Distributed Selfish Replication under Node Churn

Eva Jaho, Ina Jaho and Ioannis Stavrakakis

Department of Informatics and Telecommunications
National & Kapodistrian University of Athens, Greece

E-mail: {grad0836@di.uoa.gr}, {grad0839@di.uoa.gr}, {ioannis@di.uoa.gr}

***Abstract*— In this paper the impact of node churn on the effectiveness of a distributed selfish replication group is investigated. In such a group, nodes cooperate in deciding which objects to store, so that the cost of providing content to their clientele be decreased, compared to that induce when operating in isolation. In order for a cooperation scheme to be sustainable, nodes should not be mistreated; i.e., their performance should never be lower than that in isolation. In this paper it is shown that node churn can introduce mistreatment to otherwise mistreatment-free cooperation schemes. Finally, by properly modifying a previously described scheme that looses its mistreatment-free property under node churn, a scheme is developed that maintains the mistreatment-free property in a distributed selfish replication group in the presence of node churn.**

I. INTRODUCTION

We consider a group of nodes that collectively replicate content as shown in Figure 1. A network node caches or replicates content in its storage in order to provide it to local or remote nodes effectively and economically. Network nodes can decide on their own on the content they will store or they can cooperate with other nodes that are in close distance, form a group, and collectively decide which content to store. A user's request is first received by the local node. If the requested object is stored locally, it is returned to the requesting user immediately, thereby incurring a minimal access cost. Otherwise, the requested object is searched for, and fetched from other nodes of the group, at a potentially slightly (due to the proximity) higher access cost. If the object can not be located anywhere in the group, it is retrieved from an origin server, which is assumed to be outside the group, thus incurring a maximal access cost [2].

A key design issue in this distributed replication group is to ensure that the object replication strategy employed does not mistreat any of the participating nodes. If a node is mistreated (i.e., the average access cost for objects requested by its clientele (users having this node as the first place to look for content) is higher than the cost incurred if it were not part of the group), then the node will leave the group. In Laoutaris et al. [1] a mistreatment-free policy (referred to as TSLS) has been devised based on a game-theoretic formulation. A key assumption in developing the TSLS policy has been that nodes are always available and that there is no node churn in the group.

This work has been supported in part by the following projects funded by the IST FET Program of the European Commission: CASCADAS (Component-ware for Autonomic Situation-aware Communications, and Dynamically Adaptable Services) under contract FP6-027807, BIONETS (BIOlogically-inspired autonomic NETworks and Services) under contract FP6-027748, and NoE CONTENT (IST-384239).

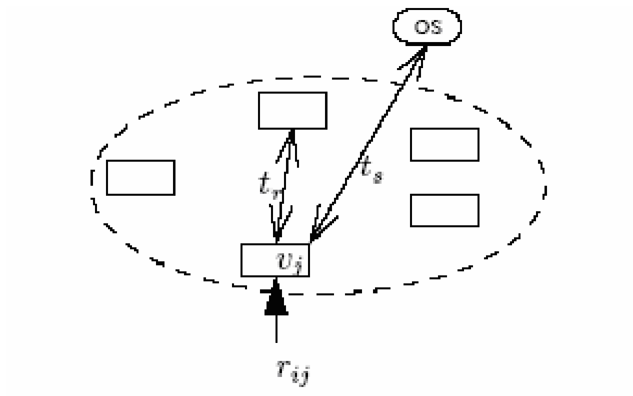

Fig. 1. A distributed replication group

In the present work we first calculate the access cost under the TSLS policy and then under the greedy local (GL) policy (i.e. when each node decides on the content to store in isolation and not as part of a group) and we verify that TSLS can never induce an access cost higher than the GL policy. In the sequel, we assume that each of the cooperating nodes is not always available but node churn occurs with some probability. The resulting costs are derived and it is shown that the TSLS policy looses its mistreatment-free property. Finally, the original TSLS strategy is modified and a new strategy is developed that is shown to maintain the mistreatment-free property in the presence of node churn.

II. DISTRIBUTED SELFISH REPLICATION UNDER NODE CERTAINTY

When a user requests an object, its local node returns it, if the object is stored locally incurring a cost t_l. Otherwise, if the object is not stored locally, then the object is searched in the storage of the other nodes of the group and is returned if at least one of them has it incurring a cost t_r. Finally, if the object is not found in any of the nodes of the group or in the local node, then the object is received from the origin server with a cost t_s, where $t_l \leq t_r \leq t_s$. The total cost $C_j(P)$ of a node v_j for accessing all the objects (under the global placement referred to as P), is given by:

$$C_j(P) = \sum_{o_i \in P_j} r_{ij} \cdot t_l \ + \sum_{o_i \notin P_j, o_i \in P_{-j}} r_{ij} \cdot t_r \ + \sum_{o_i \notin P} r_{ij} \cdot t_s \ ,$$

where r_{ij} denotes the rate at which node v_j requests object o_i, P_j is the placement of node v_j, which is the set of objects replicated at this node and P_{-j} is the set of objects collectively held by nodes in the group other than v_j.

As shown in [1], TSLS achieves a pure Nash equilibrium [4], and the resulting local utility of each node is at least as good as that under a GL placement strategy and possibly better ($C_j^{TSLS}(P) \leq C_j^{GL}(P')$).

These conclusions hold good for distributed systems where participating nodes are always available (ON). Very frequently though node churn is present, due to the autonomicity of the nodes or simply temporary unavailability of various reasons (resource constraints or faults), which can increase costs or decrease service quality [3]. The first question asked here is the following: *Would TSLS under the cooperative replicating of objects still be more effective than GL and mistreatment be still avoided*? The answer is no, as shown in the next section, where in addition a modified strategy is introduced that maintains the mistreatment-free property under node churn.

III. Distributed Selfish Replication under Node Uncertainty

When a user requests an object, its local node returns it at the (low) cost t_l as long as the object is stored locally and the node is ON (with a probability π). Otherwise, if the object cannot be provided by the local node (i.e., if the object is not stored locally, or if it is but the node that has it is OFF), then the object is searched in the storage of other nodes of the group and is returned if at least one of them has it and is ON, at the (higher) cost t_r. Finally, if the object cannot be provided by the local node or any of the nodes of the group then the object is received from the server with a maximal cost t_s. In this case, the average total cost $C_j(P)$ of a node v_j for accessing all the objects, is given by:

$$C_j(P) = \sum_{o_i \in P_j} r_{ij} \cdot t_l \cdot \pi_j + \sum_{o_i \notin P} r_{ij} \cdot t_s$$
$$+ \sum_{o_i \notin P_j, o_i \in P_k} \left[r_{ij} \cdot t_r \cdot [1 - \prod_{\substack{k=1, k \neq j, \\ k: o_i \in P_k}}^{n} (1 - \pi_k)] \right]$$
$$+ \sum_{o_i \in P_j, o_i \in P_k} \left[r_{ij} \cdot t_r \cdot (1 - \pi_j) \cdot [1 - \prod_{\substack{k=1, k \neq j, \\ k: o_i \in P_k}}^{n} (1 - \pi_k)] \right]$$
$$+ \sum_{o_i \in P} \left[r_{ij} \cdot t_s \cdot \prod_{k=1, k: o_i \in P_k}^{n} (1 - \pi_k) \right] .$$

Under node uncertainty, we develop a modified TSLS strategy (referred to as a mistreatment-free object placement (MfOP) strategy) that is shown to be mistreatment-free under node churn (see upcoming report in [5]).

The original TSLS policy requires each node to know only its local demand pattern and the objects selected for replication by remote nodes, whereas the MfOP policy requires additionally that each node knows its probability and the probabilities of the other remote nodes to be available.

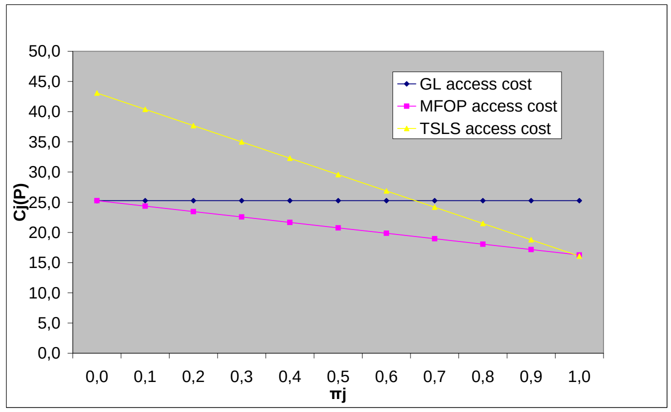

Fig. 2. Access cost under MfOP, TSLS and GL

IV. Results

This work has analytically evaluated the access cost under both cases: when nodes are always available and when they are available with some probability π. The results shown in Figure 2 illustrate the loss of the mistreatment-free property by TSLS under node churn and the preservation of this property by the proposed MfOP policy. Notice that the average total cost under the GL policy is always higher than that the MfOP policy, while it can be lower than that under the TSLS policy under high node churn rates.

References

[1] Nikolaos Laoutaris, Orestis Telelis, Vassilios Zissimopoulos and Ioannis Stavrakakis, "Distributed Selfish Replication".

[2] Avraham Leff, Joel L. Wolf, and Philip S. Yu, "Replication algorithms in a remote caching architecture," IEEE Transactions on Parallel and Distributed Systems, vol. 4, no. 11, pp. 1185-1204, Nov.1993.

[3] Philip Brighten Godfrey, Scott Shenker, and Ion Stoica, "Minimizing Churn in Distributed Systems". March 17, 2006.

[4] Martin J. Osborne and Ariel Rubinstein, A Course in Game Theory, MIT Press, 1994.

[5] http://cgi.di.uoa.gr/ istavrak/publications.html.