A Universal Quorum Protocol for N-Dimensional Structures*

Robert Schadek and Oliver Theel
Carl von Ossietzky University of Oldenburg
Department of Computer Science
D-26111 Oldenburg, Germany
{*robert.schadek,theel*}*@informatik.uni-oldenburg.de*

Abstract—*Existing structured quorum protocols usually exploit topologies to achieve higher availabilities and lower costs for read and write operations. But these topologies often do not match the existing physical network topologies or the topologies are just a logical level on top of them. Therefore, structured quorum protocols either simply do not work as expected for a given network topology wrt. operation availabilities and costs as they do not take the impact of the underlying physical network into consideration. Contrarily, we present a quorum protocol that works right away on any given arbitrary topology of a physical network and that can directly be analyzed wrt. the operation availabilities and costs that occur based on this particular topology used. The protocol is universally applicable not only for two- but also for general* n*-dimensional topologies.*

Keywords—Distributed Systems, Fault Tolerance, Structured Quorum Protocols, Data Replication, Distributed Mutual Exclusion, Operation Availability, Operation Costs

I. Introduction

Quorum protocols have a wide range of applications. They have been used, for example, for data replication [4] and distributed mutual exclusion [9]. Usually, quorum protocols, in particular structured ones, require a certain topology defined among the replicas[1] to work and to exhibit the protocol-specific operation availabilities and costs. These topologies are usually assumed being "logical" ones, i.e., they are constructed on top of existing physical networks of nodes hosting replicas. Thus, in general, these logical topologies do not match the physical network topologies in the sense that a connection between two replicas in the logical topology has a one-to-one relationship with a connection between two nodes in the physical network that host these two replicas. In fact, it can well be that two replicas in the logical topology are only connected via additional other nodes (may they host replicas or not) of the physical network with one another. Thus, whenever such an intermediate node fails, replicas on the logical level are actually separated from one another although an analysis of the quorum protocol, performed on the logical topology only, might still regard these two replicas as connected. This might result in incorrect operation availability and costs analysis results wrt. to the behavior exhibited by the quorum protocol actually used in the physical topology. In other words, a system using data replication is often designed and analyzed in terms the operation availabilities and costs of the quorum protocol only, and not in terms of the *combination* of the logical and the underlying physical network topology. Therefore, the analysis results might very easily lead the system designer astray.

Examples of structured quorum protocols are the generalized tree quorum (GTQ) protocol [1] and the Triangle Lattice (TL) protocol [13]. The TL protocol requires a triangulated grid as logical topology to work. Interestingly, the optimal quorum size of the write operation turned out to be $O(\sqrt{N})$, where N is the number of replicas. This means that in order to have optimal quorums, the grid must be a square. Such a topology is unlikely to be found in the real world as a network; it is much more likely, for example, to find a star topology as in a single router network. As an example, consider a network with 16 replicas, each having an availability of 0.9 arranged in a triangulated square. The TL

* This work has been partially supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS).

[1] In the discussion to follow, we concentrate on the application of quorums in the scope of data replication.

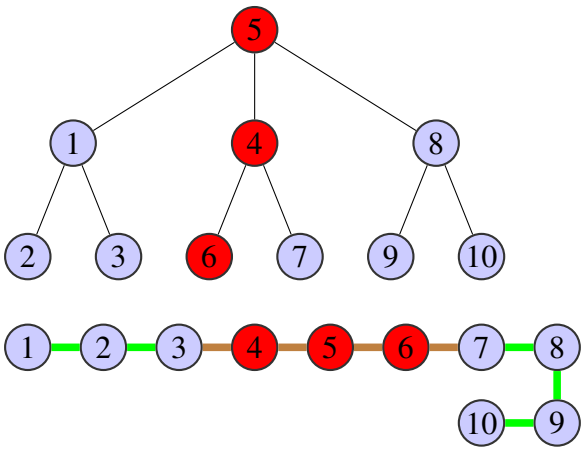

Fig. 1. GTQ protocol which uses a logical tree topology (top) mapped onto a bus-like network (bottom). The blue circles represent (nodes hosting) replicas. The (nodes hosting) replicas in red have failed.

protocol, when analyzed in the context of this logical topology only, exhibits an availability of 0.9999 for the read and of 0.997 for the write operation. But if the underlying network is a star topology and the router's availability is 0.7, then the entire system exhibits an availability is $0.7 * 0.9999 = 0.6999$ for the read and $0.7 * 0.997 = 0.6979$ for the write operation, i.e., much lower availabilities. Thus, as shown, the TL protocol can only exhibit its high operation availabilities and low operation costs, when the very specific logical topology truly maps onto the nodes hosting replicas and the communication channels (i.e., connections) among them. For real-world networks, as structured quorum protocols use very specific, often very regular logical topologies, this will very likely not often be the case.

The stated matching problem is also present in the GTQ protocol even though a tree topology seems more suitable for real-world use. The GTQ protocol arranges replicas in a logical tree. Again, the problem lies in the purely logical nature of the used topology abstracting from the physical topology. Figure 1 shows this graphically. The GTQ protocol requires for correct functioning that if the root replica, in this case replica 5, is not available for reading, then a majority of replicas of the next lower level should be used for the read operation. This means that replicas 1 and 8 must be read. According to the GTQ protocol rules, this seems possible. But when taking the state of the underlying network into consideration, it actually turns out that this is currently impossible, since the nodes hosting replicas 1 and 8 belong to different network partitions. The reader might verify, that currently no read (or write) operation can be performed, although, on the logical level, several read quorums required for consistent read access seem to be available.

Prominent operation cost measures in the context of quorum protocols are quorum cardinalities (i.e., sizes) or minimal or average quorum sizes or both. For example, when using quorum cardinalities, the read quorum consisting of replicas 1 and 8, obviously, causes costs of 2 (using the GTQ protocol shown in Figure 1 as an example). Often, these cost measures are also easily misleading when using such a protocol in the scope of a real-world network topology: in the example, contacting nodes hosting replicas 1 and 8, in a failure-free network, requires at least seven messages be sent via communication channels. A read quorum of replicas 2, 3, 9, and 10 (quorum cardinality of 4) might only require five messages for establishing contact. Clearly, it depends on the cost measure used, but – as shown – cost measures analyzed on the logical level might be too abstract for actually judging the costs truly arising on the physical level. Again, a system designer might easily choose an inadequate quorum protocol based on misleading cost analyses done in the context of a logical topology.

In order to bridge this gap between incompatible operation availability and cost analyses of the different topological levels, we present a quorum protocol that works on arbitrary (physical) topologies of arbitrary dimension, right away. As our protocol does not exploit any topology beside the physical one, the mapping among topologies as well as the discrepancies in operation availability and cost analyses become a non-issue.

In the next section, we first present the basic idea behind the new quorum protocol.

In Section III, we show how the protocol can be generalized to N dimensions. In Section IV, we relate our approach to other work found in literature. Finally, in Section V, we conclude the paper and sketch directions of further related research.

II. The New Protocol

The protocol we propose, works on an *a priori* given arbitrary topology. This topology is neither modified nor superseded by any other (logical) topology which the protocol might use instead and which could potentially lead to "mapping problems." The basic rationale behind this is, that the protocol should be able to directly use the topology of the physical network, it is intended to finally work in, thereby avoiding the pit-falls discussed in the previous section. Nevertheless, the protocol should be able to somehow exploit the particular characteristics of the topology given, in order to provide highly available and cheap read and write operations on a replicated data object. Since not too much can be assumed about an arbitrary topology, achieving this might appear to be a tough aim.

Our approach is as follows. We assume that every node of the topology hosts one copy of the replicated data object. As usually, such a copy is called "replica."

In the first step, we partition the replicas in two classes based on their location in the topology. The partitioning is governed by the following rules which are given in an intuitive, informal manner:

Border replicas: The class of *border replicas* contains all replicas of the topology that – in combination with their edges – form the "outside" or, more formally, the geometric hull of the topology.

Inner replicas: The class of *inner replicas* contains all replicas that are not border replicas. Thus, inner replicas do not lie on the outside, i.e., border of the topology. Instead, they form the "interior" of the topology.

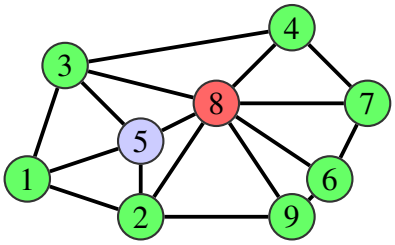

Fig. 2. Two-dimensional example topology showing border and inner replicas as well as the middle replica

Additionally, one replica must be selected as a distinguished replica:

Middle replica: Exactly one replica of one of the former two classes of replicas is chosen as the so-called *middle replica.*

It is intended that the particular replica chosen as middle replica lies in the "center of the topology" due to reasons discussed in the subsequent paragraphs. Thus, the middle replica should be a member of the class of inner replicas. Note that – although it appears to be an oxymoron – the protocol also functions correctly, if the middle replica belongs to the border replica class.

Figure 2 illustrates the concepts introduced by a two-dimensional example topology. Green replicas represent the class of border replicas. Red or blue replicas belong to the inner replica class. The red replica was chosen as middle replica.

In the second step, based on the selection of the middle replica, the protocol's quorums are constructed. In contrast to many other data replication protocols, the new protocol presented does not use different types of quorums for read and write operation execution. Thus, only one type of quorums exists and any of the protocol's quorums can be used for executing a read or a write operation in a one-copy serializable manner.

For a better understanding and due to visualization reasons of the accompanying examples, we restrict the discussions of how the quorums are constructed to the two-dimensional case. A generalization to higher dimensions is postponed until Section III. The idea behind the construction of quorums

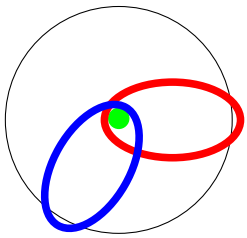

Fig. 3. Basic idea of the new protocol: any two circles around the middle that also each touch the border always intersect

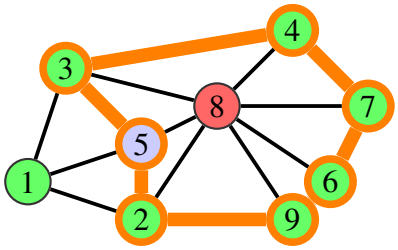

Fig. 4. A circle around the middle replica that touches the border of the topology. The orange outlined replicas represent a valid quorum. The orange edges between them represent the (closed) path through the graph to combine them.

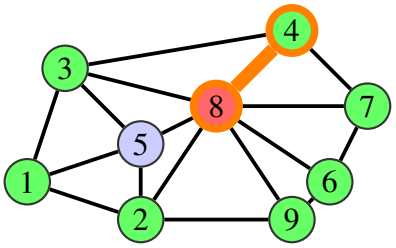

Fig. 5. A minimized circle that touches the middle replica as well as the border. Again, the orange outlined replicas form the quorum. The orange edges between them represent the path through the graph to combine them.

is to create a hull around the middle replica which "touches the border of the topology" as sketched in Figure 3 and to use the replicas belonging to that hull as a quorum. In the figure, the middle replica is given as green dot and the border replicas are abstracted to the black circle. Replicas forming the red or blue or black circle represent a valid quorum of the protocol. Note, that one-copy serializability is guaranteed, if any two circles intersect and an intersection of two circles implies the nonempty intersection of the two quorums represented by the two circles in at least one replica. Obviously, on the one hand side, the more circles of this kind exist, the higher available are read and write operations. On the other hand side, the larger – in terms of the number of replicas – such a circle is, the higher are the communication costs (in terms of messages sent between nodes) associated with this quorum and the lower is the probability that this quorum actually will function at a particular point in time.

A. Exemplary Approach of Finding a Quorum

As stated before, the basic idea is to find a circle in the topology that touches the border and encloses the middle replica. The orange path in Figure 4 shows such a circle. Again, as in Figure 2, the green replicas form the border, the red replica represents the middle replica and the blue replica is an inner one. The orange path connecting the replicas $2, 5, 3, 4, 7, 6$ and 9 form a quorum. This quorum, although not the most efficient one in the scope of the example, allows to accurately reason about operation availabilities and costs when the topology resembles the physical network topology. The operation costs arising when using this quorum may simply be number of replicas in the quorum. The operation availability based on this quorum only, is simply the multiplied availability of each individual replica when availabilities of the communication channels are assumed being 1. This simplification of the associated fault model is often assumed in related literature, since it eases the (approximate) calculation of operation availabilities.

In order to allow for very small quorums, we conceptionally "tighten" the circle around the middle replica and replica 4 of Figure 4 in a way such that its result is shown in Figure 5. This procedure can be understood as a rubber band tightening around these two points. The resulting path, from a border replica to the middle replica, may – in the optimal case – only include two replicas. This is the case in the example, where only replicas 8 (middle replica) and 4 (border replica) form the associated quorum. In the scope of the protocol, these "deflated circles" count as valid circles; their associated quorums as valid quorums since they intersect with the

other quorums of the protocol in a non-empty manner. Note, that not all "deflated circles" result in quorums of size 2: a possible outcome may also be a circle consisting of the replicas $4,7,6,9,2,5,1,3,8$ in the example given in the figure. The associated quorum is by no means optimal in terms of availability and costs.

B. Border Replicas

Identifying the replicas belonging to the border class of a given topology is an important step in our protocol. For this, it is required that the replicas have static positions in the topology. Dynamic change of topology is not allowed.

The trivial case is a topology consisting of one replica only. Here, the only replica must belong to the border class whereas the inner class is empty. Obviously, the only replica must act as middle replica.

For a topologies with two dimensions, our algorithm works in two stages. In the first stage, the convex hull is constructed. This convex hull has no notion of edges. In the next step, an arbitrary replica of the convex hull is chosen as a starting point to traverse the border edges. Traversing the border replicas is done by following the next adjoining edge with the highest angle to the current edge. If no more edges are adjoint, then the current edge is traversed in opposite direction. This process is repeated until the replica serving as starting point is visited again. All the replicas visited in the course of this algorithm belong to the class of border replicas.

C. Middle Replica

As the middle replica can be any replica of a given topology, the particular choice has a very severe influence on the operation availabilities and costs of the resulting instance of the quorum protocol. If, for instance, the middle replica in Figure 2 is border replica 4, then the operation availability of the protocol decreases drastically but so do the costs. As replica 4 is a border replica, there is obviously no way to construct an enclosing circle around this middle replica. Consequently, the middle replica must be included in every quorum. Actually, middle replica 4, being also a border replica, can actually form a quorum all by itself. Thus, in this example, the operation availability of the system cannot exceed the availability of this particular middle replica, rendering the replication approach practically useless. Varying the middle replica is obviously an elegant and efficient way of adjusting operation costs and availabilities of the protocol.

III. Adding Dimensions

Before we add dimensions, we are first going to reduce them. This is in order to show how the proposed protocol works for topologies with dimension of less than two. For the one-dimensional case, consider the bus shown in Figure 1. In order to position a replica, a vector of dimension 1 is sufficient. This leads to a topology where every replica is a border replica. Any replica can be chosen as middle replica, still there is no difference wrt. operation availability and minimal operation costs (in terms of minimal quorum cardinality) as the middle replica must be part of every quorum. As every replica is also a border replica, the smallest quorum only contains the middle replica. If the middle replica is not located at either end of the bus-like network, its failure leads to a network partitioning [10].

As described in [7], a concave hull can be found for any n-dimensional topology [6]. The replicas of such a concave hull can act as the border replicas in the scope of the proposed protocol. Together with the choice of a middle replica, this structure can be used to specify "circles of higher dimension" around the middle replica and including at least one border replica. These circles, obviously, represent quorums valid for consistent operation execution.

IV. Related Work

Existing quorum protocols like the *Majority Consensus Protocol* [5] or the generalized

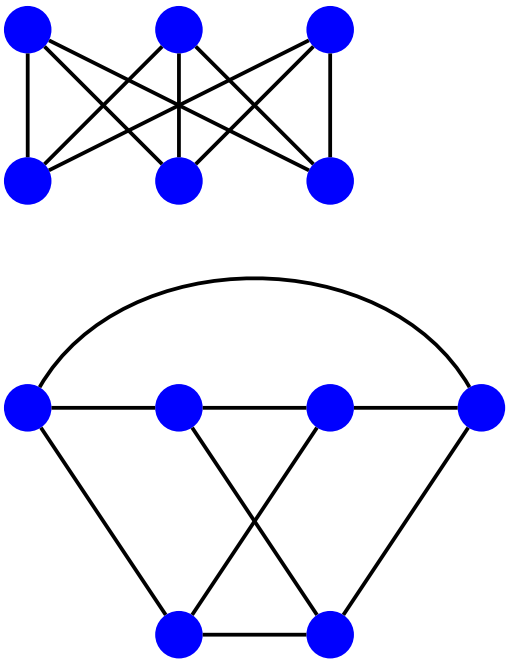

Fig. 6. $K_{3,3}$ non-planar graph and $K_{3,3}$ with minimal edge crossing

Weighted Voting approach [11] do not use the concept of topologies. Protocols that do use topologies are limited to very specific ones like the *Tree Quorum Protocol* [2] the *Generalized Tree Quorum Protocol* [1] or the *Grid Protocol* [3]. The TL protocol, too, only works on one particular topology [13].

A more general structured quorum strategy is the *Crumbling Wall Protocol* [8]. This protocol requires that replicas are arranged in rows and columns. The number of rows or the number of replicas per row can freely be chosen. This allows the protocol to work on all tree- or mesh-like structures, but it is not sufficient to represent arbitrary graphs. For instance, the non-planar graph in Figure 6 cannot be expressed in a row/column topology. This graph is known as $K_{3,3}$. If this graph or a graph called K_5 is found as a subgraph, the containing graph is not planar. This is known as the *Kuratowski Theorem* [12]. Therefore, not all two-dimensional topologies can be projected on the planar row/column topology of the Crumbling Wall Protocol.

V. Future Work and Conclusion

In this paper, we presented a quorum protocol that works on arbitrary topologies of all possible dimensions. The motivation for this protocol stemmed from the observation that existing quorum protocols are misleading in their operation availability and costs measures when being mapped from the logical to real-world topologies like physical networks, the protocol has to execute within. The proposed protocol does not have this drawback. But preliminary analysis results indicate that the protocol yields comparable operation costs and availabilities when applied to logical structures used by existing protocols and performs superior when applied to real-world topologies.

Future work will include a thorough-fully evaluation of the protocol proposed together with a comparison to existing quorum protocols. Furthermore, we will try to identify specially suited topologies. We believe this to be important even though the proposed protocol conceptionally works on arbitrary topologies: by modifying real-world physical networks such that they coincide with specially suited ones, will increase operation availabilities as well as reduce certain cost measures. Furthermore, we will work on the construction of a circle search algorithm that identifies the smallest circle that "touches the border" and embeds the middle replica right away in contrast to the current algorithm, that walks the border if possible and *a posteriori* checks whether the middle replica is indeed embedded. This will allow to speed up the brute-force analysis of the protocol.

References

[1] D. Agrawal and A. El Abbadi. The generalized tree quorum protocol: an efficient approach for managing replicated data. *ACM Trans. Database Syst.*, 17(4):689–717, December 1992.

[2] Divyakant Agrawal and Amr El Abbadi. The tree quorum protocol: An efficient approach for managing replicated data. In *Proceedings of the 16th International Conference on Very Large Data Bases*, VLDB '90, pages 243–254, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[3] Shun Yan Cheung, Mostafa H. Ammar, and Mustaque Ahamad. The grid protocol: A high performance scheme for maintaining replicated data. *Knowledge and Data Engineering, IEEE Transactions on*, 4(6):582–592, 1992.

[4] Susan B. Davidson, Hector Garcia-Molina, and Dale Skeen. Consistency in partitioned networks. *ACM Comput. Surv.*, (3):341–370, 1985.

[5] David K. Gifford. Weighted voting for replicated data. In *Proceedings of the seventh ACM symposium on Operating systems principles*, SOSP '79, pages 150–162, New York, NY, USA, 1979. ACM.

[6] Bruce Kleiner and John Lott. Notes on perelman's papers. *Geometry and Topology*, pages 2587–2855, 2008.

[7] Jin-Seo Park and Se-Jong Oh. A new concave hull algorithm and concaveness measure for n-dimensional datasets. *J. Inf. Sci. Eng.*, 28(3):587–600, 2012.

[8] David Peleg and Avishai Wool. Crumbling walls: a class of practical and efficient quorum systems. In *Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing*, PODC '95, pages 120–129, New York, NY, USA, 1995. ACM.

[9] Michel Raynal. Algorithms for mutual exclusion. *The MIT Press, Cambridge, MA*, 1986.

[10] Lynn Arthur Steen and J. Arthur Seebach Jr. *Counterexamples in Topology*. Dover, 1995.

[11] Robert H. Thomas. A majority consensus approach to concurrency control for multiple copy databases. *ACM Trans. Database Syst.*, 4(2):180–209, June 1979.

[12] C. Thomassen. *Kuratowski's Theorem*. Preprint series: Matematisk Institut. Matematisk Inst., Univ., 1980.

[13] C. Wu and G.G. Belford. The triangular lattice protocol: a highly fault tolerant and highly efficient protocol for replicated data. In *Proceedings of Reliable Distributed Systems in 11th Symposium*, pages 66–73, 1992.