Distributed Snapshot

A java library implementation

Luca Danelutti(XXXX)
Gian Marco Paldino (XXXXX)
Ian Di Dio Lavore (XXXX)

07-04-2021

Contents

1 Introduction

Distributed systems cannot take advantage neither of shared global memory nor of a shared global clock. Consequently, the problem of creating a global snapshot, i.e., a description of the system state at a specific point in time, is nontrivial.

A solution to the problem was proposed by K. Chandy and L. Lamport in [1]. Their proposed solution is based on the sending of a *marker*, i.e. a unique identifier for a global snapshot, that communicates to the entities involved that a snapshot is being created and that they must record their state and incoming messages.

In this work we implement the Chandy-Lamport proposed solution by providing a library that allows a distributed application developer to conveniently add the functionality of a distributed snapshot to its project.

1.1 Project statement

Implement a library that offers the capability of storing a distributed snapshot on disk. The library should be state and message agnostic. Implement an application that uses the library to cope with node failures (restarting from the last valid snapshot).

1.2 Acronyms

1.3 Assumptions

- Nodes do not crash in the middle of the snapshot.

- The topology of the network (including the set of nodes) does not change during a snapshot.

- Multiple snapshots may run in parallel.

References

[1] K. Mani Chandy and Leslie Lamport. "Distributed Snapshots: Determining Global States of Distributed Systems". In: *ACM Trans. Comput. Syst.* 3.1 (Feb. 1985), pp. 63–75. ISSN: 0734-2071. DOI: 10.1145/214451.214456. URL: https://doi.org/10.1145/214451.214456.