Optimal Placement of Replicas in Trees with Read, Write, and Storage Costs

Konstantinos Kalpakis and Koustuv Dasgupta and Ouri Wolfson

Abstract— **We consider the problem of placing copies of objects in a tree network in order to minimize the cost of servicing read and write requests to objects when the tree nodes have limited storage and the number of copies permitted is limited. The set of nodes that have a copy of the object is the residence set of the object. A node wishing to read the object will read the object from the closest node in the residence set. A node wishing to update the object will update the copy of the object at all the nodes in the residence set; updates are propagated over a certain minimum spanning tree. The cost associated with a residence set equals the cost of servicing all the read and write requests, and the storage costs for those copies.**

We describe a $O(n^3p^2)$–time algorithm for finding an optimal residence set of size p for an object in a tree with n nodes, taking into consideration the read, write, and storage costs. Furthermore, we describe a $O(n^3p^2\Lambda_{\max}^2)$–time algorithm for finding a minimum cost normal p–residence set for an object in a tree, this time also taking into account the load imposed by the nodes of the tree on the nodes in a residence set and their capacity constraints, where $\Lambda_{\max}$ is an upper bound on the capacity of each node of the tree.

Keywords—**data replication, multicasting, facility location, p–medians, file allocation, tree networks.**

I. Introduction

The recent growth in the World Wide Web is moving us to distributed highly interconnected information systems. In such systems, an object (a web page, an image, a video clip, a file, etc) is read and written from multiple locations on the Internet. Maintaining multiple copies of an object at various locations is an approach for improving system performance (time to read or write an object).

In this paper, we consider the problem of placing copies of an object at multiple locations in a distributed system, whose interconnection network is a tree, in order to minimize the total cost of servicing the read/write requests. Consider an object and a tree that connects a set of nodes V. Each node of the tree issues a number of read requests and a number of write requests to the object. Let S be a subset of the nodes of the tree at which there exist copies (replicas) of the object. The subset of nodes S is called a *residence set*. Each node of the tree sends its read and write requests to the closest node that has a copy of the object, and each write to a copy of the object must be propagated to all other copies of the object. The cost of propagating a write to all copies of the object is taken to be equal to the cost of a minimum spanning tree of a certain graph; this graph is the subgraph of the complete graph that corresponds to the tree network restricted to the the set of nodes that have a copy of the object. The model we use in this paper is based on the model of Wolfson and Milo [1]. Further, we assume that having a copy of the object at a node has an associated storage cost that depends on the object and that node. The cost of a residence set is equal to the sum of its total read, write, and storage costs. We are interested in finding a residence set of minimum cost. We call this problem the *optimal residence set* problem for trees with read, write, and storage costs.

The optimal residence set (file allocation) problem has been studied extensively in the literature. Dowdy and Foster [2] survey a number of mixed linear programming models for the file allocation problem. Wolfson and Milo [1] consider the optimal residence set problem without storage costs for various interconnection networks (completely–connected, tree, and ring networks). They show that the optimal residence set problem without storage costs is NP–hard for general topologies, and they provide efficient algorithms for finding optimal residence sets: an $O(n)$–time algorithm for tree networks, an $O(n^5)$–time algorithm for ring networks, and an $O(1)$–time algorithm for a completely connected network with edges of unit length, where n is the number of nodes in the network. Our model is different from that of Wolfson and Milo [1] since we also consider storage costs. Fisher and Hochbaum [3] consider the problem of database location in computer networks (a problem similar to the one of this paper), and describe computational experiments based on mixed integer programming models. Our model is different from that of [3], since they assume what Wolfson and Milo [1] call a "naive–write policy" (i.e. the write cost is equal to the sum of the distances between the writing node and each one of the nodes with a copy of the object).

When there are no write–costs, our problem reduces to the (uncapacitated) facility location problem which also has been studied extensively due to its applications (see [4], [5], [6]). When there are only read costs, our problem reduces to the p–median problem [7], [5], [8], with the restriction that no more than p copies of the object are made. Hochbaum [5] describes approximation algorithms for find-

K. Kalpakis is with the Computer Science & Electrical Engineering Department, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250. Email: kalpakis@csee.umbc.edu. Phone: 410-455-3143. Fax: 410-455-3969. He is supported in part by National Science Foundation under grant number IRI–9729495, and by NASA under Cooperative Agreement NCC5–315. Corresponding author.

K. Dasgupta is with the Computer Science & Electrical Engineering Department, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250. Email: dasgupta@csee.umbc.edu. Phone: 410-455-6368. Fax: 410-455-3969.

O. Wolfson is with the Department of Electrical Engineering and Computer Science, University of Illinois at Chicago, IL 60607. Email: wolfson@ouri.eecs.uic.edu. Phone: 312-996-6770. He is supported in part by National Science Foundation under grant numbers IRI-9712967, CCR-9816633, and CCR-9803974, DARPA grant N66001-97-2-8901, and Army Research Labs grant DAAL01-96-2-0003.

ing p–medians on the plane (discrete and continuous) as well as on networks. Arora et al [7] describe randomized polynomial approximation schemes for finding p–medians in Euclidean spaces. Kariv and Hakimi [8] show that the p–median problem is NP–hard for general networks, and they provide an $O(n^2p^2)$–time algorithm for finding a p–median for a tree with n nodes.

We describe an $O(n^3p^2)$–time dynamic programming algorithm for finding an optimal residence set of size p for an object on a tree with n nodes, taking into consideration the read, write, and storage costs. Our algorithm can find an optimal residence set for a tree with n nodes in $O(n^5)$ time when read, write, and storage costs are taken into consideration. Our algorithm is a non–trivial extension of the p–median algorithm for trees by Kariv and Hakimi [8].

We extend our algorithm to also consider load and capacity constraints on the nodes. We assume that each node that is serviced by a node of a residence set imposes a certain integer load on that node, and that the total load at each node of a residence set should not exceed a certain integer upper bound for that node (e.g. the node's capacity). The *capacitated optimal residence set* problem for trees is the problem of finding a minimum read–write–storage cost residence set that respects the capacities of the nodes. This problem, when there are no write costs, reduces to the capacitated facility location problem [6]. We prove that the optimal capacitated residence set problem for trees is NP–complete, and we describe an $O(n^3p^2\Lambda^2_{\max})$–time algorithm for finding an optimal capacitated normal p–residence set for a tree with n nodes, where $\Lambda_{\max}$ is the maximum capacity of each node of the tree. A *normal residence set* is a residence set for which each node v of the tree has all its requests serviced by either a node in the subtree of T rooted at v or by a node which also services all the requests of v's parent.

The rest of the paper is organized as follows. In section 2 we provide the necessary preliminaries and notations. In section 3, we present our algorithm for finding optimal residence sets. In section 4, we present an algorithm for finding normal residence sets with minimum total read–write–storage cost, this time taking into consideration the load imposed upon the nodes in a residence set by the nodes of the tree and the capacity constraints of the nodes. Finally, in section 5, we conclude the paper.

II. Preliminaries

Consider a rooted tree T with a set V of n vertices. Assume, w.l.o.g., that the children of each vertex are ordered left–to–right. Let $\mathrm{root}(T)$ be the root of T. For each vertex $v \in T$, let $\deg(v)$ be the number of children of v, let T_v be the subtree of T rooted at v, and let $T_v^{(i)}$ be the subtree of T that consists of v and all the subtrees rooted at each of the i leftmost children of v.

Each edge of of the tree T has an associated non-negative length (cost). The distance $\delta(u, v)$ between any two vertices u and v of T is equal to the length of a shortest path from u to v in T. The distance of a vertex $v \in T$ from any subset $S \subseteq V$ is $\delta(v, S) = \min\{\delta(v, u) : u \in S\}$, and a vertex $u \in S$ *covers* v if $\delta(v, u) = \delta(v, S)$.

The distance graph $\hat{D}_T = (V, V \times V)$ is an edge–weighted complete undirected graph, such that the weight of each edge (u, v) of $\hat{D}_T$ is equal to the distance $\delta(u, v)$ between u and v in T. For any subset $S \subseteq V$, let $\mathrm{mst}(S)$ be the cost of a minimum spanning tree (MST) of the subgraph of $\hat{D}_T$ induced by S.

Suppose that the vertices of T issue read and/or write requests for an object, and that copies of that object can be stored at multiple vertices of T. Let r, w, and s be three integer–valued functions representing the number of read requests, the number of write requests, and the storage cost for each vertex of T respectively. The set of vertices at which copies of that object are placed is called a *residence set*. Let $S \subseteq V$ be any residence set. We call S a k–*residence set* if $|S| = k$. Every residence set S induces a function $\sigma : V \to S$ that assigns to each vertex $v \in V$ one vertex $u \in S$ such that u covers v (breaking ties arbitrarily). We call σ a *replica assignment* induced by S.

Consider a residence set S for T. Each node of T sends its read and write requests to a node in S that covers it. Whenever a node in S receives a write request, it propagates that write request to all other nodes in S using an MST of the subgraph of $\hat{D}_T$ induced by S.

The read cost of S is the cost of servicing the read requests issued from all the vertices of T and is given by

$$\sum_{v \in V} r(v) \cdot \delta(v, S). \tag{1}$$

The write cost of S is the cost of servicing all the write requests issued from all the vertices of T and is given by

$$\sum_{v \in V} w(v) \cdot (\delta(v, S) + \mathrm{mst}(S)). \tag{2}$$

The storage cost of S is the cost of placing copies of that object at all the vertices in S and is given by

$$\sum_{v \in S} s(v). \tag{3}$$

The total cost for a residence set S for T is

$$\begin{aligned}\mathrm{Cost}(S, V) \;=\; & \sum_{v \in V} r(v) \cdot \delta(v, S) + \\ & \sum_{v \in V} w(v) \cdot (\delta(v, S) + \mathrm{mst}(S)) + \\ & \sum_{v \in S} s(v). \end{aligned} \tag{4}$$

Given any integer $1 \le k \le n$, we are interested in finding a k–residence set S for T of minimum cost. We call such an S an *optimal k–residence set* for T. An *optimal residence set* for T is an optimal k–residence set for T for some integer $1 \le k \le n$.

Let $\alpha(v) = r(v) + w(v)$ denote the total number of requests issued by a vertex $v \in V$, and let $W_{\mathrm{total}} = \sum_{v \in V} w(v)$ be the total number of write requests. Let

V' be any subset of V. We define the *modified total cost* $\mathrm{C}(S,V')$ for a residence set $S \subseteq V'$ with respect to V' to be

$$\mathrm{C}(S,V') = \sum_{v\in V'} \alpha(v)\cdot\delta(v,S) + \sum_{v\in S} s(v) + W_{\text{total}}\cdot \mathrm{mst}(S). \quad (5)$$

The modified total cost for S with respect to V' is the cost of servicing all the read and write requests from all the vertices in V', the storage cost for S, and the cost of propagating an additional $W_{\text{total}} - \sum_{v\in V'} w(v)$ write requests to all the nodes in S. Note that $\mathrm{C}(S,V) = \mathrm{Cost}(S,V)$, and that $\mathrm{C}(S,V)$ accounts for the cost of propagating the total number of writes W_{total} by all the nodes of T to all the nodes of S. We call a k–residence set for V' of minimum modified total cost an *optimal modified k–residence set* for V'.

Consider a residence set $S \subseteq V$ for T and a node v of T. Node v can be covered either by a vertex that is in the subtree of T rooted at v or by a vertex that can also cover the parent of v in T. We say that a replica assignment σ induced by S is a *normal replica assignment* if for all nodes $v \in V$ either $\sigma(v)$ is in the subtree T_v or $\sigma(v) = \sigma(u)$, where u is the parent of v in T. A *normal residence set* is a residence set that induces a normal replica assignment. Clearly, every optimal k–residence set for T is an optimal normal k–residence set for T. Therefore, in order to find an optimal k–residence set for T it is sufficient to only consider normal residence sets.

We define $R(u,k,v,i)$ as

$$R(u,k,v,i) = \min\Big\{\mathrm{C}(S, T_v^{(i)}\cup\{u\}) \;:\; S\subseteq T_v^{(i)}\cup\{u\}, |S| = k, \text{ and } u \text{ covers } v\Big\}. \quad (6)$$

i.e. the modified total cost of a minimum–cost k-residence set $S \subseteq T_v^{(i)} \cup \{u\}$ for the subtree $T_v^{(i)}$ rooted at the vertex v, such that u covers v. We call S an *optimal modified u–restricted k–residence set* for $T_v^{(i)}$. Observe that, since there always exists an optimal k–residence set for T that is normal, it is sufficient to consider only normal residence sets for in minimization in equation (6). The cost of an optimal modified k–residence set for T_v is given by

$$\min\Big\{R(u,k,v,\deg(v)) \;:\; u\in T_v\Big\}. \quad (7)$$

Moreover, given an efficient method for computing $R(u,k,v,i)$, we can find an optimal k–residence set for T using the equation above with v equal to the root of T. Note that the restriction of a replica assignment of a (normal) optimal modified u–restricted k–residence set S for a subtree $T_v^{(i)}$ induces, in a natural way, a (normal) residence set S' for the subtree $T_v^{(i-1)}$ of $T_v^{(i)}$: S' consists of those nodes that cover nodes of $T_v^{(i-1)}$. In the following section, we will show that S' is also an optimal modified u'–restricted k'–residence set for $T_v^{(i-1)}$, for some u' and k'. We do that by showing certain relationships between the cost of MSTs for S and S', since the number of writes that need to be propagated to all the nodes in S and S' is the same. This in turn, provides us with an efficient recursive algorithm for computing $R(u,k,v,i)$.

III. Finding Optimal Resident Sets

We describe a dynamic programming algorithm for finding u–restricted k–residence sets with minimum total modified cost, for all $u \in T$ and $1 \le k \le n$.
First, we show a recurrence for computing $R(u,k,v,i)$ in Theorem 1, using Lemmas 1–4 which we provide immediately after. Second, we analyze a dynamic programming algorithm for computing that recurrence, and therefore finding an optimal k–residence set.

Theorem 1: Let v be any vertex of T, let v_i be the ith child of v, $0 \le i \le \deg(v)$, and let u be any vertex of T. The cost of an optimal modified u–restricted k–resident set for $T_v^{(i)}$ is given by

$$R(u,k,v,i) = \begin{cases} X_0, & \text{if } k=1 \text{ and } i \ge 0 \\ \min\{X_1, X_2\}, & \text{if } u \notin T_{v_i} \\ X_1, & \text{if } u \in T_{v_i} \\ \text{undefined}, & \text{otherwise.} \end{cases} \quad (8)$$

where

$$X_0 = \sum_{v'\in T_v^{(i)}} \alpha(v')\cdot\delta(v',u) + s(u), \quad (9)$$

$$X_1 = \min_{k_1}\Big\{R(u,k_1,v,i-1) + R(u,k-k_1+1,v_i,\deg(v_i)) - s(u)\Big\}, \quad (10)$$

$$X_2 = \min_{k_2,u'}\Big\{R(u,k_2,v,i-1) + R(u',k-k_2,v_i,\deg(v_i)) + W_{\text{total}}\cdot\delta(u,u')\Big\}, \quad (11)$$

and where $1 \le k_1, k_2 \le k$, and where $u' \in T_{v_i}$ such that $\delta(v_i,u') \le \delta(v_i,u)$ and $\delta(v,u') \ge \delta(v,u)$. Furthermore, the cost of an optimal modified k–resident set for T_v is given by

$$\min\Big\{R(u,k,v,\deg(v)) \;:\; u\in T_v\Big\}, \quad (12)$$

the cost of an optimal k–resident set for T is given by

$$\min\Big\{R(u,k,\mathrm{root}(T),\deg(\mathrm{root}(T))) \;:\; u\in T\Big\}, \quad (13)$$

and the cost of an optimal resident set for T is given by

$$\min_{1\le k\le |T|}\Big\{R(u,k,\mathrm{root}(T),\deg(\mathrm{root}(T))) \;:\; u\in T\Big\}. \quad (14)$$

Proof: The proof of this theorem uses Lemmas 3, and 4 which are shown immediately after this theorem.

First, we show the initial conditions for $R(u,k,v,i)$. By definition of the total modified cost for a residence set for $T_v^{(i)}$ we have that

$$R(u,1,v,i) = \sum_{v' \in T_v^{(i)}} \alpha(v') \cdot \delta(v',u) + s(u), \quad (15)$$

for $0 \le i \le \deg(v)$. Furthermore, again by definition, $R(u,k,v,i)$ is undefined when either $i=0$ and $k \neq 1$, or $k \le 0$, or $k > |T_v^{(i)} \cup \{u\}|$.
Second, we show how to compute $R(u,k,v,i)$ recursively for $k>1$. There are two cases to consider depending on whether u is a descendant of v_i or not.

Case 1: u is not a descendant of v_i. If u covers v_i then, from Lemma 3, we have that

$$\begin{aligned} R(u,k,v,i) &= R(u,k_1,v,i-1) + \\ &\quad R(u,k-k_1+1,v_i,\deg(v_i)) - \\ &\quad s(u), \end{aligned} \quad (16)$$

for some integer $1 \le k_1 \le k$. Hence, if u covers v_i then $R(u,k,v,i) = X_1$. If u does not cover v_i, then, from Lemma 4, we have that

$$\begin{aligned} R(u,k,v,i) &= R(u,k_2,v,i-1) + \\ &\quad \min_{u'} \Big\{ R(u',k-k_2,v_i,\deg(v_i)) + \\ &\quad W_{\text{total}} \cdot \delta(u,u') \Big\}, \end{aligned} \quad (17)$$

where u' is a vertex in T_{v_i} such that $\delta(v,u) \le \delta(v,u')$ and $\delta(v_i,u') < \delta(v_i,u)$, and k_2 is a positive integer $< k$. Hence, if u does not cover v_i then $R(u,k,v,i) = X_2$. Therefore, $R(u,k,v,i) = \min\{X_1, X_2\}$.
Case 2: u is a descendant of v_i. In this case, since u covers v and since v_i is a child of v, it follows that u covers v_i as well. Hence, from Lemma 3 we have that

$$\begin{aligned} R(u,k,v,i) &= R(u,k_1,v,i-1) + \\ &\quad R(u,k-k_1+1,v_i,\deg(v_i)) - \\ &\quad s(u), \end{aligned} \quad (18)$$

for some integer $1 \le k_1 \le k$. Therefore, in this case, $R(u,k,v,i) = X_1$.
Third, by definition, it follows that: (a) the cost of an optimal modified k–resident set for T_v is given by (12); (b) the cost of an optimal k–resident set for T is given by (13); and (c) the cost of an optimal resident set for T is given by (14). ■
Next, we prove Lemmas 3 and 4 together with Lemmas 1 and 2 which are used in the proofs of the former two lemmas. These four lemmas characterize the optimal substructure of the problem of finding u–restricted k–resident sets of minimum total modified cost.

Lemma 1: Let v_1 be any vertex of T, let v_2 be a child of v_1, and let u be any vertex of T. Let T_j be any subtree of T rooted at v_j, $j=1,2$, such that $T_1 \cap T_2 = \emptyset$. Let S_j be any u–restricted k_j–residence set for T_j, $j=1,2$. Then, $S_1 \cup S_2$ is a u–restricted (k_1+k_2)–residence set for $T_1 \cup T_2$, and

$$\begin{aligned} \mathrm{C}(S_1 \cup S_2, T_1 \cup T_2 \cup \{u\}) &= \mathrm{C}(S_1, T_1 \cup \{u\}) + \\ &\quad \mathrm{C}(S_2, T_2 \cup \{u\}) - \\ &\quad s(u). \end{aligned} \quad (19)$$

Proof: It is sufficient to show that $\mathrm{mst}(S_1 \cup S_2) =$

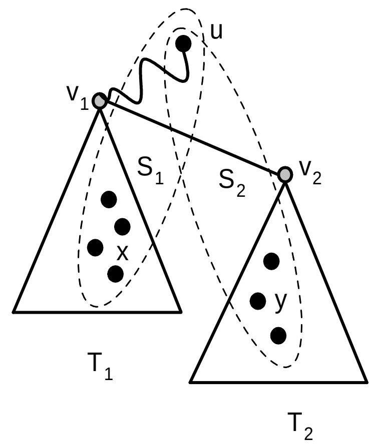

Fig. 1. Subtrees used in the proof of Lemma 1. Black nodes are nodes in the residence set.

$\mathrm{mst}(S_1) + \mathrm{mst}(S_2)$. Note that $S_1 \cap S_2 = \{u\}$. Consider an MST for $S_1 - \{u\}$ and an MST for $S_2 - \{u\}$. Since u covers both v_1 and v_2, $\delta(x,v_1) \ge \delta(u,v_1)$ for all $x \in S_1$, and $\delta(y,v_2) \ge \delta(u,v_2)$ for all $y \in S_2$ (see Figure 1). Furthermore, for any two vertices $x \in S_1 - \{u\}$ and $y \in S_2 - \{u\}$, we have that

$$\delta(x,y) = \delta(x,v_1) + \delta(v_1,v_2) + \delta(v_2,y), \quad (20)$$

since the edge (v_1,v_2) is on the single path from x to y in T. Then, since $\delta(x,v_1) \ge \delta(u,v_1)$, it follows that $\delta(x,y) \ge \delta(u,v_1) + \delta(v_1,v_2) + \delta(v_2,y) \ge \delta(u,y)$. Similarly, since $\delta(v_2,y) \ge \delta(v_2,u)$, we have that $\delta(x,y) \ge \delta(x,v_1) + \delta(v_1,v_2) + \delta(v_2,u) \ge \delta(x,u)$. Therefore, (see Theorem 24.1 in [9])

$$\begin{aligned} \mathrm{mst}(S_1 \cup S_2) &= \mathrm{mst}(S_1 - \{u\}) + \delta(u, S_1 - \{u\}) + \\ &\quad \mathrm{mst}(S_2 - \{u\}) + \delta(u, S_2 - \{u\}), \end{aligned} \quad (21)$$

which implies

$$\mathrm{mst}(S_1 \cup S_2) = \mathrm{mst}(S_1) + \mathrm{mst}(S_2). \quad (22)$$

■

Lemma 2: Let v_1 be any vertex of T, let v_2 be any child of v_1, and let u be any vertex of T. Let T_j be any subtree of T rooted at v_j, $j=1,2$, such that $T_1 \cap T_2 = \emptyset$. Let S_1 be a u–restricted k_1–residence set for T_1, and S_2 be a k_2–residence set for T_2 such that $S_1 \cap S_2 = \emptyset$, $\delta(u,v_1) \le \delta(v_1,S_2)$, and $\delta(u,v_2) > \delta(v_2,S_2)$. Then $S_1 \cup S_2$ is a u–restricted (k_1+k_2)–residence set for $T_1 \cup T_2$, and

$$\begin{aligned} \mathrm{C}(S_1 \cup S_2, T_1 \cup T_2 \cup \{u\}) &= \mathrm{C}(S_1, T_1 \cup \{u\}) + \\ &\quad \mathrm{C}(S_2, T_2) + \\ &\quad W_{\text{total}} \cdot \delta(u, S_2). \end{aligned} \quad (23)$$

Proof: Let $S = S_1 \cup S_2$. It is sufficient to show that

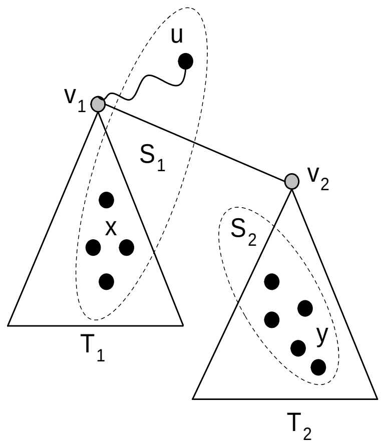

Fig. 2. Subtrees used in the proof of Lemma 2. Black nodes are nodes in the residence set.

$\mathrm{mst}(S) = \mathrm{mst}(S_1) + \mathrm{mst}(S_2) + \delta(u, S_2)$. Consider an MST for S_1 and an MST for S_2. Observe that $\delta(v_1, S) \geq \delta(v_1, u)$ and $\delta(v_2, u) > \delta(v_2, S)$, since u covers v_1 but it does not cover v_2 (see Figure 2). Furthermore, for any two vertices $x \in S_1 - \{u\}$ and $y \in S_2$, we have that

$$\delta(x, y) = \delta(x, v_1) + \delta(v_1, v_2) + \delta(v_2, y), \tag{24}$$

since the edge (v_1, v_2) is on the single path from x to y. Then, since $\delta(x, v_1) \geq \delta(u, v_1)$, $\delta(x, y) \geq \delta(u, v_1) + \delta(v_1, v_2) + \delta(v_2, y) \geq \delta(u, y)$. Therefore, it follows that (see Theorem 24.1 in [9])

$$\mathrm{mst}(S_1 \cup S_2) = \mathrm{mst}(S_1) + \mathrm{mst}(S_2) + \delta(u, S_2). \tag{25}$$

■

Lemma 3: Let S be an optimal modified u–restricted k–residence set for $T_v^{(i)}$, such that u covers both v and the ith child v_i of v, $1 \leq k \leq |T_v^{(i)} \cup \{u\}|$, $1 \leq i \leq \deg(v)$. Let $S_1 = (T_v^{(i-1)} \cap S) \cup \{u\}$ and $S_2 = (S - S_1) \cup \{u\}$. Then, S_1 is an optimal modified u–restricted k_1–residence set for $T_v^{(i-1)}$ and S_2 is an optimal modified u–restricted $(k - k_1 + 1)$–residence set for T_{v_i}, where $k_1 = |S_1| \geq 1$. Furthermore,

$$\begin{aligned} R(u, k, v, i) \;\; = \;\; & R(u, k_1, v, i-1) + \\ & R(u, k - k_1 + 1, v_i, \deg(v_i)) - \\ & s(u). \end{aligned} \tag{26}$$

Proof: Since u covers both v and v_i, and since every path from any vertex in $y \in S_2 - \{u\}$ to any vertex $x \in T_v^{(i-1)}$ goes through v, it follows that $\delta(y, x) \geq \delta(u, x)$. Hence, all the vertices in $T_v^{(i-1)}$ are covered by vertices in S_1. Similarly, since u covers both v and v_i, and since every path from any vertex in $x \in S_1 - \{u\}$ to any vertex $y \in T_{v_i}$ goes through v_i, it follows that $\delta(x, y) \geq \delta(u, y)$. Hence, all the vertices in T_{v_i} are covered by vertices in S_2.

Next, we show that S_1 is an optimal modified u–restricted k_1–residence set for $T_v^{(i-1)}$. Suppose for the sake of contradiction that S_1 is not an optimal modified u–restricted k_1–residence set for $T_v^{(i-1)}$. Let S_1' be a u–restricted k_1–residence set for $T_v^{(i-1)}$ such that $\mathrm{C}(S_1', T_v^{(i-1)} \cup \{u\}) < \mathrm{C}(S_1, T_v^{(i-1)} \cup \{u\})$. Consider the set $S' = S_1' \cup S_2$. Observe that $S_1' \cup S_2$ is a u–restricted k–residence set for $T_v^{(i)}$. From Lemma 1 we have that

$$\begin{aligned} \mathrm{C}(S_1' \cup S_2, T_v^{(i)} \cup \{u\}) \;\; = \;\; & \mathrm{C}(S_1', T_v^{(i-1)} \cup \{u\}) + \\ & \mathrm{C}(S_2, T_{v_i} \cup \{u\}) - \\ & s(u), \end{aligned} \tag{27}$$

and that

$$\begin{aligned} \mathrm{C}(S_1 \cup S_2, T_v^{(i)} \cup \{u\}) \;\; = \;\; & \mathrm{C}(S_1, T_v^{(i-1)} \cup \{u\}) + \\ & \mathrm{C}(S_2, T_{v_i} \cup \{u\}) - \\ & s(u). \end{aligned} \tag{28}$$

Hence, the total modified cost for $S_1' \cup S_2$ is less than that of $S_1 \cup S_2 = S$, which contradicts the optimality of S. The proof that S_2 is an optimal modified u–restricted $(k - k_1 + 1)$–residence set for T_{v_i} is similar.

Finally, from Lemma 1, it follows that

$$\begin{aligned} R(u, k, v, i) \;\; = \;\; & R(u, k_1, v, i-1) + \\ & R(u, k - k_1 + 1, v_i, \deg(v_i)) - \\ & s(u). \end{aligned} \tag{29}$$

■

Lemma 4: Let S be an optimal modified u–restricted k–residence set for $T_v^{(i)}$, such that u covers v but it does not cover the ith child v_i of v, $1 \leq k \leq |T_v^{(i)} \cup \{u\}|$, $1 \leq i \leq \deg(v)$. Let $S_1 = (T_v^{(i-1)} \cap S) \cup \{u\}$ and $S_2 = (S - S_1)$. Then, S_1 is an optimal modified u–restricted k_2–residence set for $T_v^{(i-1)}$ and S_2 is an optimal modified u–restricted $(k - k_2)$–residence set for T_{v_i}, where $k_2 = |S_1| \geq 1$. Furthermore,

$$\begin{aligned} R(u, k, v, i) \;\; = \;\; & R(u, k_2, v, i-1) + \\ & \min_{u'} \Big\{ R(u', k - k_2, v_i, \deg(v_i)) + \\ & W_{\mathrm{total}} \cdot \delta(u, u') \Big\}, \end{aligned} \tag{30}$$

where $u' \in S_2$ is a vertex that covers v_i. Moreover, $\delta(v, u) \leq \delta(v, u')$ and $\delta(v_i, u') < \delta(v_i, u)$.

Proof: Since u covers v but not v_i, and since every path from any vertex $y \in S_2$ to any vertex $x \in T_v^{(i-1)}$ goes through v_i, it follows that $\delta(y, x) \geq \delta(u, x)$ (see Figure 3). Hence, all vertices in $T_v^{(i-1)}$ are covered by vertices in S_1. Similarly, since u covers v but it does not cover v_i, and since every path from any vertex $x \in S_1 - \{u\}$ to any vertex $y \in T_{v_i}$ goes through v, it follows that $\delta(x, y) \geq \delta(u, y) \geq \delta(u', y)$. Hence, all vertices in T_{v_i} are covered by vertices in S_2.

We show, by contradiction, that S_1 is an optimal modified u–restricted k_2–residence set for $T_v^{(i-1)}$. Let S_1' be a

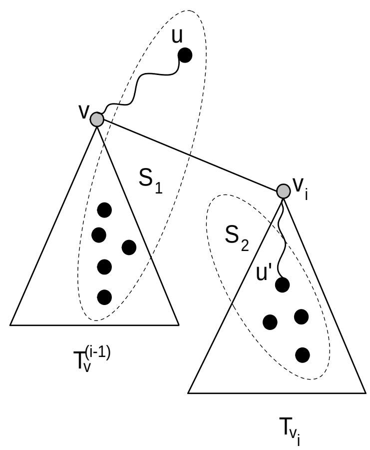

Fig. 3. Subtrees used in the proof of Lemma 4. Black nodes are nodes in the residence set.

u–restricted k_2–residence set for $T_v^{(i-1)}$ with smaller total modified cost than S_1. Then, from Lemma 2 it follows that $S_1' \cup S_2$ is a u–restricted k–residence set for $T_v^{(i)}$ whose total modified cost if smaller than that of S, which contradicts the optimality of S.

Consider now the set S_2. Recall that $u' \in S_2$ covers v_i. Clearly, $\delta(v_i, u') < \delta(v_i, u)$ and $\delta(v, u') \geq \delta(v, u)$ (see Figure 3). Then,

$$\begin{aligned} \mathrm{C}(S_1 \cup S_2, T_v^{(i)} \cup \{u\}) \;=\; & \sum_{v' \in T_v^{(i-1)}} \alpha(v') \cdot \delta(v', S_1) + \\ & \sum_{v' \in T_{v_i}} \alpha(v') \cdot \delta(v', S_2) + \\ & \sum_{v' \in S_1 \cup S_2} s(v') + \\ & W_{\mathrm{total}} \cdot \mathrm{mst}(S_1 \cup S_2). \end{aligned} \tag{31}$$

Since $\mathrm{mst}(S_1 \cup S_2) = \mathrm{mst}(S_1) + \mathrm{mst}(S_2) + \delta(u, u')$ (see proof of Lemma 2),

$$\begin{aligned} \mathrm{C}(S_1 \cup S_2, T_v^{(i)} \cup \{u\}) \;=\; & \sum_{v' \in T_v^{(i-1)}} \alpha(v') \cdot \delta(v', S_1) + \\ & \sum_{v' \in T_{v_i}} \alpha(v') \cdot \delta(v', S_2) + \\ & \sum_{v' \in S_1 \cup S_2} s(v') + \\ & W_{\mathrm{total}} \cdot \mathrm{mst}(S_1) + \\ & W_{\mathrm{total}} \cdot \mathrm{mst}(S_2) + \\ & W_{\mathrm{total}} \cdot \delta(u, u'), \end{aligned} \tag{32}$$

which implies that

$$\begin{aligned} \mathrm{C}(S_1 \cup S_2, T_v^{(i)} \cup \{u\}) \;=\; & \mathrm{C}(S_1, T_v^{(i-1)} \cup \{u\}) + \\ & \mathrm{C}(S_2, T_{v_i}) + \\ & W_{\mathrm{total}} \cdot \delta(u, u'). \end{aligned} \tag{33}$$

where u covers v but it does not cover v_i, and u' covers v_i. Therefore, S_2 must be an optimal modified u'–restricted $(k - k_2)$–resident set for T_{v_i}, since otherwise S will not be an optimal modified u–restricted k–residence set for $T_v^{(i)}$. Finally, from Lemma 2 we have that

$$\begin{aligned} R(u, k, v, i) \;=\; & R(u, k_2, v, i-1) + \\ & \min_{u'} \Big\{ R(u', k - k_2, v_i, \deg(v_i)) + \\ & W_{\mathrm{total}} \cdot \delta(u, u') \Big\}. \end{aligned} \tag{34}$$

■

Finally, we analyze the complexity of a straightforward dynamic programming algorithm for computing optimal residence sets for a tree with storage costs in the following theorem.

Theorem 2: Let T be a tree with n vertices and p be an integer $1 \leq p \leq n$. We can find an optimal p–resident set for T in $O(n^3p^2)$ time, and an optimal resident set for T in $O(n^5)$ time.

Proof: The recursive equation (8) in Theorem 1 can be evaluated by a straightforward dynamic programming algorithm, using either a memoization or a bottom–up evaluation approach. The running–time of such a dynamic programming algorithm for computing an optimal modified p–resident set for subtrees $T_v^{(i)}, v \in T, 0 \leq i \leq \deg(v)$ via equation (8) is $O(n^3p^2)$. This is because $R(u, k, v, i)$ is defined for at most n^2p entries, and each entry requires $O(np)$ time for its evaluation. Hence, we can compute an optimal p–resident set for T in $O(n^3p^2)$ time, and an optimal resident set for T in $O(n^5)$ time. ■

We show a sample tree network and optimal residence sets with and without storage costs in Appendix A.

IV. Considering Node Capacities and Load

Consider a residence set S for a tree with a set of vertices V. Each vertex $v \in V$ has all its requests serviced by one node $u \in S$ which covers v. Suppose that vertex v imposes an integer load $\lambda(v) \geq 0$ on the vertex u. Further, suppose that each vertex $u \in V$ has an integer capacity $\Lambda(u) \geq 0$, and that the total load imposed on u by all the vertices assigned to u should be $\leq \Lambda(u)$. We are interested in finding a minimum cost residence set for a tree that takes into account the read, write, and storage costs and that in addition respects the capacity constraints of the nodes. We call this problem the *capacitated optimal residence set* problem for trees. Note that, when there are no write costs, this problem reduces to the capacitated facility location problem [6].

Taking into account capacity constraints is an important consideration. The average number of requests serviced by each node u of a residence set directly affects the average response that the nodes covered by u observe (due to queuing delays and network congestion). Further, in certain types of applications, such as image, video, and map servers, it is necessary to place with each copy of an object a copy of an appropriate software system, for example a DBMS or a GIS system, for servicing read and write requests. In many cases however, such systems imposes restrictions on the number of concurrent users. This kind of situation can be easily modeled by using loads and capacity constraints for the nodes.

In this section we show how to extend $R(u,k,v,i)$ in order to find minimum–cost normal residence sets that respect the capacity constraints of the nodes. When capacity constraints and loads are present, it is not necessarily true that every optimal k–residence set induces a normal replica assignment, i.e. a replica assignment in which every vertex $v \in T$ is assigned either a vertex in the subtree of T rooted at v or a vertex that is also assigned to the parent of v. This is due to the fact that, when capacity constraints are present and there is a tie between nodes of the residence set that can cover a node v of the tree, the choice of the node u that covers (is assigned to) v does affect the total load imposed on u.

Theorem 3: The capacitated optimal residence set problem for trees is NP–complete.

Proof: Clearly, the capacitated optimal residence problem for trees is in NP. We show that the problem is NP–hard by a straightforward reduction from the bin–packing problem which is known to be NP–hard [10]. The bin–packing problem is defined as follows: given n items with integer sizes $x_i \geq 0$, $i = 1, \ldots, n$ and an unlimited supply of bins each with integer size $M \geq 0$, pack all the items into the smallest number of bins. Given a instance of the bin-packing problem, we construct an instance of the capacitated optimal residence set for trees as follows: take any rooted tree with n nodes and edges of length 0, and assume that each node i, $i = 1, \ldots, n$ issues 1 read and 1 write request, has storage cost 1, imposes load x_i on a node that covers (is assigned to) it, and has capacity M. Then, it is easy to see that there exists a solution to the above instance of the bin–packing problem that uses k bins if and only if there exists a solution to the above instance of the capacitated optimal residence set problem for trees whose cost is equal to k. Thus, the capacitated optimal residence set problem is NP–complete. ∎

Next, we extend $R(u,k,v,i)$ in order to find minimum–cost normal residence sets that respect the capacity constraints of the nodes of the tree. The idea is to further constraint $R(u,k,v,i)$, the minimum cost of an optimal modified u–restricted k–residence set for the set of nodes $T_v^{(i)} \cup \{u\}$, to also include an upper bound on the load imposed on u. In particular, we define the cost $\hat{R}(u,k,v,i,l)$ of an optimal modified capacitated normal u–restricted k–residence set for the set of nodes $T_v^{(i)} \cup \{u\}$ as

$$\begin{aligned}\hat{R}(u,k,v,i,l) \;=\; \min\Big\{ & \mathrm{C}(S, T_v^{(i)} \cup \{u\}) \;: \\ & S \subseteq T_v^{(i)} \cup \{u\}, |S| = k, \\ & u \text{ covers } v, \\ & \text{and the total load } \textstyle\sum_{\sigma(x)=u} \lambda(x) \\ & \text{on } u \text{ is } \leq l. \Big\} \end{aligned} \tag{35}$$

We call a set S that achieves the minimum in the equation above an *optimal modified capacitated normal u–restricted capacitated k–residence set* for $T_v^{(i)} \cup \{u\}$.

We show a recurrence for computing $\hat{R}(u,k,v,i,l)$ in the following theorem.

Theorem 4: Let v be any vertex of T, let v_i be the ith child of v, $0 \leq i \leq \deg(v)$, and let u be any vertex of T. The cost of an optimal modified capacitated normal u–restricted k–resident set for $T_v^{(i)}$, where the total load imposed on u is $\leq l \leq \Lambda(u)$, is given by

$$\hat{R}(u,k,v,i,l) = \begin{cases} X_0, & \text{if } k = 1,\ i \geq 0, \text{ and} \\ & \sum_{v' \in T_v^{(i)}} \lambda(v') \leq l \\ \min\{X_1, X_2\}, & \text{if } u \notin T_{v_i} \\ X_1, & \text{if } u \in T_{v_i} \\ \text{undefined}, & \text{otherwise.} \end{cases} \tag{36}$$

where

$$X_0 = \sum_{v' \in T_v^{(i)}} \alpha(v') \cdot \delta(v', u) + s(u), \tag{37}$$

$$\begin{aligned} X_1 \;=\; \min_{k_1, l_1} \Big\{ & \hat{R}(u, k_1, v, i-1, l_1) + \\ & \hat{R}(u, k - k_1 + 1, v_i, \deg(v_i), l - l_1) - \\ & s(u) \Big\}, \end{aligned} \tag{38}$$

$$\begin{aligned} X_2 \;=\; \min_{k_2, u'} \Big\{ & \hat{R}(u, k_2, v, i-1, l) + \\ & \hat{R}(u', k - k_2, v_i, \deg(v_i), \Lambda(u')) + \\ & W_{\text{total}} \cdot \delta(u, u') \Big\}, \end{aligned} \tag{39}$$

and where $1 \leq k_1, k_2 \leq k$, $0 \leq l_1 \leq l$, and $u' \in T_{v_i}$ such that $\delta(v_i, u') \leq \delta(v_i, u)$, and $\delta(v, u') \geq \delta(v, u)$. Furthermore, the cost of an optimal modified capacitated normal k–resident set for T_v is given by

$$\min\Big\{ \hat{R}(u, k, v, \deg(v), \Lambda(u)) \;:\; u \in T_v \Big\}, \tag{40}$$

the cost of an optimal capacitated normal k–resident set for T is given by

$$\min\Big\{ \hat{R}(u, k, \mathrm{root}(T), \deg(\mathrm{root}(T)), \Lambda(u)) : u \in T \Big\}, \tag{41}$$

and the cost of an optimal capacitated normal resident set for T is given by

$$\min_{1 \leq k \leq |T|} \Big\{ \hat{R}(u, k, \mathrm{root}(T), \deg(\mathrm{root}(T)), \Lambda(u)) \;:\; u \in T \Big\}. \tag{42}$$

Proof sketch.

The proof of this theorem uses Lemmas 1, and 2, and straightforward extensions of Lemmas 3, and 4.

First, we show the initial conditions for $\hat{R}(u,k,v,i,l)$. By definition of the total modified cost for a capacitated normal residence set for $T_v^{(i)}$ we have that

$$\hat{R}(u,1,v,i,l) = \sum_{v' \in T_v^{(i)}} \alpha(v') \cdot \delta(v', u) + s(u), \tag{43}$$

if $\sum_{v' \in T_v^{(i)}} \lambda(v') \leq l$, for $0 \leq i \leq \deg(v)$. Furthermore, again by definition, $\hat{R}(u,k,v,i,l)$ is undefined when either $i = 0$ and $k \neq 1$, or $k \leq 0$, or $k > |T_v^{(i)} \cup \{u\}|$, or $k = 1$ and $\sum_{v' \in T_v^{(i)}} \lambda(v') > l$.

Second, we show how to compute $\hat{R}(u,k,v,i,l)$ recursively for $k > 1$. There are two cases to consider depending on whether u is a descendant of v_i or not.

Case 1: u is not a descendant of v_i. If u covers v_i then, by a straightforward extension of Lemma 3, we have that

$$\begin{aligned} \hat{R}(u,k,v,i,l) &= \hat{R}(u,k_1,v,i-1,l_1) + \\ & \quad \hat{R}(u,k-k_1+1,v_i,\deg(v_i),l-l_1) - \\ & \quad s(u), \end{aligned} \tag{44}$$

for some integers $1 \leq k_1 \leq k$ and $0 \leq l_1 \leq l$. Hence, if u covers v_i then $\hat{R}(u,k,v,i,l) = X_1$. If u does not cover v_i, then, from a straightforward extension of Lemma 4, we have that

$$\begin{aligned} \hat{R}(u,k,v,i,l) &= \hat{R}(u,k_2,v,i-1,l) + \\ & \quad \min_{u'} \Big\{ \hat{R}(u',k-k_2,v_i,\deg(v_i),\Lambda(u')) + \\ & \quad W_{\text{total}} \cdot \delta(u,u') \Big\}, \end{aligned} \tag{45}$$

where u' is a vertex in T_{v_i} such that $\delta(v,u) \leq \delta(v,u')$ and $\delta(v_i,u') < \delta(v_i,u)$, and k_2 is an positive integer $< k$. Hence, if u does not cover v_i then $\hat{R}(u,k,v,i,l) = X_2$. Therefore, $\hat{R}(u,k,v,i,l) = \min\{X_1, X_2\}$.

Case 2: u is a descendant of v_i. In this case, since u covers v and since v_i is a child of v, it follows that u covers v_i as well. Hence, from a straightforward extension of Lemma 3 we have that

$$\begin{aligned} \hat{R}(u,k,v,i,l) &= \hat{R}(u,k_1,v,i-1,l_1) + \\ & \quad \hat{R}(u,k-k_1+1,v_i,\deg(v_i),l-l_1) - \\ & \quad s(u), \end{aligned} \tag{46}$$

for some integers $1 \leq k_1 \leq k$ and $0 \leq l_1 \leq l$. Therefore, in this case, $\hat{R}(u,k,v,i,l) = X_1$.

Third, by definition, it follows that: (a) the cost of an optimal modified capacitated normal k–resident set for T_v is given by (40); (b) the cost of an optimal capacitated normal k–resident set for T is given by (41); and (c) the cost of an optimal capacitated normal resident set for T is given by (42). ■

The complexity of our algorithm for computing optimal capacitated normal residence sets for a tree with storage costs is given in the following theorem.

Theorem 5: Let T be a tree with n vertices and p be an integer $1 \leq p \leq n$. Let $\Lambda_{\max} = \max\{\Lambda(u) \,:\, u \in T\}$. We can find an optimal capacitated normal p–resident set for T in $O(n^3 p^2 \Lambda_{\max}^2)$ time, and an optimal capacitated normal resident set for T in $O(n^5 \Lambda_{\max}^2)$ time.

Proof: Equation (36) in Theorem 4 can be evaluated by a straightforward dynamic programming algorithm. The running–time of such a dynamic programming algorithm for computing an optimal modified capacitated normal p–resident set for subtrees $T_v^{(i)}, v \in T, 0 \leq i \leq \deg(v)$ via equation (36) is $O(n^3 p^2 \Lambda_{\max}^2)$. Note that $\hat{R}(u,k,v,i,l)$ is defined for at most $n^2 p \Lambda_{\max}$ entries, and computing each entry requires $O(np\Lambda_{\max})$ time. Hence, we can compute an optimal capacitated normal p–resident set for T in $O(n^3 p^2 \Lambda_{\max}^2)$ time, and an optimal capacitated normal resident set for T in $O(n^5 \Lambda_{\max}^2)$ time. ■

Next, consider the case where the number of read and write requests issued by the nodes of the tree are random variables, and the loads imposed by the nodes of the tree to the nodes of a residence set that are covering them are also random variables. Using the algorithm in this section, we can find a normal residence set with minimum average total read, write, and storage cost so that the average load of each node in the residence set does not exceed the node's capacity constraint. To do so, we let $r(v)$, $w(v)$, and $\lambda(v)$ be equal to the average number of read requests, the average number of write requests, and the average load for each node v of the tree, respectively.

V. Conclusions

We described a $O(n^3 p^2)$–time algorithm for finding an optimal residence set of size p for an object on a tree with n nodes, taking into consideration not only the read and write costs, but also the associated storage costs. Furthermore, we described a $O(n^3 p^2 \Lambda_{\max}^2)$–time algorithm for finding minimum total read–write–storage cost normal residence sets for trees, this time also taking into account the load imposed by the nodes of the tree on the nodes in a residence set and their capacity constraints, where $\Lambda_{\max}$ is an upper bound of the capacity of the nodes of the tree.

There are a number of additional important issues to be resolved regarding the problem of finding optimal residence sets. Some of those issues are: (a) finding optimal or near–optimal residence sets for multiple objects, (b) finding optimal residence sets that have additional properties, such as high availability of the replicated object, (c) finding near–optimal residence sets, that consider read, write, and storage costs as well as capacity constraints, in a distributed manner, (d) finding optimal or near–optimal replica schemes for other network topologies, and (e) finding efficient approximation algorithms for the capacitated residence problem for trees with a small constant approximation ratio.

Acknowledgments

The authors would like to thank the referees for their suggestions and comments.

References

[1] O. Wolfson and A. Milo, "The multicast policy and its relationship to replicated data placement," *ACM Transactions on Database Systems*, vol. 16, no. 1, pp. 181–205, 1991.

[2] L. W. Dowdy and D. V. Foster, "Comparative models of the file assignment problem," *ACM Computing Surveys*, vol. 14, no. 2, pp. 287–313, 1982.

[3] M. L. Fisher and D. S. Hochbaum, "Database location in computer networks," *Journal of the ACM*, vol. 27, no. 4, pp. 718–735, 1982.
[4] S. Guha and S. Khuller, "Greedy strikes back: Improved facility location algorithms," in *Proceedings of the 9th ACM–SIAM Symposium on Discrete Algorithms*, 1998.
[5] D. S. Hochbaum, "Heuristics for the fixed cost median problem," *Mathematical Programming*, vol. 22, pp. 148–162, 1982.
[6] D. B. Shmoys, E. Tardos, and K. Aardal, "Approximation algorithms for facility location problems (extended abstract)," in *Proceedings of the 29th ACM STOC*, 1997, pp. 265–274.
[7] S. Arora, P. Raghavan, and S. Rao, "Approximation schemes for euclidean k–medians and related problems," in *Proceedings of the 30th ACM STOC*, 1998, pp. 106–113.
[8] O. Kariv and S. L. Hakimi, "An algorithmic approach to location problems. ii: The p–medians," *SIAM Journal of Applied Mathematics*, vol. 37, no. 3, pp. 539–560, 1979.
[9] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, Massachusetts, 1990.
[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP–Completeness*, Freeman, San Fransisco, 1979.

Konstantinos Kalpakis received the B.S. degree in Computer Engineering and Informatics from the University of Patras, Greece, in 1989, and the M.S. and Ph.D. degrees in Computer Science from the University of Maryland Graduate School, Baltimore, in 1992 and 1994 respectively. He is an Assistant Professor in the Computer Science and Electrical Engineering Department at the University of Maryland Baltimore County since the Fall of 1996. Dr. Kalpakis was a visiting computer scientist at the National Institute of Standards and Technology and a research assistant professor at UMBC from 1994 until 1996, working in the areas of digital libraries and electronic commerce.

Dr. Kalpakis's research interests are in the areas of parallel and distributed systems, digital libraries and electronic commerce, database systems, and combinatorial optimization. Dr. Kalpakis is a member of the ACM, IEEE Computer Society, Upsilon Pi Epsilon, and an associate member of Sigma Xi.

Koustuv Dasgupta received his BS degree in Computer Science and Engineering from Jadavpur University, India in 1997. He received his MS in Computer Science from the University of Maryland, Baltimore County in 2000 and is currently pursuing his PhD at the same University. His research interests include distributed systems, data replication and Internet technologies.

Ouri Wolfson's main research interests are in database systems, distributed systems, transaction processing, and mobile computing. He received his B.A. degree in mathematics, and his Ph.D. degree in computer science from Courant Institute of Mathematical Sciences, New York University. He is currently a Professor in the Department of Electrical Engineering and Computer Science at the University of Illinois at Chicago, where he directs the Databases and Mobile Computing Laboratory. He is also a consultant to Argonne National Laboratory, to the US Army Research Laboratories, and to the Center of Excellence in Space Data and Information Sciences at NASA. Before joining the University of Illinois he has been on the computer science faculty at the Technion and Columbia University, and he has been a Member of Technical Staff at Bell Laboratories.

Ouri Wolfson authored over eighty publications in leading journals and conference proceedings. He is a Fellow of the Association of Computing Machinery, an editor of the ACM/URSI/Baltzer Wireless Networks Journal, a Member of the ACM SIGMOD Digital Review Editorial Board and a guest editor of the ACM/Baltzer Journal on Special Topics in Mobile Networks. From 1991 to 1996 he served as a National Lecturer for the Association of Computing Machinery professional society. He participated in numerous conferences (including ACM-SIGMOD, VLDB, PODS, ICDE, NGITS, ICDCS, MOBIDATA, DOOD, SSD, GIS, PDIS, CIKM) as a program committee member, keynote speaker, session chairman, and panelist. Most recently he was the program committee co-chair of the First International Workshop on Satellite-based Information Systems (Wosbis), and the general co-chair of the Second International Workshop on the same topic. He was also the General co-Chairman of the IEEE Knowledge and Data Engineering Exchange Workshop, and he serves on the Advisory Committee of the NSF Center of Research Excellence in Science and Technology, at Florida A&M University. His work has been funded by the National Science Foundation, Air Force Office of Scientific Research, Defense Advanced Research Projects Agency, NATO, US Army, the New York State Science and Technology Foundation, Hughes Research Laboratories, and Informix Co.

Appendix A. Sample Tree Network and Residence Sets

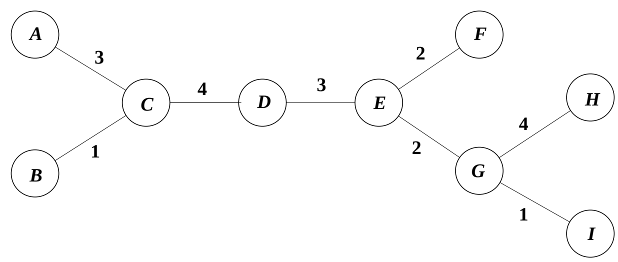

Fig. 4. A tree network with 9 nodes, and with the edge lengths shown next to each edge.

TABLE I

The number of read and write requests issued by the nodes of the tree in Figure 4, as well as the node storage costs. The total number of reads is 654 and the total number writes is 124.

Node	No. Reads	No. Writes	Storage Cost
A	2	1	5
B	6	1	5
C	312	3	20
D	1	5	20
E	120	69	5
F	8	21	0
G	99	5	5
H	6	6	5
I	100	13	1

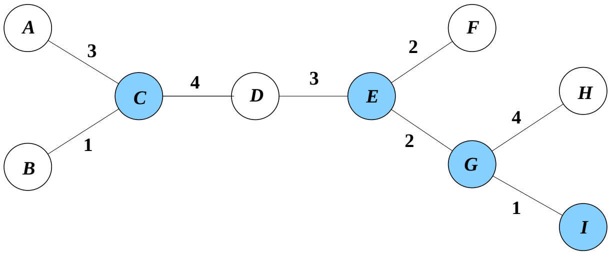

Fig. 5. An optimal 4–residence set $S = \{C, E, G, I\}$ for the tree in Figure 4, without considering write and storage costs, and with the number of reads given in Table I. Gray nodes are in the residence set. The total (read) cost of S is 55.

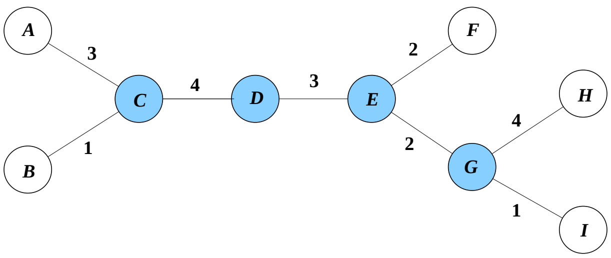

Fig. 6. An optimal residence set $S = \{C, D, E, G\}$ for the tree in Figure 4, with both read and write requests, given in Table I, but without taking into account any storage costs. Gray nodes are in the residence set. The read, write, and total costs of S are 152, 1199, and 1351 respectively.

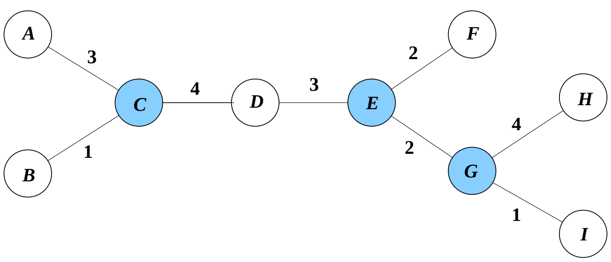

Fig. 7. An optimal residence set $S = \{C, E, G\}$ for the tree in Figure 4, with both read and write requests, and storage costs given in Table I. Gray nodes are in the residence set. The read, write, storage, and total costs of S are 155, 1269, 30, and 1454 respectively. Note that if D is included in S, in order to have a connected residence set, the total cost of the resulting residence set is 1471.

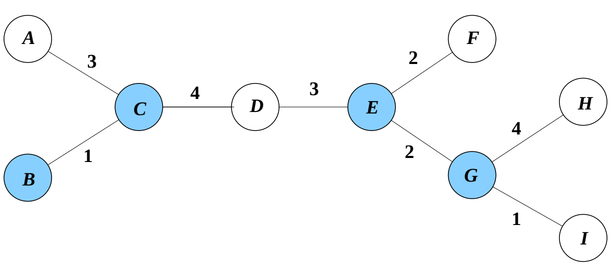

Fig. 8. An optimal residence set $S = \{B, C, E, G\}$ for the tree in Figure 4, with both read and write requests, and storage costs given in Table I, with the exception of node C whose storage cost is assumed to be 163 only for this case. Gray nodes are in the residence set. The read, write, storage, and total costs of S are 149, 1515, 178, and 1842 respectively.

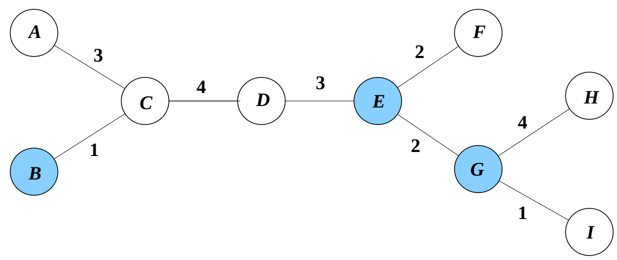

Fig. 9. An optimal residence set $S = \{B, E, G\}$ for the tree in Figure 4, with both read and write requests, and storage costs given in Table I, with the exception of node C whose storage cost is assumed to be 164 for this case only. Gray nodes are in the residence set. The read, write, storage, and total costs of S are 463, 1366, 15, and 1844 respectively.