ACM SIGACT News Distributed Computing Column 13

Sergio Rajsbaum*

Abstract

The Distributed Computing Column covers the theory of systems that are composed of a number of interacting computing elements. These include problems of communication and networking, databases, distributed shared memory, multiprocessor architectures, operating systems, verification, Internet, and the Web.

This issue describes the PODC 20th anniversary Special Issue published by *Distributed Computing* journal.

Request for Collaborations: Please send me any suggestions for material I should be including in this column, including news and communications, open problems, and authors willing to write a guest column or to review an event related to theory of distributed computing.

The PODC 20th Anniversary Distributed Computing Special Issue

The 20th ACM Principles of Distributed Computing (PODC) conference was held in 2002. To celebrate this event, Hagit Attiya and myself edited a special issue of the journal *Distributed Computing,* with the support and encouragement of its editor, Vassos Hadzilacos. We invited contributions from the community, including surveys, retrospectives, new results, and personal perspectives. In particular we contacted all twenty past program committee chairs encouraging them to participate:

Michael Fischer 82	Nancy Lynch 83	Jayadev Misra 84	Ray Strong 85
Joseph Halpern 86	Fred B. Schneider 87	Dany Dolev 88	Michael Merritt 89
Cynthia Dwork 90	Richard Ladner 91	Maurice Herlihy 92	Sam Toueg 93
David Peleg 94	Vassos Hadzilacos 95	Yoram Moses 96	Hagit Attiya 97
Yehuda Afek 98	Jennifer Welch 99	James Anderson 00	Nir Shavit 01

We ended up with the following nine papers:

1. Nancy Lynch, “Some perspectives on PODC”
2. James H. Anderson, Yong-Jik Kim, Ted Herman, “Shared-memory mutual exclusion: major research trends since 1986”
3. Cyril Gavoille, David Peleg, “Compact and localized distributed data structures”
4. Faith Fich, Eric Ruppert, “Hundreds of impossibility results for distributed computing”
5. James Aspnes, “Randomized protocols for asynchronous consensus”

*Instituto de Matemáticas, UNAM. Ciudad Universitaria, Mexico City, D.F. 04510 rajsbaum@math.unam.mx.

6. Oded Goldreich, "Cryptography and cryptographic protocols"

7. R. Kurki-Suonio, "Action systems in incremental and aspect-oriented modeling"

8. Leslie Lamport, "Arbitration-free synchronization"

9. Michael J. Fischer, Michael Merritt, "Appraising two decades of distributed computing theory research"

The following is a reprint of the introduction to the Special Issue by Hagit Attiya and myself.

* * *

This issue celebrates twenty years of the Annual ACM Symposium on Principles of Distributed Computing (PODC). First held in 1982, PODC has been dedicated to promoting research in the foundations of distributed computing. The twenty five or so papers presented each year in PODC have been instrumental in creating a diverse collection of principles for designing distributed systems.

Over the years many particular problems have been studied that are of interest to databases, operating systems, computer architecture, networking and the Internet. Many variants of message passing and shared-memory models of computation have been used to design algorithms and explore the cost of these problems in terms of resources such as messages, time, size of shared objects, congestion, and local computation. Lower bounds and impossibility results have explored the inherent limitations of distributed computation. Fault-tolerance has been a central concern; many failures types have been considered, and techniques such as wait-freedom, self-stabilization, failure-detection and security have been developed. Topics in formal methods such as specification, verification, modeling and semantics have appeared since the first PODC, although not as frequently today.

To mark this anniversary, we collected papers that reflect, survey, and criticize some of the major developments in this area. The resulting issue, which you now read, covers a variety of problem domains, of computing models, of applications and approaches. Nevertheless, the research featured in this special issue is merely a sampler—covering all the interesting and influential research carried out within PODC is simply impossible.

The issue starts with a personal account of the history of PODC and its future directions, by Nancy Lynch [1]. Lynch recalls how the conferences originated, and how it has evolved over time. She reflects on the role of formal methods and the future of distributed computing theory.

The *mutual exclusion* problem captures the essence of coordinating access to a shared resource; it comes up in numerous situations, ranging from concurrent programming to operating systems. Mutual exclusion received theoretical treatment even prior to the first PODC conference and it is still the focus of much research. James Anderson, Yong-Jik Kim and Ted Herman [2] survey research achievements in this area, emphasizing the impact of architectural assumptions on algorithm design.

Network algorithms is another subject that had been studied even before PODC was established, and is still central to it. In particular, papers on routing appear regularly since the very first PODC. Cyril Gavoille and David Peleg [3] discuss *distributed data structures* for this and other applications, illustrating a characteristic trait of PODC: the relationship between the underlying communication network (and its graph theoretic properties) and efficient algorithmic and data placement schemes.

Curiosity for understanding the inherent limitations of distributed computing has been paramount in the foundation of PODC. It is not surprising, therefore, that some of the deepest theoretical results in PODC have to do with *impossibility results*. An impossibility result tells us that a certain problem cannot be solved in a particular system architecture or that it requires some amount of resources. It can indicate that we should either weaken the problem specification or strengthen the assumptions about the environment in which it should be solved. Faith Fich and Eric Ruppert [4] survey key techniques and ideas that have been used to prove hundreds of impossibility results for distributed computing.

The *consensus* problem requires processes to agree on a common value, in a non-trivial manner. This fundamental problem abstracts key aspects of coordination needed in both shared-memory and message passing systems, in a variety of problem areas such as middleware support, database systems, concurrent objects, and broadcasting. It is in some sense a universal problem in that many other problems can be reduced to it. Synchronous consensus algorithms were presented already in the first PODC, but it did not take long to discover what is perhaps the most influential impossibility result in distributed computing: in an asynchronous environment, consensus cannot be solved in the presence of even a single failure. James Aspnes [5] writes about one of the techniques developed for dealing with the impossibility of solving consensus, namely, to provide only probabilistic termination guarantees.

Distributed computing inherently brings with it questions of security, and cryptography has always been synergetic to PODC: Cryptographic protocols employ distributed computing techniques and distributed systems making use of cryptographic mechanisms. Oded Goldreich [6] describes the cryptographic principles underlying the design of cryptographic protocols and presents the state-of-art in an area that is of particular importance in the world today.

The need for formal methods was another driving force in the creation of PODC: many problems and algorithms were previously described somewhat informally, and it was clear early on that it is hard to reason about and verify the correctness of distributed systems. The second PODC included a paper by Back and Kurki-Suonio about modeling distributed systems, one out of many dedicated to this important area. Twenty years later, Reino Kurki-Suonio [7] returns to this work and describes the use of action systems in incremental and aspect-oriented modeling.

Research in concurrency and multiprocess synchronization dates back to the 60's. Concurrent reading and writing of a single register requires an arbiter—a device for making a binary decision based on changing inputs. Computer designers have known since the 50s that it is impossible to build an arbiter that is guaranteed to reach a decision within bounded time. It is therefore interesting to know what is the power of *arbiter-free computation*. Leslie Lamport [8] investigates the rudiments of synchronization, and shows a class of problems—including the producer/consumer problem—that can be solved in the absence of arbitration, in a 25-years old result that was not published before.

The jubilee issue ends with a critique by Michael Fischer and Michael Merritt [9], who reflect on two decades of distributed computing theory research, and chart directions in which this research should evolve.

Producing this issue demanded a great investment of time and effort from many people. First and foremost, the authors who contributed papers for the jubilee issue, and went through several rounds of reviewing and revision. Each contribution was reviewed by three or four Reviewers, Who provided helpful, often insightful, comments and suggestions for improving the papers. Finally, the editor-in-chief, Vassos Hadzilacos worked with us hand-in-hand throughout this endeavor.

References

[1] Nancy Lynch, "Some perspectives on PODC," *Distributed Computing* Volume 16, Numbers 2-3, September 2003, Pages: 71–74.

[2] James H. Anderson, Yong-Jik Kim, Ted Herman, "Shared-memory mutual exclusion: major research trends since 1986," *Distributed Computing* Volume 16, Numbers 2-3, September 2003, pp. 75–110.

[3] Cyril Gavoille, David Peleg, "Compact and localized distributed data structures," *Distributed Computing* Volume 16, Numbers 2-3, September 2003, pp. 111–120.

[4] Faith Fich, Eric Ruppert, "Hundreds of impossibility results for distributed computing," *Distributed Computing* Volume 16, Numbers 2-3, September 2003, pp. 121–163.

[5] James Aspnes, "Randomized protocols for asynchronous consensus," *Distributed Computing* Volume 16, Numbers 2-3, September 2003, pp. 165–175.

[6] Oded Goldreich, "Cryptography and cryptographic protocols," *Distributed Computing* Volume 16, Numbers 2-3, September 2003, pp. 177–199.

[7] R. Kurki-Suonio, "Action systems in incremental and aspect-oriented modeling," *Distributed Computing* Volume 16, Numbers 2-3, September 2003, pp. 201–217.

[8] Leslie Lamport, "Arbitration-free synchronization," *Distributed Computing* Volume 16, Numbers 2-3, September 2003, pp. 219–237.

[9] Michael J. Fischer, Michael Merritt, "Appraising two decades of distributed computing theory research," *Distributed Computing* Volume 16, Numbers 2-3, September 2003, pp. 239–247.