Script **generated by TTT**

Title: Distributed_Applications (18.06.2012)

Date: Mon Jun 18 09:15:29 CEST 2012

Duration: 46:14 min

Pages: 11

Group communication

Introduction

Group communication facilities the interaction between groups of processes.

Generated by Targeteam

Message delivery

Message delivery is an important issue of group communication; two aspects are relevant:

a) ***who*** gets the message, and

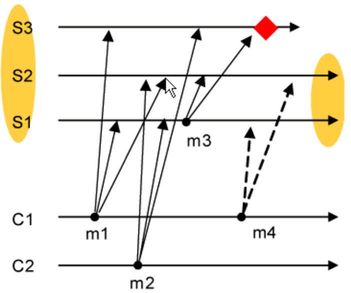
b) ***when*** is the message delivered.

Generated by Targeteam

Sequence of message delivery

It is desired to deliver all messages sent to the group G to all group members of G in the ***same sequence*** , because otherwise we might get non-deterministic system behavior.

Example for group reconfiguration



m4 is sent by C1 before the group composition is modified. However, in order to guarantee atomicity, m4 should not be delivered to S1 and S2 (since, due to the crash, it is no longer possible to deliver m4 to S3).

Generated by Targeteam

Delivery of messages without delay in the same sequence is not possible in a distributed system ⇒ ordering methods for message delivery.

synchronously, i.e. there is a system-wide global time ordering.

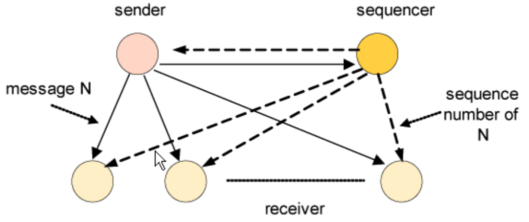
loosely synchronous, i.e. consistent time ordering, but no system-wide global (absolute) time.

[Total ordering by sequencer](#)

[Virtually synchronous ordering](#)

[sync-ordering](#)

A selected group member serializes all the messages sent to the group.



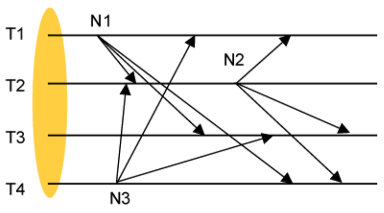
1st step: the sender distributes the message N to ***all*** group members;

2nd step: sequencer (serializer) determines a sequence number for N and distributes it to all group members; delivery of N to the application processes takes place according to this number.

determination of a correct sequence based on the ***before*** relation between two events modeling their causal dependency (see [causally distributed breakpoints](#)).

Example

1. T_1 sends N_1 , and T_2 sends N_2 with N_2 dependent on N_1
2. T_4 sends N_3 with N_1 and N_3 concurrent
3. at T_2 : N_3 is received before N_1
4. at T_3 : N_3 is received after N_1



Ordering for message delivery

Delivery of messages without delay in the same sequence is not possible in a distributed system ⇒ ordering methods for message delivery.

synchronously, i.e. there is a system-wide global time ordering.

loosely synchronous, i.e. consistent time ordering, but no system-wide global (absolute) time.

[Total ordering by sequencer](#)

[Virtually synchronous ordering](#)

[sync-ordering](#)

Delivery of messages without delay in the same sequence is not possible in a distributed system ⇒ ordering methods for message delivery.

- synchronously, i.e. there is a system-wide global time ordering.
- loosely synchronous, i.e. consistent time ordering, but no system-wide global (absolute) time.

Total ordering by sequencer

Virtually synchronous ordering

sync-ordering

Depending on the message delivery guarantee, five classes of multicast services can be distinguished.

1. ***unreliable multicast*** : an attempt is made to transmit the message to all members without acknowledgement; at-most-once semantics with respect to available members; message ordering is not guaranteed.
2. ***reliable multicast*** : the system transmits the messages according to "best-effort", i.e. the "at-least-once" semantics is applied.
   - B-multicast primitive: guarantees that a correct process will eventually deliver the message as long as the multicaster does not crash.
   - B-deliver primitive: corresponding primitive when a message is received.
3. ***serialized multicast*** : consistent sequence for message delivery; distinction between
   - totally ordered
   - causally ordered (i.e. virtually synchronous)
4. ***atomic multicast*** : a reliable multicast which guarantees that either all operational group members receive a message, or none of them do.
5. ***atomic, serialized multicast*** : atomic message delivery with consistent delivery sequence

Taxonomy of multicast

Multicast messages for constructing distributed systems based on group communication;

- different multicast communication semantics

Multicast classes

Relationship between multicast classes

Multicasting can be realized by using IP multicast which is built on top of the Internet protocol IP.

- Java API provides a datagram interface to IP multicast through the class `MulticastSocket` .

Multicast messages for constructing distributed systems based on group communication;

- different multicast communication semantics

Multicast classes

Relationship between multicast classes

Multicasting can be realized by using IP multicast which is built on top of the Internet protocol IP.

- Java API provides a datagram interface to IP multicast through the class `MulticastSocket` .

Generated by Targeteam