Evaluation of Pseudo-Active Replication in Wide-Area Networks *

1R－5

Hiroaki Higaki, Nobumitsu Morishita, and Makoto Takizawa †
Tokyo Denki University ‡
{hig,nobu,taki}@takilab.k.dendai.ac.jp

1 Introduction

According to the advance of computer and network technologies, network applications are widely developed. These applications are realized by the cooperation of multiple *objects*. Here, mission critical applications are also implemented and these applications are required to be executed fault-tolerantly. *active replication* has been proposed where multiple replicated objects are operational in the network system. In the conventional *active replication* all the replicated objects are required to be synchronized. In the network environment, each replicated objects may be placed on different kinds of computes so that the synchronization induces additional time-overhead. Therefore *pseudo active replication* was proposed [4]. However, the proposed protocol for the *pseudo active replication* has been proposed for local-area networks with different kinds of computers. In this paper we extend the *pseudo active replication* to be used in wide-area and large scale network environment and propose a novel protocol.

2 Pseudo-Active Replication

There are two main approaches for replication technic : *passive replication* and *active replication*. In the *passive replication*, only one replica o^s_{j1} is operational. Another replicas o^s_{jk} ($2 \leq k \leq n$) are not operational, however o^s_{jk} get the newest state information from o^s_{j1} and o^s_{jk} update the state information. However if o^s_{j1} fails, recovery procedure takes time because one of the replica o^s_{j2} becomes operational and o^s_{j2} has to re-executed before failure. In the *active replication*, all replicas o^s_{jk} are operational. Each client sends request message to all replicas, and they return the result that operational the request. However each client o^c_i accepts these messages and deliver the result to the application after receiving all the messages from o^s_{jk}, therefore it takes response time and synchronization overhead. That is, the synchronization overhead for receiving the response is required to be reduced. Therefore *pseudo active replication* was proposed. In *pseudo active replication*, o^c_i only wait for the first response from the replicas. On receiving the first response, o^c_i continues to execute the application. Thus, the synchronization overhead is reduced. However, since o^s_{jk} are placed on processors with different speed and are not synchronized, some replica $o^s_{jk'}$ might finish the computation for all the request from the clients an another replica $o^s_{jk''}$ might keep many requests not to be computed because $o^s_{jk''}$ is placed on a slow processor. If $o^s_{jk'}$ fails, the recovery procedure takes time because $o^s_{jk''}$ has to execute the methods that $o^s_{jk'}$ has already executed before the failure as in the passive replication. In order to solve this problem, (1) each client object tells the serve replicas which server is fast, and (2) if o^s_{jk} finds to be slower, it omits some methods requested by clients to catch up the faster server replicas [Figure 1]

[Faster / Slower request] If the response from o^s_{jk} has been received and that from $o^s_{jk'}$ has not yet which o^c_i sends a request to o^s_{jk}, o^c_i informs that o^s_{jk} is faster replica and $o^s_{jk'}$ is a slower one.

[Omissible request] If an operation *op* is *identity* or *idempotent*, *op* is defined to be omissible [2]

[Omission rule] If the following conditions are satisfied, and operation

op is omitted in o^s_{jk}:

1. o^s_{jk} is a slower replica.
2. *op* is an omissible operation.
3. Some $o^s_{jk'}$ has already executed *op*.

In [2] and [4],by using vector clocks [3], rule 1 and 3 are checked in o^s_{jk}. In addition, every request is assumed to be delivered to all the server replicas in the same order, i.e. *totally ordered delivery* is assumed.

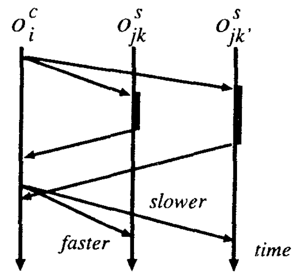

Figure 1: Pseudo-Active Replication

3 Pseudo-Active in Wide-Area Network

Replication technique with an environment of different kinds of computers is being discussed until now. However each replica must be distributed geographically when it thinks the failure of earthquake. Therefore we consider not only processing speed but also network delay and reliability. In *pseudo active replication* with wide area network, the information of processing speed on each replica has processing time and trancemission time. Therefore each replica can not recognize an obvious processing time [Figure 2]. Hence, we think about case that it can not be adapted. In this paper, we extended *pseudo active replication* to consider the environment which difference processing speed and network delay.

In this section, we propose another protocol for *pseudo active replication* based on the total ordering protocol [1]. Each replicas o^s_{jk} manipulates the following variables:

---

*WANのための擬似能動的多重化の評価
†桧垣 博章　森下 展光　滝沢 誠
‡東京電機大学

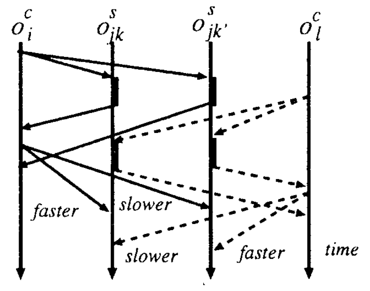

Figure 2: Pseudo-Active in Wide-Area Networks

- Logical clock ck_{jk} for totally ordering the requests from clients.
- Number of executed operations ne_{jk} for the measurement of processing speed of server replica o^s_{jk}.

In the following total ordering protocol, the information which operations have been executed in every replicas o^s_{jk} is executed among the replicated servers [Figure 3].

1. A client o^c_i sends request message $req(op)$ with an operation op to all replicas o^s_{jk} $(1 \leq k \leq n_j)$.
2. On receipt of $req(op)$, o^s_{jk} stores op in the buffer with ck_{jk}. o^s_{jk} sends back an ordering message $ord(ck_{jk}, ne_{jk})$ piggy backing ck_{jk} and ne_{jk}. ck_{jk} is incremented by one.
3. After receiving all the ordering messages from o^s_{jk} $(1 \leq k \leq n_j)$, o^c_i sends final messages $fin(max\ ck, max\ ne)$ where $max\ ck = max_k\ ck_{jk}$ and $max\ ne = max_k\ ne_{jk}$.
4. On receipt of $fin(max\ ck, min\ ne)$, op is restored from the buffer and enqueued to an application queue APQ_{jk} ordered by $oi(op) = max\ ck$.

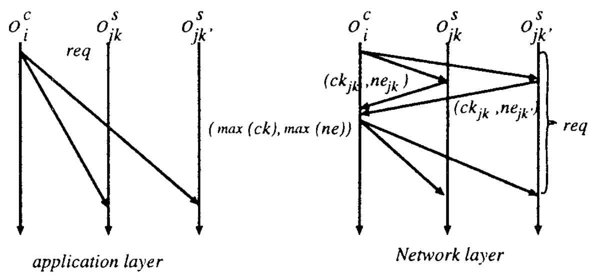

Figure 3: Protocol for Pseudo-Active Replication

o^s_{jk} compares $max(ne)$ and own ne_{jk} after receiving $fin(max\ ck, max\ ne)$. If $max\ ne$ equal to own ne_{jk}, it is fastest replica and, if $max\ ne$ is larger than own ne_{jk}, it is slower replica. Slower replica invokes the catch up procedure that slower replica omits the omissible operations in APQ to catch up with fastest replica.

4 Evaluation

Here, we evaluate our protocol described in the previous section by comparing with the protocol in [2]. Here we assume that there are two server replicas o^s_{j1} and o^s_{j2}. Difference qd_j among numbers of request messages in APQ_{j1} and APQ_{j2} of o^s_{j1} and o^s_{j2} are used to as the measurement for our protocol. We measured the average of qd_j which is difference message between o^s_{j1} and o^s_{j2} in APQ ($qd_j = |\ APQ_{j1} - APQ_{j2}\ |$). The smaller qd_j is, the shorter the recovery time is. The evaluation result is shown in Figure 4. In this Figure, T_c[sec] denotes the interval of two successive request messages in a client and P_r and P_w denote the probability of requests of *read* and *write* ($P_r + P_w = 100$ [%]). In every evaluation environment, qd_j, in our protocol is smaller than in a conventional one. That is, less recovery time is required in our protocol. Moreover, consider the case where $P_w = 0$ ($P_r = 100$ [%]). Here, all the requested operations are read ones. That is, all the queued requests in APQ of slower replica are omitted in both protocols. Therefore, the result shows there are more chances to detect slower replicas in our protocol.

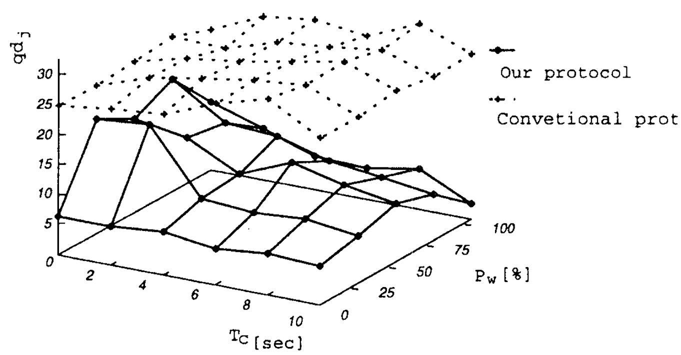

Figure 4: Evaluation for result

5 Concluding Remarks

In order to apply the pseudo-active replication in a wide-area and large-scale network systems, we proposed another protocol designed by modifying the total ordering protocol. In order to make clear the efficiency of our protocol, we have implemented our protocol and a conventional one and evaluate their performances in a simulation environment of a wide-area network. The evaluation results show that

- there are more chances to detect slower replicas
- shorter recovery time is required in a failure of replica in our protocol than in a conventional protocol.

The future work is to evaluate our protocol in a real wide-area network, e.g. in the Internet.

References

[1] Birman, K.P and Joseph, T.A., "Reliable Communication in the Presence of Failures," ACM Trans. on Computer Systems, Vol. 5, No. 1, pp. 47-76 (1987).

[2] Ishida, T., Higaki, H. and Takizawa, M., "Pseudo-Active Replication of Objects in Heterogeneous Processors," IPSJ Technical Report, vol. 98, No.15, pp.67-72 (1998).

[3] Mattern, F., "Virtual Time and Global States of Distributed Systems," Parallel and distributed Algorithms, North-Holland, pp. 215-226 (1989).

[4] Shima, K., Higaki, H. and Takizawa, M., "Pseudo Active Replication in Heterogeneous Clusters," IPSJ Trans., Vol. 39, No.2, pp.379-387 (1998).