LADC 2011: Proposal for a tutorial

Michel Raynal

IRISA, Université de Rennes 1
Campus de Beaulieu, 35042 Rennes Cedex, France
raynal@irisa.fr

1 Title

Synchronization is coming back but is it the same?
(Modern synchronization: facing the net effect of asynchrony and failures)

2 Area and duration of the tutorial

Area: Theory and practice of dependability for shared memory distributed systems: the synchronization issues.
Duration : 4 hours.

3 Description of the topic

Informally, "wait-free" means that the progress of a process depends only on itself. This notion is more and more pervasive in a lot of problems that basically relie (in one way or another) on the definition and the use of concurrent objects in presence of failures. This tutorial will visit wait-free computing: its underlying concepts and its basic mechanisms. To that end, the lecture will also visit fundamental problems of asynchronous computing in presence of failures such as renaming, set agreement, collect, snapshot, etc. It will also present fundamental notions related to the implementation of concurrent objects, such as t-resilience and graceful degradation.

The literature on this topic is mostly technical and appears mainly in theoretically inclined journals and conferences. The aim of this tutorial is to offer an introductory survey to the new synchronization concepts that have been introduced in the recent past. The tutorial is destined to the people who are not familiar with these concepts and want to quickly understand their aim, their basic principles, their power and limitations. The tutorial will adopt an algorithmic approach to explain these new concepts. From a practical point of view, the advent of multicore architecture makes this topic central for researchers and engineers whose main interests lie in distributed fault-tolerance and dependability for shared memory systems. Moreover, whatever the problem they have to solve, one aim of the tutorial is to enlarge the knowledge and the background of researchers and engineers whose main interest is dependability.

4 Structure and technical content of the tutorial

Planned duration: 2h * 2 = 4 hours.

Supporting material: Transparencies and survey papers corresponding to parts of talk.

Content:

1. Introduction
   (a) Context: asynchronous shared memory distributed systems prone to failures
   (b) Where the problems are coming from?
2. On the safety side: the concept on linearizability
3. On the liveness side: To lock or not to lock, that is the question
   (a) The traditional lock-based approach
   (b) obstruction-freedom, non-blocking and wait-freedom
4. Examples of a few concurrent objects
   (a) Collect and Snapshot objects
   (b) Adaptive renaming and concurrent set objects
5. The fundamental problem
   (a) How to build any concurrent object (defined from a seq. spec.) despite failures
   (b) Consensus number and Herlihy's hierarchy
   (c) A universal construction
6. Implementing consensus
   (a) Using an appropriate synchronization primitive
   (b) Using timed registers or sticky bits
   (c) The failure detector-based approach
7. Boosting obstruction-freedom to non-blocking or wait-freedom
   (a) Assuming additional synchrony
   (b) The failure detector-based approach
8. The notion of t-resilient object
   (a) t-resilience vs obstruction/wait-freedom
   (b) t-resilient register object and t-resilient consensus object
9. When resilience is not enough: the notion of graceful degradation
   (a) Object failure modes: graceful degradation vs resilience
   (b) Gracefully degrading implementations
   (c) What is possible and what is not

5 The audience

This tutorial is aimed at:

- Practitioners who want to understand the basics of shared memory synchronization in presence of failure and related dependability problems.
- Master/PhD students involved in failure-prone distributed systems or parallel systems, who want to understand the fundamental concepts that have been recently stated and investigated, know their power and limitation, and be able to put them into practice.

The attendees will acquire a solid knowledge on synchronization in presence of failures: what can be done, and how it can be done. A fundamental statement is the following: "When a solution works, we should know why it works. When a solution does not work, we should know why it does not work". The tutorial will help people make this statement true, when they have to implement concurrent objects (i.e., shared services) in asynchronous shared memory systems prone to failures.

Last but not least, it is important to notice that the advent of multicore architectures makes synchronization in presence of failures more important than ever.

6 Biographical sketch

Michel Raynal has been a professor of computer science since 1981. At IRISA (CNRS-INRIA-University joint computing research laboratory located in Rennes), he founded a research group on Distributed Algorithms in 1983. His research interests include distributed algorithms, distributed computing systems, networks and dependability. His main interest lies in the fundamental principles that underly the design and the construction of distributed computing systems. He has been Principal Investigator of a number of research grants in these areas, and has been invited by many universities all over the world to give lectures on distributed algorithms and distributed computing.

Professor Michel Raynal has published more than 115 papers in journals (Journal of the ACM, Algorithmica, Acta Informatica, SIAM Jounrnal of Computing, Distributed Computing, Comm. of the ACM, Information and Computation, Journal of Computer and System Sciences, JPDC, IEEE Transactions on Computers, IEEE Transactions on SE, IEEE Transactions on KDE, IEEE Transactions on TPDS, IEEE Computer, IEEE Software, IPL, PPL, Theoretical Computer Science, Real-Time Systems Journal, The Computer Journal, etc.); and more than 250 papers in conferences (ACM STOC, ACM PODC, ACM SPAA, IEEE ICDCS, IEEE DSN, DISC, COCOON, IEEE IPDPS, Europar, FST&TCS, IEEE SRDS, etc.). (His h-index is 45.) He has also written nine books devoted to parallelism, distributed algorithms and systems (MIT Press and Wiley). His last two books "*Communication and Agreement Abstractions for Fault-Tolerant Asynchronous Distributed Systems*" (June 2010) and "*Fault-tolerant Agreement in Synchronous Message-passing Systems*" (September 2010) have been published by Morgan & Clapypool.

Professor Michel Raynal has been an invited speakerin more than 20 international conferences (including the prestigious DISC, Europar, ICDCN, OPODIS and NCA conferences). He belongs to the editorial board of several international journals (including JPDC, IEEE TC, IEEE TPDS, JCSSE and FDCS). He has served in program committees for more than 140 international conferences (including ACM PODC, DISC, ICDCS, IPDPS, DSN, LADC, SRDS, SIROCCO, etc.) and chaired the program committee of more than 15 international conferences (including DISC -twice-, ICDCS, OPODIS, SIROCCO and ISORC). He has also been general chair of several major conferences. Moreover, Michel Raynal served as the chair of the steering committee leading the

DISC symposium series in 2002-2004, and was a member of the steering committees of ACM PODC (ACM Symposium on the Principles of Distributed Computing) and SIROCCO (Colloquium on Structural Information and Communication Complexity). He is currently member of the steering committees of ICDCN (Int'l Conference on Distributed Computing and Networks) and IEEE ICDCS (Int'l Conference on Distributed Computing Systems). He is also the European representative in the IEEE technical committee on Distributed Computing.

Michel Raynal received the IEEE ICDCS "Best Paper" award three times in a row: 1999, 2000 and 2001. Recently, he also received the "Best Paper" award at the Int'l conference SSS 2009, and the "Distinghised Paper" award at EUROPAR 2010.

More information can be obtained at http://www.irisa.fr/prive/michel.raynal/ or, as far as publications are concerned, from DBLP, CiteSeer, or any other system. Michel Raynal is Senior member of the prestigious *Institut Universitaire de France* (IUF).