where each server knows the full mapping of replicas in the network.

When a read or write request arrives at a site, the corresponding quorum cached at the site is used. Some of the mappings cached for a quorum group may be stale due to zone reassignments trigered by membership changes. Stale mappings are detected by timing out, or by confronting the cached mapping with the actual mapping. Mappings are refreshed by identifying the correct mapping using hypercube routing in the overlay network. This is reflected in latency overhead and increased number of messages. However, refreshing a stale cached mapping does not necessarily incur full hypercube routing. For all practical purposes, refreshing an entry incurs one or at most two overlay hops.

We illustrate this in Figure 4 with a 2^4 d-space, and examples of cached quorums at one of the servers (assume that there also 16 servers in the network, and that each node is mapped to a different server). For every node in a cached quorum there are other cached nodes in its proximity. For instance, if the mapping of node 14 needs to be refreshed, we can optimize the routing by starting at node 6 which is only one hop away and has just been validated. Similarly, nodes 2, 8, 9 and 12 are within two hops from 14. The overall cost of keeping cached quorums up-to-date in light of membership changes is low, and the approach approximates the performance of global views even for high rates of changes. Note that if there are no membership changes in the network, cached quorums are constantly up-to-date and the performance penalty is null.

7.5 Fault tolerance

Virtualizing the node space enables sites to join and leave the structure without invalidating existing quorums (by creating holes in the structure), or requiring a global reconfiguration mechanism. Faulty sites that do not recover (or recover too late) can also be eliminated through local reconfiguration.

With quorum consensus schemes, transient failures require no special treatment. If a site fails and recovers too late, or does not recover at all, the failure is considered permanent. Sites that fail permanently and recover subsequently must join the structure anew before further processing requests. The data and associated versions hosted by the permanently failing site are lost and cannot be recovered. To ensure consistency, the neighbor that takes over the subspace and the lock will need to perform a read operation for the reassigned data items.

8 Concluding remarks

We have described d-space, a replica control protocol using quorum consensus on replicas logically arranged in multi-dimensional spaces. D-space is a generalization of the Grid protocol to multiple dimensions. It is also reciprocal to Grid in the sense that it defines inverted read and write quorum groups with respect to subspace covers.

The central argument of our study is that d-space quorums offer a flexible way to build protocols with ideal balances of low communication complexity and high availability. First, d-spaces allow the more frequent read operations to execute efficiently, at a limited and controllable expense of more rarely executed write operations. This leads to our first result: for any given read/write access ratio, a d-space can be configured to give optimal communication complexity. Second, read operations can be performed efficiently without adversely affecting the availability of updates. To our knowledge, the quality of trade-off between read efficiency and update availability of our approach is not matched by existing quorum protocols. Surprisingly, for high access ratios the availability of updates can approximate or even match the availability of read operations.

Existing structured quorum schemes are based on global views. This allows good access latencies but hurts a protocol's adaptivity, as it must rely on heavyweight global reconfiguration mechanisms. We show how to implement lightweight d-space reconfiguration through a combination of local information and caching of global views.

References

[1] D. Agrawal and A. E. Abbadi. An efficient solution to the distributed mutual exclusion problem. In *Proc. of the* 8^{th} *ACM Symposium on Principles of Distributed Computing*, pages 193–200, Edmonton, Alberta, Canada, June 1989.

[2] M. Ahamad, M. Ammar, and S. Cheung. Replicated data management in distributed systems. In T. L. Casavant and M. Singhal, editors, *Readings in Distributed Computing Systems*, pages 572–591. IEEE Computer Society Press, Los Alamitos, CA, January 1994.

[3] P. A. Bernstein and N. Goodman. An algorithm for concurrency control and recovery in replicated distributed databases. *ACM Transactions on Database Systems*, 9(4):596–615, December 1984.

[4] P. A. Bernstein and N. Goodman. A proof technique for concurrency control and recovery algorithms for replicated databases. *Distributed Computing*, 2(1):32–44, Jan 1987.

[5] S. Y. Cheung, M. H. Ammar, and M. Ahamad. The Grid protocol: A high performance scheme for maintaining replicated data. *IEEE Transactions on Knowledge and Data Engineering*, 4(6):582–592, December 1992.

[6] D. K. Gifford. Weighted voting for replicated data. In *Proc. of the* 7^{th} *SOSP*, pages 150–162, Pacific Grove, CA, Dec 1979.

[7] N. Goodman, D. Skeen, A. Chan, U. Dayal, S. Fox, and D. Ries. A recovery algorithm for a distributed database system. In *Proc. of the* 2^{nd} *ACM Symposium on Principles of Database Systems*, pages 8–15, Atlanta, GA, March 1983.

[8] R. Huebsch, J. M. Hellerstein, N. Lanham, B. T. Loo, S. Shenker, and I. Stoica. Querying the internet with PIER. In *Proc. of the* 29^{th} *International Conference on Very Large Data Bases*, pages 321–332, Berlin, Germany, Sep 2003.

[9] A. Kumar. Hierarchical quorum consensus: A new algorithm for managing replicated data. *IEEE Transactions on Computers*, 40(9):996–1004, September 1991.

[10] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *Proc. of the ACM SIGCOMM*, San Diego, CA, August 2001.

[11] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. of the* 18^{th} *Intl. Conf. on Distributed Systems Platforms*, Heidelberg, Germany, Nov 2001.

[12] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proc. of MMCN*, San Jose, CA, Jan 2002.

[13] R. D. Schlichting and F. B. Schneider. Fail-stop processors: an approach to designing fault-tolerant computing systems. *ACM Transactions on Computer Systems*, 1(3):222–238, August 1983.

[14] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable P2P lookup service for Internet applications. In *Proc. of SIGCOMM*, San Diego, CA, Aug 2001.

[15] R. H. Thomas. A majority consensus approach to concurrency control for multiple copy databases. *ACM Transactions on Database Systems*, 4(2):180–209, June 1979.