Multimedia Checkpoint Protocol Based on Recovery Overhead *

3L−08

Shinji Osada and Hiroaki Higaki †
Tokyo Denki University §

1 Introduction

Advanced computer and network technologies have lead to the development of computer networks. Here, an application is realized by multiple processes located on multiple computers connected to a communication network such as the Internet. Each process computes and communicates with other processes by exchanging messages through communication channels. Mission-critical applications are required to be executed fault-tolerantly. That is, even if some processes fail, execution of an application is required to be continued. One of the important methods to realize fault-tolerant networks is *checkpoint-recovery*. During failure-free execution, each process takes local checkpoints by storing state information into a stable storage If a certain process fails, the processes restart from the checkpoints by restoring the state information from the stable storage. For restarting execution of applications correctly in conventional data communication networks, a set of local checkpoints taken by all the processes and from which the processes restart should form a *consistent global checkpoint*. A global checkpoint is defined to be consistent if there is neither *orphan* nor *lost message*. However, in a multimedia communication network, applications require transmission of large-size multimedia messages and low overhead failure-free execution rather than complete consistency. Hence, this paper proposes a novel criteria for consistent global checkpoints based on properties of multimedia communication networks and applications. Hence, this paper proposes a novel criteria for consistent global checkpoints based on properties of multimedia communication networks and applications.

2 Consistency

If a process is required to take a local checkpoint during communication event, the process postpones taking it after the event. Hence, synchronization overhead for checkpointing and recovery gets high. In order to solve this problem, a local checkpoint is taken even during a communication event. Here, the conventional criteria for consistency of a global checkpoint cannot be applied and a novel criteria is required. In a multimedia network, a message is composed of multiple packets. Hence, a message sending event is composed of a sequence of multiple packet sending events and a message receipt event is composed of a sequence of multiple packet receipt events. A local checkpoint may be taken between two successive packet sending events or packet receipt events. Here, each packet has distinct value. For example in an MPEG data transmission, value of a packet for an I-picture is higher than that for a B-picture. Based on timing relation among local checkpoints, a packet sending event and a packet receipt event, orphan and lost packets are defined. Even if there is an orphan packet, the checkpoints are consistent since the packet is surely retransmitted after recovery. However, a lost packet reduces message consistency since a lost packet is never retransmitted after recovery. Hence, mesage consistency depends on only lost packets. Part of a message may be lost for a multimedia application. The more part of a message is lost, the more global consistency is lost. In addition, compatibility with the conventional consistency is kept. The evaluated value of global consistency is in a closed interval b etween 0 and 1.

3 Recovery time

As discussed in the previous section, the newly introduced consistency of a global checkpoint is induced only by the number of lost packets. In recovery, before a process p which is receiving a messages when it takes a local checkpoint restarts execution of an application, p has to wait until p receives all the retransmitted orphan packets from communication channels destined to p. Thus, the more orphan packets are, the longer recovery time is. Hence, we introduce the number of orphan packets as a metric of global checkpoint.

4 State infomation

During failure-free execution, each process takes local checkpoints by storing state information into a stable storage. If a certain process fails, the system restarts from the global state. Here, every process restores the state information from the stable storage and restarts execution of an application from the local checkpoint. Here, we focus communication buffer which is one of the state information. Every process has a message transmission and reception buffer, repectively. A transmitted message is divided into the packet, and stored in the transmitting buffer by the application. Thus, by removing orphan packets from the transmission buffer before the state information is stored into the stable storage, shorter recovery time is achieved. In Figure 1, packets $pa_1^1, \ldots, pa_5^1$ into which a message m_1 is decomposed are stored in a transmission buffer. Here, three packets pa_1^1, pa_2^1, pa_3^1 are orphan packets. By removing these packets before storing the buffer into the stable storage for taking a checkpoint, the shorter recovery time is achieved after restoring the buffer from the stable storage since these packets are not required to be re-transmitted.

5 Checkpoint protocol

Here, we design a checkpoint protocol based on recovery overhead according to the definition. The protocol is based on a 3-phase coordinated checkpoint protocol. In a data communication network, for avoiding inconsistent messages, each process is required to be blocked, i.e. suspend execution of an application for a

*リカバリオーバヘッドに基づいたマルチメディアチェックポイントプロトコル

†長田 慎司　桧垣 博章
‡{shinji, hig}@higlab.k.dendai.ac.jp
§東京電機大学

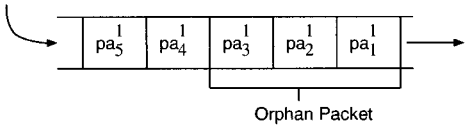

Figure 1: Communication buffer.

certain period. However, for time-bounded failure-free execution of an application, our protocol does not require processes to suspend execution of an application during checkpoint protocol. Each process p_i takes a local checkpoint c_i immediately when p_i is required to take c_i.

5.1 Checkpoint protocol $\mathcal{P}_{\mathcal{E}}$

We proposed protocol that improves consistency by making occurrence probability of the lost packet decrease. In this protocol, checkpointing by the following process.

[Checkpoint Protocol $\mathcal{P}_{\mathcal{E}}$]

1) Let RC be required global consistency. p_c sends a checkpoint request message Req to every process.
2) On receipt of the Req, each process p takes a tentative local checkpoint and sends back a acknowledgement message Ack. If process is receiving a messsage, Here, if p is receiving an application message when p receives the Req, the process p postpones taking a checkpoint for $tau - 2\delta$ where τ is the time-bound for checkpointing and 2δ is the roundtrip time between p_c and p. In addtion, for evaluation of global consistency, needed information is carried by Ack message.
3) On receipt of all the Ack, p_c calculates consistency. If calculated consistency is higher than required one, p_c sends $Done$ message to every process. Otherwise, p_c sends $Cancel$ message to every process.
4) On receipt of $Done$, each process changes tentative checkpoint to a stable local checkpoint. On receipt of $Cancel$, each process discards tentative checkpoint.

This protocol realizes to achieve higher consistency by receiver process delaying taking checkpoint.

5.2 Extended protocol $\mathcal{P}_{\mathcal{E}}{}^{+}$

As discussed in Section 3, the more orphan packets are, the longer recovery time is required. Hence, for supporting time-bounded execution of an application, this section proposes the following checkpoint protocol based on the number of orphan packets.

[Extended protocol $\mathcal{P}_{\mathcal{E}}^{+}$] In $\mathcal{P}_{\mathcal{E}}^{+}$, due to postponing taking checkpoint in a process which is recieving a message, the probability of suffering orphan packets becomes higher. In an extended protocol $\mathcal{P}_{\mathcal{E}}^{+}$, the coordinator process calculates the number O_{ij} of orphan packets in all the communication channels between two processes p_i and p_j. This is realized by using the information carried by Ack messages. O_{ij} is carried by Done message destined to p_i. On receipt of this Done message, p_i takes some packets out of the message buffer buf_{ij} before buf_{ij} is storing into the message log in a stable storage. The procedure for taking a checkpoint in $\mathcal{P}_{\mathcal{E}}^{+}$ is as follows:

1) In cast that there are orphan packets in a communication channel $\langle p_i, p_j \rangle$, p_c calculates the number of orphan packets in $\langle p_i, p_j \rangle$
2) p_c informs p_i of the number N of orphan packets in $\langle p_i, p_j \rangle$ by carrying N with Done message in the checkpoint protocol.
3) On receipt of the Done messages, p_i removes N packets from the transmission buffer before p_i stores its state information into the stable storage.

If multiple message sending event occur concurrently in p_i, by removing N successive packets from the transmission message buffer, some packets in the transmission buffer may become lost packets. In the worst case, the global consistency becomes lower than the required consistency. Hence, the number of removed packets should be controlled not to make the achieved global consistency less than the required one. Hence, the following procedure is added to the above step 1).

- pc calculates the consistency achieved by removing all the orphan packets from the transmission buffers.
- If the calculated consistency is higher than the required one, the orphan packets are really removed from the transmission buffer and then stored them into the stable storage for taking checkpoints. Otherwise, without removing orphan packets, the packets in the transmission buffers are stored into the stable storage for taking checkpoints.

By using the above protocol, without reducing the achieved global consistency, less recovery time is achieved.

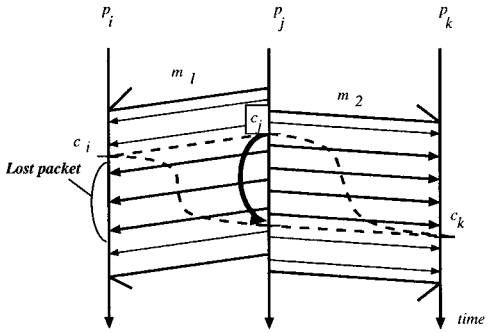

Figure 2: Extended Protocol $\mathcal{P}_{\mathcal{E}}^{+}$.

6 Conclusion and Remarks

The protocol proposed in this paper not only avoids the reduction of the global consistency due to lost packets but also achieves shorter recovery time for retransmission of orphan packets. In future work, the authors will evaluate the performance of the checkpoint protocol by applying it to the MPEG-2 data transmission.

References

[1] Osada, S., Hiraga, K. and Higaki, H., "Qos based Checkpointing Protocol in Multimedia Network Systems," The Annual IEEE International Workshop on Fault-Tolerant Parallel and Distributed Systems, CD-ROM (2001).