COMMUNICATION COST OF MANAGING REPLICATED DATA
(Preiliminary Version)

by

A. Milp and O. Wolfson

Technical Report #476
November 1987
COMMUNICATION COST OF MANAGING REPLICAED DATA

Amir MILO and Ouri WOLFSON
Department of Computer Science, Technion - Israel Institute of Technology, Technion City, Haifa 32000, ISRAEL.

1. Introduction

1.1. Background

Management of replicated data is an important topic in distributed database and operating systems research. The two most popular management methods are read-one-write-all (rowa) (e.g. [1],[2]), and majority access (e.g. [2],[3],[4],[6]). In algorithms that use the first method, a logical read of a data item is done by reading one physical replica, and a logical write is done by writing all physical replicas. In algorithms that use the second method, a logical read is done by examining the time stamps of a majority of replicas, and physically reading the most recent one. A logical write is done by physically writing a majority of the physical replicas. The physical replicas transmitted over the communication network as a result of a logical read or write may be very large (whole files or relations), therefore minimizing transmission of such replicas is crucial for the performance of the communication network. If there are n replicas of a logical entity, a write will require n messages in the rowa case, and at least n/2 messages in the majority case. However, the message cost of a method does not accurately reflect the load on the communication network, since a message that has to traverse ten hops in order to reach its destination, and a message that does so in one hop are considered equal. Therefore we feel that the total number of hops traversed, or the communication cost, is a better measure of the load placed by a write. This measure was used in [5] to devise an efficient transaction commitment algorithm. In this note the problem considered is that of minimizing the communication cost of a logical write in the popular methods mentioned above (the read case is simple). We establish that the problem can be solved in quadratic time for the rowa method, whereas it is \textit{NP-Complete} for the majority method.

1.2. Definitions

The \textit{communication network} is an undirected graph $G=(V,E)$. $V$ is a set of processors and an edge in the
communication network between processors \( V_i \) and \( V_j \) represents a bidirectional link between them. We define the communication cost of a message from \( V_j \) to \( V_k \) to be the length of the shortest path in \( G \) between \( V_j \) and \( V_k \). Denote by \( R \subseteq V \) is the set of processors where some arbitrary fixed entity is replicated; the set \( R \) is called the residence set. Processor \( w \in V \) writes the entity, and is called the writer; The writer may or may not belong to the residence set. The processors \( R \cup \{ w \} \) are the participants in an algorithm, and are denoted by \( P \).

2. Read-one-write-all case

A rowa write instance is a directed graph \( I = (P, A) \). Each arc of \( A \) is called a message. Since the entity sent by the writer reaches every other participant we require that there is a path in \( I \) from \( w \) to each node \( V_i \) in \( R \). As we shall prove shortly, the only instances of interest are acyclic. A possible algorithm which sends exactly the messages of such a given instance, \( I \), is the following (assuming \( I \) is known to all participants): the writer sends the entity to its sons and each node after receiving the entity forwards it to its sons and so on until the entity reaches the leaves of \( I \). For the proof of theorem 1 we need to define the distance graph, \( D_G(V) \), for an arbitrary subset of processors \( V \subseteq V \). \( D_G(V) \) is a complete weighted graph with the set of vertices \( V \). The weight of an edge between \( V_j \) and \( V_k \) in \( D_G(V) \) is the length of the shortest path in \( G \) between \( V_j \) and \( V_k \). Denote by \( MST(D_G(V)) \) the minimum spanning tree of \( D_G(V) \).

Theorem 1: The necessary and sufficient communication cost of a rowa write instance is the total weight of \( MST(D_G(P)) \).

Proof: (necessary) By the definition of the rowa write instance the underlying graph of any rowa write instance is a connected subgraph of \( D_G(P) \) that spans all the nodes of \( P \). The minimal weight of such a graph is the weight of \( MST(D_G(P)) \).

(sufficient) Given an \( MST(D_G(P)) \) we can build the required rowa write instance whose communication cost is equal to the weight of \( MST(D_G(P)) \), by directing the edges of the \( MST(D_G(P)) \) to form a rooted tree, rooted at the writer. \( \square \)
3. Majority access case

In this case we assume the same communication network, communication cost, residence set, and writer. A majority write instance is a directed graph which satisfies the following conditions: 1) The writer is a node of the graph. 2) more than half of the nodes in \( P \) are in the graph. 3) there is a path from the writer to each node in the graph.

The majority write instance problem (MWI) is defined as follows:

**Input:** Graph \( G=(V,E) \), a set of nodes \( R \subseteq V \), a node \( w \in V \) and an integer \( K \).

**Question:** Is there a majority write instance whose total communication cost is no more than \( K \)?

**Theorem 2:** MWI is NP-Complete.

For the proof of theorem 2 we need some preliminary lemmas and definitions.

The majority tree problem is defined as follows:

**Input:** Graph \( G=(V,E) \) with weights on the edges, a node \( w \in V \) and an integer \( K \).

**Question:** Is there a subgraph of \( G \) which is a tree, such that the number of nodes in the tree is greater than \( \frac{|V|}{2} \), \( w \) is a node in the tree, and the sum of the weights of the edges is no more than \( K \)?

We denote the following subproblems: \( MT \) where \( G \) is any graph with non-negative weights, \( MT^+ \) where the weights are positive, and \( MT^c \) where \( G \) is complete graph and all the weights are positive. In all the problems the triangle inequality holds. We prove that \( MT, MT^+ \) and \( MT^c \) are in NP-Complete. To complete the proof of theorem 2 we also prove that MWI can be reduced to \( MT^c \).

**Lemma 1:** \( MT \) is NP-Complete.

**Proof:** \( MT \in NP \). Guess a subgraph of \( G \). Verify whether the subgraph is a tree, contains more than half of the nodes of \( R \) and \( w \) is a node of the subgraph and its total communication cost is no more than \( K \). Obviously, this verification can be done in polynomial time.

\( MT \) is NP-Hard. We transform the problem of 3 Exact Cover (3XC) to \( MT \). In 3XC the input consist of \( n \) sets over \( 3t \) elements; each set contains exactly 3 elements. The question is whether there are \( t \) sets that contain all the elements.

Given the sets and the elements from the 3XC problem, we construct the following graph (see fig. 1): each set is represented by a set-node \( s_1, s_2, \ldots, s_n \); each element is represented by a element-node
There are auxiliary nodes: $w$ and $a_1, a_2, \ldots, a_x$, where $x = 3t^4 + 2t - n$. To each element-node is connected a chain of $t^3 - 1$ nodes. The total number of nodes in the graph is $2(3t^4 + t) + 1$. There is an edge with weight 1 from $w$ to each set-node and to each aux-node. From every set-node to each one of its 3 members there is an edge with weight 1. Each element-node is connected with its chain by an edge with weight 0. The weight of the edges in the chain is also 0. Let $K = 4t$.

We will prove that there exists a solution to the 3XC problem if and only if there exists a solution to the MT problem for the constructed graph. (only if) Given a 3XC solution, the tree constituting the MT solution is defined as follows. Its nodes are: the element-nodes and the chain nodes (totaling $3t^4$ nodes) and the $t$ set-nodes that represent the sets in the solution to the 3XC problem. Also, $w$ is in the MT. Therefore in the tree there are more than half of the nodes in the graph. The edges in the tree are: from $w$ to the $t$ set-nodes (total weight 1), from these $t$ set-nodes to their element-nodes (total weight 3) and from the element-nodes to their chain nodes (weight 0). (if) First we prove that if there exists a solution to the MT problem then this solution contains all the element-nodes. Then we prove that if there exists a solution $T$, to the MT problem then there exists a solution $T^*$, such that: $T$ and $T^*$ have the same weight and the same set of nodes, and every element-node in $T^*$
is connected to a single set-node. The sets that are represented by the set-nodes in $T'$ are the solution to the 3XC problem.

**Proposition 1:** If there is a solution to the $MT$ problem with a total weight of no more than $K$, then it contains all the element nodes.

**Proof:** Assume the contrary, that some element-node is not in the tree. The chain nodes of this element-node are also not in the tree since they are connected only to this node. The other element-nodes and their chain nodes can provide at most $(3t-1)t^3$ nodes. We still have to find $t^3 + t$ nodes for the tree. All the other nodes that we can choose from are the set-nodes and the aux-nodes. For including each such node in the tree we have to use an edge of weight 1. The total weight would exceed $4t (= K)$; contradiction.

**Proposition 2:** If there exists a solution, $T$, to the $MT$ problem then there exists a solution, $T'$, with the same weight and the same set of nodes, such that every element-node in $T'$ is connected to a single set-node.

**Proof:** By examining fig. 1 it is obvious that in $T$ every element-node is connected to at least one set-node. We present an iterative procedure that transforms $T$ into $T'$. If element-node, $u$, is connected to more than one set-node, then one of them, $s_0$, is not connected to $w$ in $T$ (otherwise $T$ has a cycle). Omit the edge between $u$ and $s_0$ and add the edge between $w$ and $s_0$. We decreased the degree of the element-node without changing the weight of the tree or the set of nodes. We repeat this procedure until each element-node is connected with a single set-node.

In the above reduction we used a graph with zero edge weights. Next, we show that the problem remains NP-Complete even if we restrict the input graph to have positive weights.

**Lemma 2:** $MT^*$ is NP-Complete.

**Proof:** The transformation from $MT$ to $MT^*$ is as follows. We add one to the weight of each edge of $G$ and get a graph $G^*$ in which all the weights are positive. We add to $K$ the number of edges in the tree obtaining $K^* = K + \left\lfloor \frac{\mid V \mid + 1}{2} \right\rfloor - 1$.

Finally, we show that even if the input graph is complete the problem remains NP-Complete.

**Lemma 3:** $MT_c^*$ is NP-Complete.

**Proof:** The transformation from $MT^*$ to $MT_c^*$ is as follows. We add edges to $G^*$ to complete the graph.
The weight of each added edge from \( V_i \) to \( V_j \) is the shortest path between \( V_i \) and \( V_j \) in \( G^* \). Let \( K^*_c = K^* \).

\[ \square \]

Proof (of theorem 2): We transform \( MT_c^* \) to \( MWI \) as follows. Given a complete graph \( C = (V, E) \) and a node \( w \) we construct an instance of the \( MWI \) problem. The communication graph \( G \) consists of \( C \) with edges replaced by a path whose length is equal to the weight of the edge in \( C \). The nodes along the path, except the end points, are new. The set of participants is \( V \), and \( w, K \) are the same as in the \( MT_c^* \) problem.

\[ \square \]

REFERENCE


