Hypercube Quorum Consensus for Mutual Exclusion and Replicated Data Management

Ada Waichee Fu Wai Kwong Lau Fuk Keung Ng Man Hon Wong
Department of Computer Science and Engineering
The Chinese University of Hong Kong
email: adafu@cse.cuhk.edu.hk,mhwong@cse.cuhk.edu.hk

Index Terms: Distributed system, coterie, quorum consensus, replicated databases.

1 Introduction

Mutual exclusion in distributed systems has been studied for a long time. Many synchronization problems which demand co-ordination from multiple sites require that only one site among a number of sites is allowed to use some resources at a time. Solutions to these problems are expected to tolerate site and communication failures which may lead to network partitioning. Quorum consensus [3] is a well-studied method for such problems.

One typical case of mutual exclusion occurs in the transaction management problem of a replicated database system. A replicated database is a distributed database in which multiple copies of some data items are stored at multiple sites. One of the advantages of data replication is to increase data availability so that the system can remain operational even though some sites have failed. Another advantage of data replication is to improve performance. With many copies of each data item being available, a user transaction is more likely to find the data it needs nearby. However, these benefits are offset by the cost of maintaining data consistency. Synchronization protocols, such as those mentioned in [4, 5, 10, 13, 2] are needed to coordinate the operations on the replicas. This increases the cost of executing operations in replicated databases.

One of the proposed protocols for the above problem is the *hierarchical quorum consensus protocol* [12]. In this paper, we generalize the hierarchical quorum consensus protocol and show that the generalized protocol can exhibit hybridized behaviour of different kinds of protocols. We present a special family of such protocols, which we call the **hypercube quorum consensus**. We show that the proposed protocol has a number of desirable properties, including small quorum size, small communication delay, high availability, and being non-dominated. This protocol imposes a hypercube structure on the set of copies of the data items. We call this the *hypercube protocol.* Other quorum consensus algorithms include the tree quorum protocol [2], the grid structure protocol [13], etc.

We analyze one special instance of the hypercube protocol, known as the *majority hypercube protocol.* This protocol makes use of the majority quorum. We also analyse the use of other protocols inside the hypercube protocol.

This paper is organized as follows. In Section 2, we describe the model of a replicated database. In Section 3, we present the notion of coterie, bicoterie and wr-coterie which describe formally the requirements of the quorum consensus protocols. Section 4 introduces the generalized hierarchical quorum consensus protocol. In Section 5, we present the hypercube coterie protocol, its definition and correctness. In Section 6, we describe a method for mapping hypercubes to a 2-dimensional grid network which can achieve minimal communication delay. In Section 7, we present the quorum size and the availability analysis of the hypercube quorum protocol. Section 8 describes an experiment on the hybridization of the protocol. The final section is a conclusion.

2 Model

In our model, a distributed database system consists of a number of sites. Each pair of these sites are connected by bidirectional links. They communicate with each other by sending messages over a communication network. We assume that each site has its own memory and that each site can suffer from fail-stop failure. We also assume that communication links may fail to deliver messages, so that network partitioning can occur, and divide the system into two or more partitions. Sites within a partition can still communicate with each other; but they cannot deliver messages to other sites in a different partition.

In our model, a database consists of a number of data items being stored at several sites. *Transactions* are used to manipulate these data items. A transaction is a sequence of operations with a number of read and write operations that manipulate the data items. The execution of a transaction should be atomic.

In a replicated database system, data items are replicated at several sites. The data item is called the logical data item, whereas the copy of a data item stored at a particular site is called a physical data item. We adopt *one-copy serializability* [6] as the correctness criterion for our system. In order to ensure one-copy serializability, we may adopt some locking mechanisms, and a read/write operation on a data item X will access a read/write quorum which is a subset of the copies of X. We shall define a *coterie* for a given set of data copies (or replication sites) which defines what quorums are allowed for the operations.

3 Preliminaries

In this section, we formalize the requirements and some desirable properties of a quorum consensus protocol. Some definitions from [15] are adopted in this paper. Let $G = (V, E)$ be a network, where V is the set of sites and E is the set of links which connect the sites. A set S

of subsets of V is a **coterie** on G if the following conditions hold:

1. *Intersection:* $\forall M_1, M_2 \in S, M_1 \cap M_2 \neq \phi$. That is, any two sets in S must have at least one common node.
2. *Nonredundancy:* There are no $M_1, M_2 \in S$ such that $M_1 \subseteq M_2$.

If we think in terms of system operations, each operation will access one quorum in the coterie. Two of the desirable properties of a quorum consensus protocol are small quorum size and high availability in failure conditions. Some coteries are then definitely better than some other coteries. Suppose the network contains sites a, b and c only. Let A be coterie $\{\{a, b\}\}$ and B be coterie $\{\{a, b\}, \{a, c\}\}$. If b fails, then no operation is allowed with A, but operations are allowed with B, by the quorum $\{a, c\}$. Hence B is better than A. For another example, if A is a coterie $\{\{a, b, c\}\}$ and B is a coterie $\{\{a, b\}\}$, then B is better than A, since an operation in B needs only to access two sites a, b, but an operation in A needs to access three sites a, b, c. In [9], A is said to be dominated by B in both of the above cases. More formally, we have the following definition.

- Let R, S be coteries on G, R **dominates** S iff $R \neq S$ and, for each $H \in S$, there is an $H' \in R$ such that $H' \subseteq H$.

A coterie S on G is **dominated** iff there is another coterie on G which dominates S. If there is no such coterie, then S is **nondominated (ND)**.

Finally, we define terms for the property for communication cost required in quorum consensus as in [8]. In the following definition, the term "distance" may represent the communication overhead required between two locations in the given network instead of actually a physical distance.

Definition 1: Given a coterie C for a network, where the length of each edge is given. For a site s in the network, we can find a quorum Q in C such that the maximum distance from s to any site in Q is minimum. In other words, among all quorums, Q is the quorum that contains sites least distant from s. We call Q an **optimal quorum for** s. We call this maximum distance the **delay** of s for C. Consider the maximum of delays among all sites. We call this maximum of delays the **max-delay** of the coterie. We refer to a coterie with a minimal value of max-delay (among all coteries) a **max-delay optimal coterie**. $\Box$

3.1 Bicoteries and Wr-coteries

The previous definitions consider only the problem of mutual exclusion without considering database transactions. Here we consider an application of the ideas of coteries on the transaction management problem in distributed replicated database systems. In such an application, we distinguish read operations from write operations.

The following definitions are from [11], in which U is the given set of sites, and a group refers to a set of sites:

Bicoterie: An ordered pair $B = (P, Q)$, where P and Q are sets of subsets of U, is a bicoterie under U if

1. for each group G in P or Q, $G \neq \phi$.
2. (*Intersection Property*) for each group G in P and each group H in $Q, G \cap \mathrm{H} \neq \phi$.
3. (*Nonredundancy*) for any two groups G and $H \in$ P, G is not a proper subset of H, and for any two groups G and $H \in Q$, G is not a proper subset of H. □

wr-coterie: A bicoterie B = (P, Q) is called a write-read coterie (wr-coterie, for short), if it satisfies

- For all G, H ∈ P : $G \cap H \neq \phi$ (i.e. P is a coterie). □

A wr-coterie $B = (P, Q)$ can be used to form read and write quorums by selecting read quorums from Q and write quorums from P.

The definition of non-domination for coteries can be extended to the case of bitcoteries and hence wr-coteries.

A bicoterie $A = (\alpha, \beta)$ is **dominated** by bicoterie $B = (R, S)$ (or B dominates A) iff the following conditions hold:
1. $\forall Q_1 \in \alpha \; \exists \; Q_2 \in R$: $Q_2 \subseteq Q_1$.
2. $\forall Q_1 \in \beta \; \exists \; Q_2 \in S$: $Q_2 \subseteq Q_1$.
3. $(\alpha, \beta) \neq (R, S)$. □

A bicoterie (wr-coterie) B is said to be **nondominated (ND)** if no bicoterie (wr-coterie) dominates B. □

Given a bicoterie over a network. When the entire network is operational, for each site s, there is a read(write) quorum such that the maximum distance to some sites in the quorum is minimum. Let us call this the read(write) delay for site s.

Definition 2: The delay of site s is the greater of its read and write delay. The maximum of delays of all sites is called the **max-delay** of the bicoterie. A **max-delay optimal bicoterie** is one that has max-delay less than or equal to that of any other bicoterie over the same network. □

4 Generalized HQC

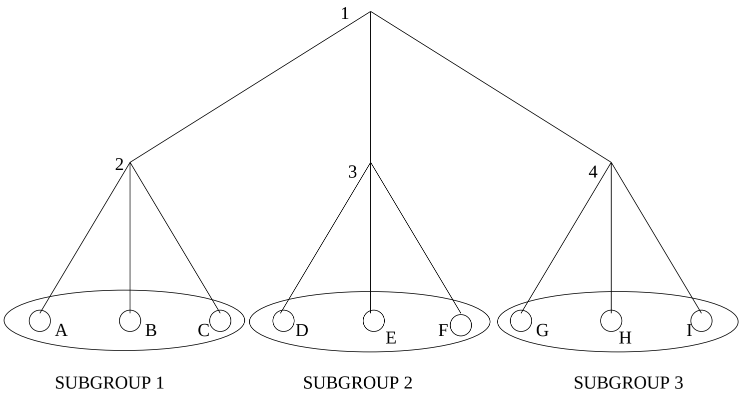


Figure 1: An example of nine copies organized into three subgroups

The hierarchical quorum consensus (HQC) [12] is based on logically organizing a set of copies of an objects in a database into a multilevel tree of depth m and with the root at level 0. The following is a description of this protocol from [12]: The physical copies of an object are stored only in the leaves of the tree (at level m). The higher level nodes of the tree correspond to logical groups. A node at level i, where i varies from 0 to $m-1$, is viewed as a logical group which in turn consists of l_{i+1} subgroups at level $i+1$. A quorum is associated with each level and to access a logical group at a certain level, a quorum formed by a majority of its subgroups must first be assembled. Figure 1 shows an example for nine copies of a data. The copies are labeled A, B, C, ... I. They are arranged in a hierarchy of 3 levels, in which each non-leaf node consists of 3 subgroups at the next level. There are 3 subgroups at the lowest level. At the root node (node 1), we may choose a majority quorum from nodes 2, 3, 4, which correspond to subgroup 1, subgroup 2 and subgroup 3 in the figure. For example, nodes 2 and 3 is a majority quorum. At node 2, the subgroups correspond to nodes A, B and C. Hence a majority quorum chosen at node 2 may be {A, B}. Examples of valid quorums are {A, B, D, E}, {B, C, H, I}, etc.

Generalized HQC: Here we point out that the hierarchical quorum consensus (HQC) method proposed in [12] can be generalized in the following way. In HQC, the coterie at each branching of the hierarchy is a majority consensus protocol. However, we observe that the choice of coterie at each branching does not affect the coterie properties of HQC. In other words, each of these coteries can be any known coterie, and so we can use different types of coteries in the hierarchy.

The hypercube coterie that we shall define in the next section will be a generalized HQC.

5 Hypercube Coterie and Wr-Coterie Protocol

In this protocol, we map sites to the nodes in a hypercube. An m^n-hypercube is a graph that consists of $N = m^n$ nodes. We say that an m^n-hypercube has n dimensions. We impose an ordering on the dimensions of the hypercube. Suppose the ordered set of dimensions are: $\{ D_1, D_2, D_3, ...D_n \}$. We assume that each dimension D_i has a set of m indices, $\{i_1, i_2, i_3, ...i_m\}$, which can be simply the set of numbers $\{1, ..., m\}$. Let the index set of dimension D_i be I_i. Each node of the m^n-hypercube can be expressed as a vector $\{ x_1, x_2, ...x_n \}$, where $x_i \in I_i$. There is an edge between nodes $\{ x_1, x_2, ...x_n \}$, and $\{ y_1, y_2, ...y_n \}$, iff $x_i \neq x_i$ for a particular i, where $1 \leq i \leq n$ and $q_j = p_j$ for $j \neq i$.

Example: For a 3^2-hypercube, there are 2 dimensions, and each dimension has a set of 3 indices. Suppose the indices are 1,2 and 3 in each dimension, then the nodes of the hypercube are A={1,1}, B={1,2}, C={1,3}, D={2,1}, E={2,2}, F={2,3}, G={3,1}, H={3,2}, I={3,3}.

We assume the sites in our distributed system has a one-to-one mapping to the nodes of a m^n-hypercube. However, note that the edges in the hypercube has no relation to the physical links between the sites.

Given a m^n-hypercube H as above, the first dimension D_1 has a set of indices $I_1 = \{i_1, i_2, i_3, ...i_m\}$. The subgraph with nodes $\{ y_1, y_2, ...y_n \}$ in the hypercube H where $y_1 = i_j$ for some j forms a m^{n-1}-hypercube G, whose index sets corresponds to the index sets of $I_2, I_3, ..., I_n$ of H. The node $\{ y_1, y_2, ..., y_n \}$ in H will be labeled $\{ y_2, ..., y_n \}$ in G. In other words, the 1st, 2nd, 3rd, ... , $(n-1)$-th dimensions of G correspond to the 2nd, 3rd, ... , n-th dimensions of H. We call G the sub-hypercube of H **subtended** by index i_j.

Example: Suppose the nodes of a 2^3-hypercube are given by {1,1,1}, {1,1,2}, {1,2,1}, {1,2,2}, {2,1,1}, {2,1,2}, {2,2,1}, {2,2,2}. Then the sub-hypercube subtended by index 1 in the first dimension has nodes {1,1}, {1,2}, {2,1}, {2,2}.

If we treat the index set that correspond to each dimension D_i in a m^n-hypercube as a set of sites, then we can assign a coterie C_i according to a certain quorum consensus protocol P_i. For example, it can be a majority quorum protocol, or a tree quorum protocol and so on. We say that coterie C_i is used for choosing quorums in the set of indices I_i, or we simply refer to C_i as **the coterie at dimension** i. We call such quorums **index-quorums**.

Example: For a 3^2-hypercube, suppose the indices for the first dimension are 1,2 and 3, and a majority quorum coterie is chosen for the first dimension, then a index-quorum for dimension 1 can be {1,2} or {1,3} or {2,3}.

An **(m^n)-hypercube quorum** is defined recursively as shown in Figure 2. A (m^n)-hypercube quorum will be a set of n-dimensional vectors which corresponds to nodes in the hypercube.[1]

[1]When the values of m and n are obvious, we simply refer to a (m^n)-hypercube quorum as a **hypercube**

Definition of a (m^n)-hypercube quorum for a (m^n)-hypercube H.
A (m^0)-hypercube quorum is ϕ.
For $n \geq 1$:
(1) Let A $= \{a_1, a_2, ..., a_k\}$ be an index quorum chosen from the coterie at the first dimension of H.
(2) Let $B_i =$ be an (m^{n-1})-hypercube quorum for the sub-hypercube subtended by a_i.
(3) For each element $\{c_1, c_2, ..., c_{n-1}\}$ in B_i, for each a_i, form the vector $\{a_i, c_1, c_2, ..., c_{n-1}\}$.
(4) The set of all vectors formed by the above step is a (m^n)-hypercube quorum.

Figure 2: Definition of a (m^n)-hypercube quorum

As an example, consider a data object with 27 copies, which correspond to nodes in 3^3-hypercube. Suppose the index set in each of the 3 dimensions is $\{1, 2, 3\}$. Following the procedure given in Figure 2, according to Line (1), we may choose $A = \{a_1, a_2\} = \{1, 3\}$ to be an index quorum for the first dimension ($D1$). By Line 2, we need to form two 3^2-hypercube quorums B_1 and $B2$.

Recursively, the 3^2-hypercube subtended by 1 is subjected to the procedure in Figure 2. From Line (91), we choose $A = \{a_1, a_2\} = \{1, 2\}$, to be an index quorum for the second dimension $D2$. In Line (2), we can pick $B_1 = \{\{1\}, \{2\}\}$, $B_2 = \{\{2\}, \{3\}\}$. From Line (3) and (4), the 3^2-hypercube quorum $\{\{1, 1\}, \{1, 2\}, \{2, 2\}, \{2, 3\}\}$, is constructed. This is returned to the first iteration as B_1.

Recursively, the 3^2-hypercube subtended by 3 in the first dimension is subjected to the procedure in Figure 2. From Line (1), we may choose $A = \{a_1, a_2\} = \{1, 3\}$, to be an index quorum for the second dimension $D2$. From Line (2), we let $B_1 = \{\{2\}, \{3\}\}$, $B_2 = \{\{1\}, \{3\}\}$. From Lines (3) and (4), the 3^2-hypercube quorum $\{\{1, 2\}, \{1, 3\}, \{3, 1\}, \{3, 3\}\}$ is constructed and this is returned to the first iteration as B_2.

Finally in the first iteration, by Lines (3) and (4) The set of sites { {1, 1, 1}, {1, 1, 2}, {1, 2, 2}, {1, 2, 3}, {3, 1, 2}, {3, 1, 3}, {3, 3, 1}, {3, 3, 3} } is the resulting 3^3-hypercube quorum.

This example can be shown more vividly by a diagram. Figure 3 shows the organization in a 3^3-hypercube. The 3 dimensions of the hypercube are labeled $D1, D2, D3$, and each dimension has a index set of {1,2,3}. Let we assign a majority quorum coterie to be the index coterie for each of the 3 dimensions. In the first dimension $D1$, we may choose indices 1 and 3 to be the majority index quorum. Then we need only consider the sub-hypercubes (3 *times* 3 planes) at the top and at the bottom of the figure. For the second dimension $D2$ of the top plane, let us choose indices 1 and 3 for the majority index quorum, we then need only to handle the corresponding 1-dimensional hypercubes (rows of size 3). For the second dimension $D2$ of the

quorum.

bottom plane, suppose we choose indices 1 and 2 as the index quorum, then again we need to handle the 2 corresponding 1-dimensional hypercubes. The resulting quorum of { {1, 1, 1}, {1, 1, 2}, {1, 2, 2}, {1, 2, 3}, {3, 1, 2}, {3, 1, 3}, {3, 3, 1}, {3, 3, 3} } is marked by crosses in the diagram.

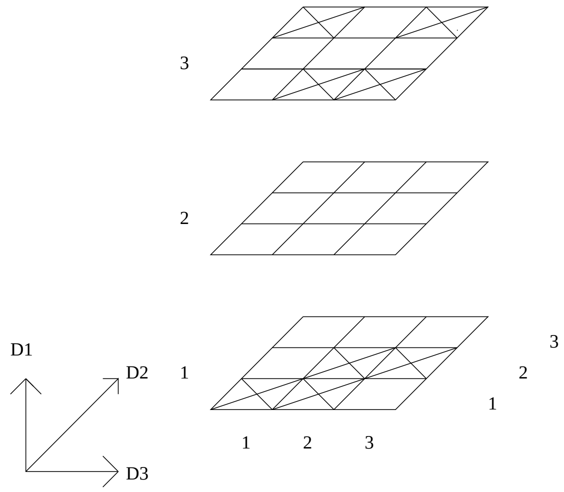


Figure 3: Constructing a hypercube majority quorum

Theorem 1 *: For a network with m^n sites, the (m^n)-hypercube quorums satisfy the properties of intersection and nonredundancy of coteries.*[2]

Proof: Given two of the hypercube quorums P and Q, we check the dimensions one by one. Let C_i be the coterie for dimension i in the (m^n)-hypercube. By the intersection property of C_1, we know that in the first dimension there is some intersection of P and Q in the first index, hence there are some nodes in P and in Q whose first dimension indices are the same. Let usch a common index be x_1. By the above definition, there will be a quorum chosen from C_2 in P and another chosen in Q for the 2nd dimension for the sub-hypercube subtended by x_1. By the intersection property of C_2, there will be some nodes in P and in Q whose first dimension indices equal x_1 and whose second dimension indices are the same, let such a common index be x_2. If we continue with this argument, then we see that in general, at the i-th dimension, there will be some nodes in P and in Q whose first i dimension indices are the same. Hence at the n-th dimension, there will be some node on P and on Q whose first n dimension indices are the same, which is an intersection node of P and Q.

At each dimension i, by non-redundancy of C_i, we choose non-redundant sets of indices, i.e. if P is an index quorum then no subset of P is also an index quorum. Therefore the final quorum formed is also non-redundant. □

A m^n**-hypercube coterie protocol** is a mutual exclusion protocol that accesses one m^n-hypercube quorum for each system operation. The hypercube quorum protocol is a special case

[2]The theorem can also be derived from results of Theorem 6.2 of [14].

of the generalized HQC since it is a hierarchy (tree) with $n+1$ levels and with the branching factor at each node kept at a constant m. Each dimension in the hypercube corresponds to one non-leaf level in the hierarchy.

5.1 m^n-hypercube Wr-Coteries Protocol

The wr-coteries are meant for handling the read and write operations of user transactions in a replicated distributed database. The m^n-hypercube wr-coterie protocol is quite similar to the m^n-hypercube coterie protocol above, except that at each dimension, instead of choosing just one index quorum we need to pick both an index quorum for read and also an index quorum for write, let us call these the m^n-hypercube read and write quorums. Hence we assign a wr-coterie to the set of indices at each dimension of the hypercube.

Theorem 2 *For a network with m^n sites, for the hypercube the (m^n)-hypercube read and write quorums satisfy the properties of intersection and nonredundancy of wr-coteries.*

Proof: Similar to the proof for Theorem 1. Omitted. ∎

5.2 Non-domination

We have seen that non-domination is a desirable property of a protocol. Here we shall show that if we choose to use ND coterie or wr-coteries in the hypercube quorum protocol, the resulting protocol is also ND. We apply the following theorem in our proof.

Theorem 2.1 of [9]: *Let S be a coterie under U. Coterie S is dominated iff there exists a group $G \subseteq U$ such that*
1. G is not a superset of any group in S.
2. G has the intersection property. That is, for all $H \in S$, $G \cap H \neq \phi$. ∎

This Theorem can be generalized for the wr-coteries as follows:

Theorem 3 *Let $A = (P, Q)$ be a wr-coterie under U, A is dominated iff there exists a group $G \subseteq U$ such that*
1. G is not a superset of any group in P, and
2. G has the intersection property. That is, for all $H \in P$, $G \cap H \neq \phi$.
or
3. G is not a superset of any group in Q, and
4. G has the intersection property. That is, for all $H \in P$, $G \cap H \neq \phi$.

We define two terms here to be used in the proof of the next theorem. Given a hypercube H and a set of nodes G, we say that an index x in H is **quorum-complete in** G if in the sub-hypercube of H subtended by x, there exists a hypercube quorum $\{q_1, q_2, ..., q_k\}$ such that the nodes $\{x, q_1\}$, $\{x, q_2\}$, ..., $\{x, q_k\}$, all exist in G.

Example: For the 3^3-hypercube given in Figure 3, and the set $G = \{ \{1, 1, 1\}, \{1, 1, 2\}, \{1, 2, 2\}, \{1, 2, 3\}, \{3, 1, 2\}, \{3, 1, 3\}, \{3, 3, 1\}, \{3, 3, 3\} \}$, the index 1 is quorum-complete in G, since there is a hypercube quorum $\{\{1, 1, 1\}, \{1, 1, 2\}, \{1, 2, 2\}, \{1, 2, 3\}\}$ in G for the the sub-hypercube of H subtended by 1.

Given a set of nodes $G = \{q_1, ..., q_k\}$ in hypercube H, the set G_{H_y} is a set of nodes in the sub-hypercube of H subtended by y, such that $x \in G_{H_y}$ iff $q_i = \{y, x\}$ for some i. We say that G_{H_y} is a **projection** of G. A projection of a projection of G is also called a projection of G.

Example: For the 3^3-hypercube H given in Figure 3, and the set $G = \{ \{1, 1, 1\}, \{1, 1, 2\}, \{1, 2, 2\}, \{1, 2, 3\}, \{3, 1, 2\}, \{3, 1, 3\}, \{3, 3, 1\}, \{3, 3, 3\} \}$, the set $G_{H1} = \{ \{ 1, 1\}, \{ 1, 2\}, \{ 2, 2\}, \{ 2, 3\} \}$.

Theorem 4 *For a (m^n)-hypercube coterie S, if the coterie C_i for each dimension i is nondominated, then S is nondominated.*

Proof: Assume that coterie S is dominated. By Theorem 2.1 in [9], there is a set of nodes G that satisfies the intersection property but is not a superset of any quorum in S. Let $A_1 = \{p_1, p_2, ..., p_k\}$ be the index set where p_i appears as the first index at some element in G. Let A'_1 be the subset of A which consists of all elements x that are quorum-complete in G. A'_1 cannot contain a quorum of coterie C_1 since otherwise G would be a superset of a quorum in S. Since coterie C_1 is non-dominated, there will be a quorum Q_1 in C_1 which A'_1 does not intersect. Each element a in Q_1, is not quorum-complete in G. We can find the projection G_{H_a} for such element a.

For such a set G_{H_y}, let $A_2 = \{q_1, q_2, ..., q_l\}$ be the index quorum where q_i appears as the first index at some element in G_{H_y}. If $A_2 = \phi$, then G does not interect with a quorum X that is formed by choosing Q_1 and this is a contradiction to the intersection property. Otherwise, let A'_2 be the subset of A_2 which consists of all elements y that are quorum-complete in G_{H_y}. A'_2 cannot contain a quorum of coterie C_2, so there must be a quorum Q_2 in C_2 which A'_2 does not intersect. Continue the argument for each dimension, we find a series of Q_i's in each case as in the above. At the n-th dimension, it reduces to the fact that some projection of G does not contain an intersecting site with a quorum Q_n, in a particular 1-dimensional hypercube. Form the m^n-hypercube quorum X by choosing these Q_i's at the corresponding sub-hypercubes at each dimension, we can see that G does not intersect X, a contradiction. ∎

Theorem 5 *For a (m^n)-hypercube wr-coterie S, if the wr-coterie C_i for each dimension i is nondominated, then S is nondominated.*

Proof: We need to show that it is not possible for a set G to exist as described in Theorem

3. That a set G cannot satisfy conditions 1 and 2 is proved as in the proof of Theorem 4. For the conditions 3 and 4, it means that there is a set of nodes G, such that G does not contain any read quorum in the original S, but G intersects each write quorum in S. The remaining part of the proof is again similar to that of Theorem 4. ∎

6 Optimal Communication Cost

Here we show how the protocol can achieve minimal communication delays in grid networks. The reason why we consider this is that in geographically distributed systems, the physical environment has only 2 dimensions, and a grid is a simpliest 2 dimensional layout.

6.1 Mapping a hypercube to a 2-dimensional surface

We describe here a way to map a (m^n)-hypercube to a $A \times B$ rectangle, where $A = m^a$ and $B = m^b$ and $a + b = n$. Imagine that the rectangle is a $A \times B$ grid so that each node in the grid corresponds to one node in the hypercube.

Initially there is only one sub-rectangle which contains all the nodes in the grid and the current dimension is 1. We keep dividing the sub-rectangles horizontally or vertically as follows, after each division, the current dimension is incremented by 1. This is repeated until all dimensions are handled (i.e. up to n). Finally, each sub-rectangle contains exactly one node.

- Horizontal division: divide each sub-rectangle into m equal sub-rectangles arranged in a horizontal row. Each node in the i-th sub-rectangle in the row has the index of the current dimension equal to i.

- Vertical division: divide each sub-rectangle in m equal sub-rectangles arranged in a vertical column. Each node in the i-th sub-rectangle of this column has the index of the current dimension equal to i.

An example is shown in Figures 4 and 5, where a 3^4-hypercube is matched to a 9×9 square by a horizontal division for dimension 1, vertical division for dimension 2, horizontal division for dimension 3, and vertical division for dimension 4. In the figures, a node indicated by *abcd* corresponds to the node $\{a, b, c, d\}$ in the 3^4-hypercube. The first three divisions are shown in Figure 4, in which a label 1xxx refers to nodes with first index being 1, and the remaining indices being 1, 2 or 3. The final result is shown in figure 5.

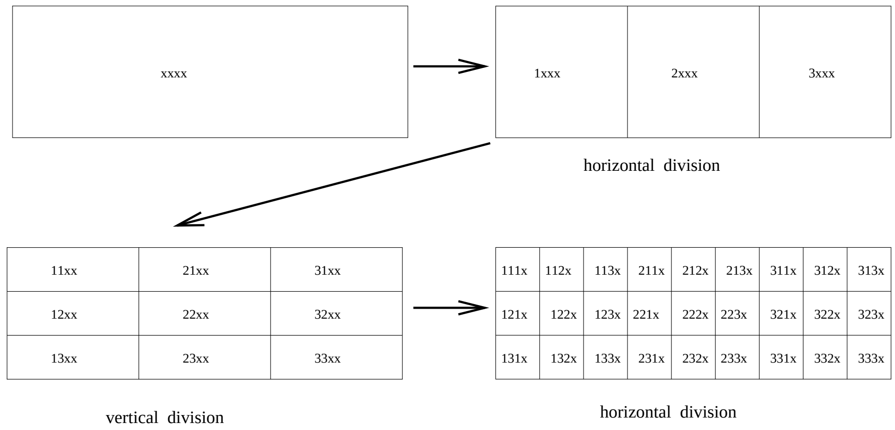


Figure 4: Example of mapping hypercube to 2-dimensional rectangle

1111	*1121*	1131	*2111*	*2121*	2131	3111	3121	3131
1112	*1122*	1132	*2112*	*2122*	2132	3112	3122	3132
1113	1123	1133	2113	2123	2133	3113	3123	3133
1211	*1221*	1231	*2211*	*2221*	2231	3211	3221	3231
1212	*1222*	1232	*2212*	*2222*	2232	3212	3222	3232
1213	1223	1233	2213	2223	2233	3213	3223	3233
1311	1321	1331	2311	2321	2331	3311	3321	3331
1312	1322	1332	2312	2322	2332	3312	3322	3332
1313	1323	1333	2313	2323	2333	3313	3323	3333

Figure 5: Result of Mapping a 3^4 hypercube to a 9×9 square

6.2 Majority Hypercube Coteries

If we let each coterie in each dimension of the hypercube be a majority quorum coterie, then the resulting hypercube coterie is called a **majority hypercube coterie**. We call a special case of the majority hypercube coterie the **cross-product hypercube coterie** when each quorum is chosen as follows: choose a majority out of the indices of each dimension in the hypercube, form a cross product of all the majority sets, elements in the cross product are elements of the quorum. For example for a 3^4-hypercube, suppose the indices for each dimensions are 1,2,3. We can choose the majority quorum of {1,2} for each dimension. The cross product will be P = { {1,1,1,1}, {1,1,1,2}, {1,1,2,1}, {1,1,2,2}, {1,2,1,1}, {1,2,1,2}, {1,2,2,1}, {1,2,2,2}, {2,1,1,1}, {2,1,1,2}, {2,1,2,1}, {2,1,2,2}, {2,2,1,1}, {2,2,1,2}, {2,2,2,1}, {2,2,2,2} }.

The difference of the cross-product hypercube coterie from the general majority hypercube coterie is that the choice of index-quorums is more restrictive. For example, the quorum in Figure 2 is not a cross-product hypercube quorum: the index quorum chosen in the second dimension in the top square is different from that in the bottom square.

The mapping to grid networks that we described above can minimize the communication delays between sites in choosing appropriate cross-product hypercube quorum. For the example in Figure 5, the cross-product hypercube quorum P achieves minimum delay and is shown in italic font. In general minimum delay can be achieved if the majority quorums chosen for each dimension are contiguous indices, e,g. 1,2, ..., $\lceil n/2 \rceil$ or $\lceil n/2 \rceil$, ..., n, where n is the number of dimensions.

Theorem 6 *A cross-product hypercube coterie is a max-delay optimal coterie for the grid network as constructed above.*

Proof : By the property of coteries, the quorums for the sites at two opposite corners must intersect. To achieve minimum delay for both sites, the intersecting site should be in the middle (or near the middle) along the diagonal joining the two sites. Hence the minimal max-delay of a coterie on the grid has value at least equal to this delay, let it be X. We see that in the cross-product hypercube, each site s can choose a quorum coterie which lies within a rectangle with distance between opposite corners being X, and s being within the rectangle. ■

Corollary 1 *The majority hypercube coterie is a max-delay optimal coterie for the grid network as constructed above.*

The above corollary follows because a cross-product hypercube coterie is also a majority hypercube coterie, so that whenever we need to pick a majority hypercube quorum, we can pick a cross-product hypercube quorum. Note that we are more interested in the majority hypercube coterie instead of the cross-product hypercube coterie because it would give us higher availability.

6.3 Optimal Hypercube Wr-coterie Protocol

The following lemma helps us to reduce the problem of finding a max-delay optimal wr-coterie to that of finding a max-delay optimal coterie.

Lemma 1 *For a given network, the max-delay of a max-delay optimal coterie is less than or equal to the max-delay of a max-delay optimal wr-coterie.*

Proof: If the max-delay optimal bicoterie is $\{P, Q\}$, then we can form a coterie with Q, and the max-delay of Q is less than or equal to the max-delay of $\{P, Q\}$. ■

From the above lemma, given a network, after finding a max-delay optimal coterie P, a max-delay optimal wr-coterie will be $\{P, P\}$.

Corollary 2 *The (m^n)-hypercube wr-coterie formed by using the majority hypercube coterie to determine both read and write index quorums at each dimension of the hypercube is a max-delay optimal wr-coterie for a grid network.*

7 Quorum Size and Availability Analysis

Besides communication delay, quorum size and availability are two important properties for coteries. These factors are evaluated in this section. Let N be the number of sites in the network, the smallest possible size for a HQC is given by $N^{0.63}$ [12] by setting the branching factor in the hierarchy to 3.[3] Such a hierarchy can be seen as a (3^k)-hypercube, for $k = log_3 N$. Therefore the smallest quorum size for a majority hypercube coterie for a (3^k)-hypercube is also $N^{0.63}$.

We consider availability of a coterie or wr-coterie to be the probability that at least one quorum is operational. We assume that the probability that any given site is up is p.

7.1 Availability analysis of the majority hypercube protocol

In this sub-section, we compare the availability of the majority hypercube protocol [1] with other protocols, namely the majority quorum protocol and the grid quorum protocol [7, 13].

For a (m^n)-hypercube, there are m^n sites. Let $Avail_i$ be the probability that we find a majority hypercube quorum in the i-th dimensional sub-hypercube. $Avail_1$ is simply the chance of finding an operational majority quorum in a 1 dimensional array. $Avail_n$ will be the required availability. $Avail_i$ is defined recursively in the following equations.

$$\begin{aligned} f_m(p) &= \sum_{i=\lceil \frac{m+1}{2} \rceil}^{m} \binom{m}{i} p^i (1-p)^{m-i} \\ Avail_1 &= f_m(p) \\ Avail_{i+1} &= f_m(Avail_i) \end{aligned}$$

The formulae for the majority quorum protocol can be found in [1]. The formulae for the grid quorum protocol can be found in [13]. Note that the grid quorum protocol distinguishes between "read quorums" and "write quorums" for the application of read and write operations in replicated databases. The grid protocol has different availabilities for read quorums and write quorums.

We compare the availabilities of the three protocols with 27 sites and 243 sites in the model. We compare the availabilities of the three protocols by varying the probability p that a particular site is up. For the grid protocol, we use a 9×3 grid (9 rows of 3 sites each) for the case of 27

[3] This is because the size of a quorum at each level is 2, and there are $log_3 N$ level, hence the total quorum size is $2^{log_3 N} = N^{0.63}$

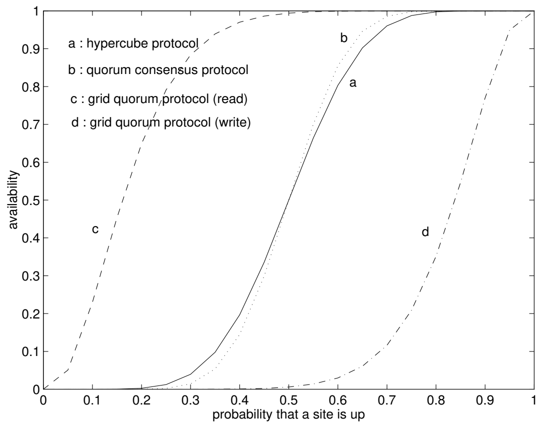


Figure 6: Availability of protocols with 27 sites

sites, and a 9×27 grid for the case of 243 sites. The results are shown in Figures 6 and 7, with the probability that a sites is up being plotted against the availabilities of the protocols.

As we can see from the graphs, the hypercube protocol and the majority quorum protocol are quite close in performance. As the value of p decreases beyond 0.6, the availabilities of both protocols dropped significantly until $p = 0.4$. This shows that the two protocols perform much better for the value of $p > 0.6$. For the grid quorum protocol, the availability of the read quorum is better than the hypercube and the majority quorum protocols. However, the availability of the write quorum is worse than the two protocols.

Although the majority quorum protocol has the best performance in availability, it has the major drawback of having a large quorum size of over half the number of sites. Also when $p > 0.9$, the availability of the hypercube quorum protocol is similar to that of the majority quorum protocol, and with current technology site availability is typically above 0.9, meaning that $p > 0.9$ can be assumed.

We believe that the probability that a site is up is in general high. It is unusual to have the value of p below 0.9 in an acceptable system. Since the availabilities of the three protocols with $p > 0.9$ are all very high, we may conclude that all of them are reliable under real world situations.

7.1.1 Availability of the Protocol with Different Dimensions

Now we try to examine the availability of the system if we have different combinations of the value of m and n for a m^n-hypercube given m^n sites. Suppose we have a total number of 81 sites and so we can choose either $m = 3$ and $n = 4$ or the combination $m = 9$ and $n = 2$. For each of the cases, we plotted the availability of the system against the value of p. The result is shown in Figure 8.

Figure 7: Availability of protocols with 243 sites

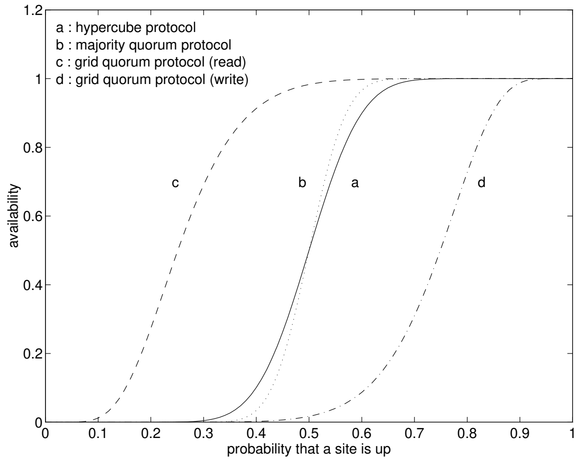
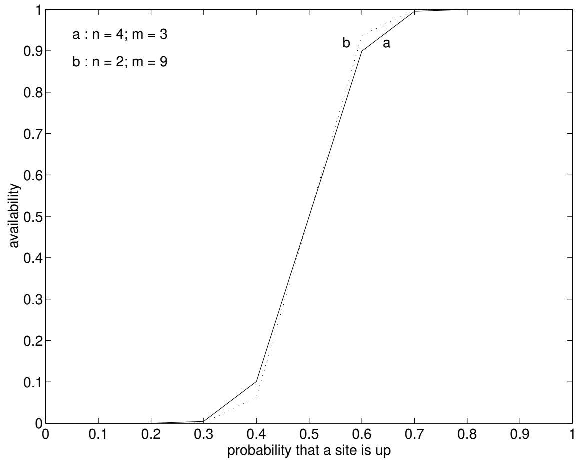


Figure 8: Comparison of Hypercube Protocol with Equal Number of Sites



We found that for the same number of sites in the system, if we increase the number of indices in each dimension(m), and hence decrease the number of dimension(n), better availability of the system may be achieved.

Now we come up with a trade-off. The set up of $m = 3, n = 4$ gives a smaller quorum size, but the availability is a little worse than $m = 9, n = 2$ when p is between 0.3 and 0.7. However, in practical environments, the availability of each site, p, is typically high (above 0.9), we can see that the availabilities of the two set-ups in such cases are similar.

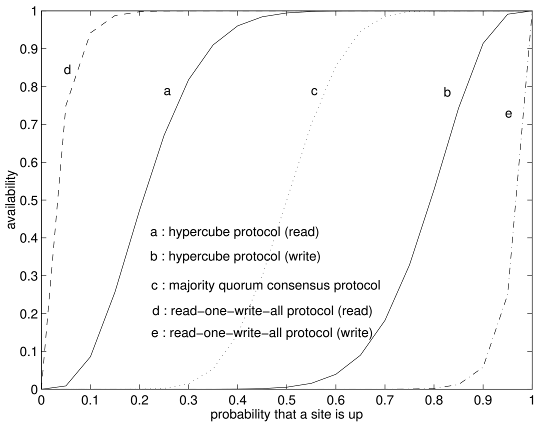


Figure 9: Availability of hybridized protocol with 27 sites

8 Hybridization of protocols

In the generalized HQC, one can have a combination of coteries in different dimensions of the hypercube, we examine the possible consequence in this section. It is known that some protocols, such as the grid protocol or the read-one write-all protocol, have very high availability for their read quorums. Therefore we try to replace the majority quorum protocol used in the previous subsections with the read-one write-all protocol on the first dimension of the hypercube to see if it can improve the read availability of the protocol. The majority quorum protocol is used in the other dimensions as before. Our model is composed of 27 sites, being organized into a (3^3)-hypercube. Figure 9 shows the availability of the hypercube protocol.

From the graphs, the *new* hypercube protocols outperform the original protocol in terms of the read availability. This shows that the hypercube protocol can adopt protocols that are known to behave well in a certain way in order to attain some of their characteristics.

9 Conclusion

In this paper, we present a hypercube quorum consensus protocol. This is a generalization of the hierarchical quorum consensus (HQC) protocol proposed in [12]. We show that our proposed method have the following advantages: (1) The protocol is shown to be non-dominated. (2) The protocol can achieve optimal communication delay in grid network topologies. (3) The quorum size of the protocol is small (with a smallest size of $N^{0.63}$ as for HQC, where N is the number of sites). (4) The protocol can achieve high availability in realistic environments.

We present not only the protocol but also the way that the hypercube can be mapped to a 2-dimensional surface so as to achieve minimum communication delay. This is important because the geographical sites in real world are distributed in a 2-dimensional way, and it is necessary for the hypercube model to map to the real world environment.

One interesting characteristic of the hypercube quorum protocol or the generalized HQC is that the quorum consensus protocol used at each dimension or each branching of the hierarchy can vary. Our finding is that we can apply protocols of known characteristics to some dimensions and obtain some of the corresponding characteristics in the resulting hypercube protocol. Since quite a number of different quorum consensus protocols are known, and most of them have some advantages over others, we may try hybridization of different protocols on the hypercube quorum consensus to derive desirable properties for a given application.

In the future, we may study the effect of *holes* in the hypercube. Currently, we assume that each node in the hypercube corresponds to a physical site. However, we may design a hypercube in which some nodes do not correspond to physical sites. A hole corresponds to a logical node in the hypercube which does not correspond to a physical site. The role of holes can be taken over by neighboring nodes. It would be of interest to find out the effects of such holes on the availabilities and other aspects of the protocol. We may also investigate another generalization of the HQC, which is to vary the branching factor of nodes located at the same level of the hierarchy.

Acknowledgements

We would like to thank the referee who has thoroughly read the initial version and given us very valuable comments and suggestions to greatly enhance the readability of the paper.

References

[1] D. Agrawal, A. El Abbadi, *An Efficient and Fault-Tolerant Solution for Distributed Mutual Exclusion*, ACM Transactions on Computer Systems, Vol.9, No.1, February 1991, pages 1-20.

[2] D. Agrawal, A. El. Abbadi, *The Generalized Tree Quorum Protocol : An Efficient Approach for Managing Replicated Data*, ACM Transactions on Database Systems, Vol. 17, No. 4, 1992, pages 689-717.

[3] P. A. Bernstein, N. Goodman, *The Failure and Recovery Problem for Replicated Databases*, ACM Transactions on Computer Systems, 1983.

[4] A. El Abbadi, S. Toueg, *Maintaining Availability in Partitioned Replicated Databases*, ACM Transactions on Database Systems, Vol.14, No.2, 1989, pages 264-290.

[5] P. A. Bernstein, N. Goodman, *An Algorithm for Concurrency Control and Recovery in Replicated Distributed Databases*, ACM Transactions on Database Systems, Vol.9, No.4, 1984, pages 596-615.

[6] Berstein P. A., V. Hadzilacos, N. Goodman, *Concurrency Control and Recovery in Database Systems*, Addison-Wesley, Reading Mass, 1987, Chapter 8.

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

[8] A. Fu, *Delay Optimal Quorum Consensus for Distributed Systems*, accepted for publication in IEEE Transactions on Parallel and Distributed Systems.

[9] H. Garcia-Molina, D. Barbara, *How to Assign Votes in a Distributed System*, Journal of the Association for Computing Machinery, Vol.32, No.4, 1985, pages 841-860.

[10] M. Herlihy, *Dynamic Quorum Adjustment for Partitioned Data*, ACM Transactions on Database Systems, Vol.12, No.2, 1987, pages 170-194.

[11] T. Ibaraki, T. Kameda, *A theory of coteries : mutual exclusion in distributed systems*, IEEE Transaction Parallel and Distributed Systems, Vol.4, No.7, 1993, pages 779-794.

[12] A. Kumar, *Hierarchical Quorum Consensus: A New Algorithm for Managing Replicated Data*, IEEE Transactions on Computers, Vol. 40, No. 9, 1991, pages 996-1004.

[13] A. Kumar, M. Rabinovich, R. K. Sinha, *A Performance Study of General Grid Structures for Replicated Data*, 13th Int. Conf. on Distributed Computing Systems 1993.

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

[15] C.H. Papadimitriou, M. Sideri, *Optimal Coteries*, ACM 10th Symposium on Principles of Distributed Computing, 1991, pages 75-80.