Luca Simoncini's retirement seminar

Andrea Bondavalli
Felicita Di Giandomenico
Roberto Baldoni


20
Years

And now the serious part…..

Long lasting contribution to the Dependability Community

Since early 70s until today, Luca has covered prominent roles, such as:

- Founder and Chairman of several European Workshops on Dependable Computing
- Founder member of the IFIP Working Group 10.4 "Dependable Computing and Fault-Tolerance" (established in October 1980)
- Active in International Conferences (as PC member, PC chair, General Chair, SC member, invited speaker, author of papers): FTCS, DSN, SRDS, ICDCS, DCCA, EDCC, ISADS.
- Chairman of the Italian National Group on "Dependable Systems" (AICA)

- **Member of the following Technical IEEE Committees:** Fault-Tolerant Computing, Distributed Systems, Security and Privacy and VLSI Systems.
- **Editorial Board member of:**
  - Journal of Computer System Science and Engineering,
  - International Journal of Adaptive, Resilient and Autonomic Systems, IGI-global
  - Springer Verlag Series on Dependable Computing and Fault Tolerant Systems
- **Participation and responsibility roles in numerous International Projects** (Delta-4, PDCS1 and PDCS2, GUARDS, Cabernet, AMSD, Crutial, RESIST)

Internationally recognized research contributions on

- Dependable Computing
- Fault-Tolerant Systems
- Safety-Critical Systems
- Design Verification and Validation
- Design Environments and Tools

My experience

- Met Luca long time ago, when I was a student at the Computer Science Dep., Univ. of Pisa
- Co-advisor of my Master Thesis on the challenging problem of **Byzantine Agreement**
  - *Optimization of the Lamport's OM algorithm, in presence of low number of faults*
- Signed my future research activity, introducing me to the "fascinating" world of Fault Tolerant Systems !!!!
- A few papers published on the thesis work
- but when I joined IEI as a researcher in 89, Luca was already with the University and I started investigating the "Adjudication problem" in redundant structures

- Nevertheless, cooperation has been continuously in place in the context of research projects, as well as:
  - Personal advices in a number of working and relationship circumstances
  - Active encouragement when coping with challenging activities
  - Opportunities to grow in experience
  - …..

  For which I would like to express my hearth-felt **THANK** to Luca!!!!

Luca Simoncini Retirement Seminar: from Leslie to Luca

Briatico Ciuffoletti Simoncini 1984

A DISTRIBUTED DOMINO-EFFECT FREE RECOVERY ALGORITHM

D. Briatico*, A. Ciuffoletti**, L. Simoncini***

* ITALTEL, Castelletto di Settimo Milanese
** Dipartimento di Informatica, Università di Pisa
*** Istituto di Elaborazione dell'Informazione, CNR, Pisa

ABSTRACT

Recovery techniques may be distinguished on the basis of the time when the recovery lines are built:

at the time of recording the recovery point,
at the time of rollback.

Consequently we distinguish "planned" and "unplanned" policies for determining recovery lines.

With an unplanned policy a "domino effect" can occur.

The planned policy is usually intended as being static, in the sense that the recovery lines are a priori established at design time.

In this paper an algorithm for "dynamic" planning of recovery line is specified. We shall define a computational model for a distributed system of communicating processes using asynchronous message passing and shall describe the recovery algorithms by means of axioms.

1 INTRODUCTION

This paper is concerned with error recovery in distributed systems.

We model such system with a set of independent components, the processes, which exchange information through message passing.

With respect to the independence of the processes, we suppose that the message passing mechanism is asynchronous, as opposed to synchronous mechanisms as defined in ADA /ADA80a/ or CSP /HOA78a/ or DP /HAN78a/. Thus we allow that messages may be shuffled during the propagation between the sender and the receiver, and we consider that communications are made reliable by lower levels.

In a distributed systems, the operation of error recovery is aimed to the restoration of a consistent state after the failure of a subset of components of the system.

In this paper we are concerned with backward recovery and we give an algorithmic solution,applicable to distributed systems.

With backward methods, a failed process or a substitute of it can resume a previously saved state. We call this action rollback and the saved state recovery point.

Thus we can identify two distinct phases in a backward recovery algorithm: the first which is devoted to the creation of recovery points, and which takes place during the normal operation of the system; the second, the rollback, which is aimed to the resumption of a correct state through the use of the information contained in the recovery points.

It is well known that in distributed systems the backward error recovery algorithm may be hindered by the occurrence of the "domino effect". This one consists in the impossibility of identifying a recovery line using the recovery points available on the different components.

A recovery line is a consistent set of recovery points, from which computation may be resumed. Two different attitudes can be identified about the time when the system builds the recovery line, joining recovery points of the processes:

- at the time of the recording of the recovery point;
- at the time of the rollback.

The first attitude involves that recovery points are recorded according to certain rules that ensure that all of them belong to a recovery line. In this case we speak of a "planned" recording of recovery points /RAN75a/.

The second attitude involves that recovery points are recorded regardless to the generation of 'recovery lines: the generation of recovery lines is attempted at the time of the rollback.

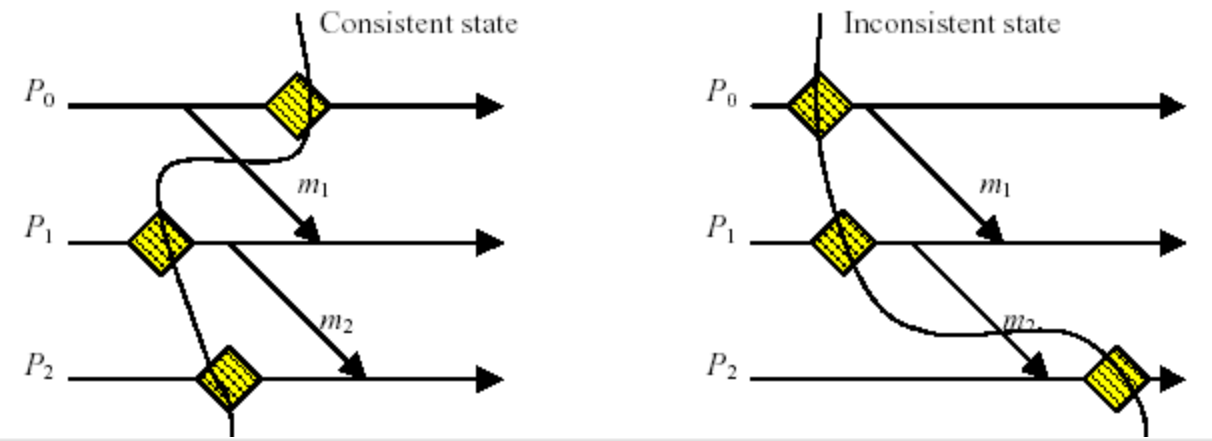
This attempt is not necessarily successful, as a domino effect can make useless part of the recorded recovery points /AND79a/. The probability and the amount of the domino effect is design dependent /RUS80a/, /MER78a/.

In this paper we shall consider the first attitude. It requires a coordination among the recovery points of the processes. This is directed by the

Daniele Briatico, Augusto Ciuffoletti, and Luca Simoncini
A Distributed Domino-effect Free Recovery Algorithm
In Proc. of Symposium on Reliability in Distributed Software and Database Systems
pp. 207-215, Silver Spring (Maryland), October 1984.

Consistent System States

- Intuitively, a consistent global state is one that may occur during a failure-free, correct execution of a distributed computation.



- The goal of a rollback-recovery protocol is to bring the system into a consistent state

BCS algorithm (1)

- upon the receive of message m(i)

  IF m.i>#i THEN

  #i=m.

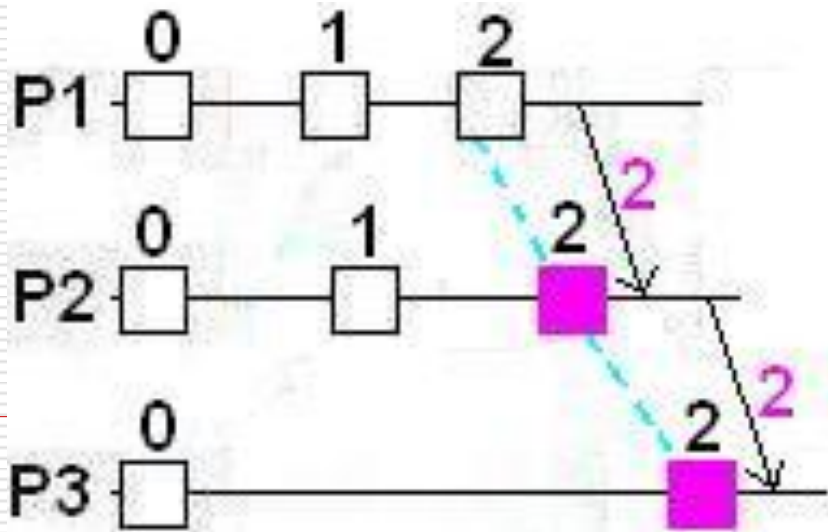
  Take_CKPT(#i)

  deliver(m)
- when sending message m

  send (m(#i))
- When Taking a local CKPT

  #i++

  Take_CKPT(#i)



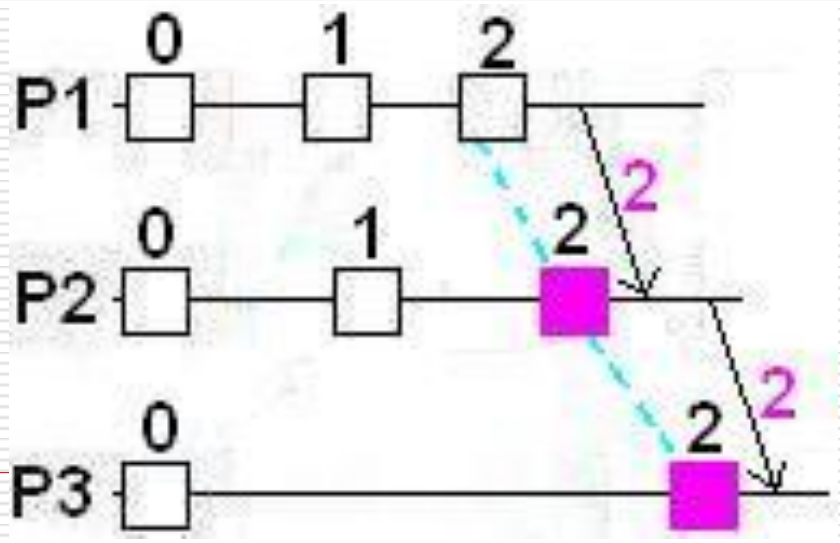
BCS algorithm (2)

- If we consider checkpoints are «relevant events» of a process,
- Then the monothonically increasing number that is labeling checkpoints is actually an instantiation of the Lamport logical clock [79]

- Important result
  - Cited in Totem Books

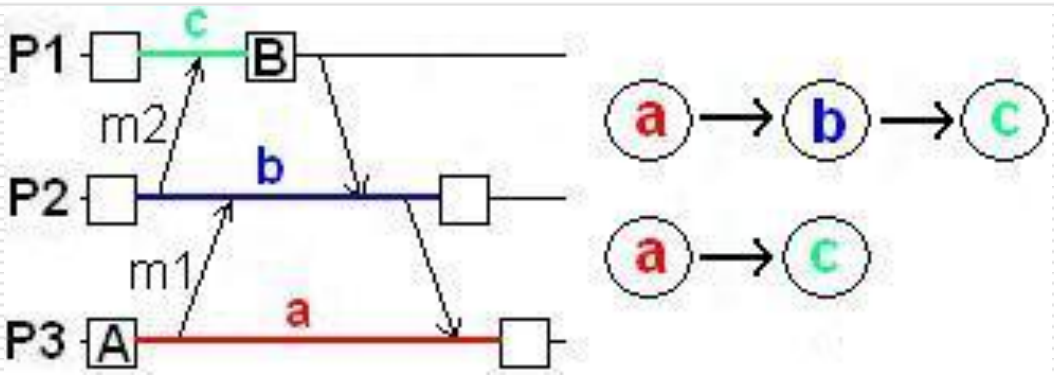
  like - Bernstein, Hadzilacos

  Goodman -
  - Many citations



BCS algorithm (3)

- Z-cycle, Z-path theory (BHR2001 Info&Compu)



- BCS became the first member of a specific class of checkpointing algorithms based on index
- Removing Z-cycle because the index is monothonically increasing like the Leslie's Clock

GRAZIE LUCA!!!