Marina Papatriantafilou
Philippe Hunel (Eds.)

LNCS 3144

Principles of Distributed Systems

7th International Conference, OPODIS 2003
La Martinique, French West Indies, December 2003
Revised Selected Papers

Opodis

Springer

Marina Papatriantafilou Philippe Hunel (Eds.)

Distributed Systems

7th International Conference, OPODIS 2003
La Martinique, French West Indies, December 10-13, 2003
Revised Selected Papers

Springer

Volume Editors

Marina Papatriantafilou
Chalmers University of Technology, Department of Computing Science
S-412 96 Gothenburg, Sweden
E-mail: ptrianta@cs.chalmers.se

Philippe Hunel
Université des Antilles et de la Guyane, Campus de Schoelcher
GRIMAAG, Département Scientifique Inter-Facultés
BP 7109, 97275 Schoelcher CEDEX, Martinique, France
E-mail: Philippe.Hunel@martinique.univ-ag.fr

Library of Congress Control Number: 2004109778

CR Subject Classification (1998): C.2.4, D.1.3, D.2.7, D.2.12, D.4.7, C.3

ISSN 0302-9743
ISBN 3-540-22667-2 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11307518 06/3142 5 4 3 2 1 0

Preface

The 7th International Conference on Principles of Distributed Systems (OPODIS 2003) was held during December 10–13, 2003 at La Martinique, French West Indies, and was co-organized by the Université des Antilles et de la Guyane, La Martinique, French West Indies and by Chalmers University of Technology, Sweden. It continued a tradition of successful conferences with friendly and pleasant atmospheres. The earlier organizations of OPODIS were held in Luzarches (1997), Amiens (1998), Hanoi (1999), Paris (2000), Mexico (2001) and Reims (2002).

OPODIS is an open forum for the exchange of state-of-the-art knowledge on distributed computing and systems among researchers from around the world. Following the tradition of the previous organizations, its program is composed of high-quality contributed and invited papers by experts of international caliber in this scientific area. The topics of interest are theory, specifications, design and implementation of distributed systems, including distributed and multiprocessor algorithms; communication and synchronization protocols; coordination and consistency protocols; stabilization, reliability and fault-tolerance of distributed systems; performance analysis of distributed algorithms and systems; specification and verification of distributed systems; security issues in distributed computing and systems; and applications of distributed computing, such as embedded distributed systems, real-time distributed systems, distributed collaborative environments, peer-to-peer systems, cluster and grid computing.

In response to the call for papers for OPODIS 2003, 61 papers in these areas were submitted. The Program Committee, following a peer-review process, selected 19 out of these for presentation at the conference. Each paper, reviewed by at least 4 reviewers, was judged according to scientific and presentation quality, originality and relevance to the conference topics. The distribution of the accepted (respectively, submitted) papers per geographic region was: Asia–Australia, 3 papers accepted (out of 14 submitted); Europe, 11 papers accepted (out of 34 submitted); Central and North America, 5 papers accepted (out of 13 submitted).

Besides the technical contributed papers, the program included invited keynote talks. We were happy that three distinguished experts accepted our invitation to share with us their views of various aspects of the field: Jo Ebergen (Sun Microsystems Laboratories), who gave the luncheon speech on circuits without clocks, Neil Gershenfeld (MIT Center for Bits and Atoms), who talked about physical error correction in building reliable systems out of unreliable components, and Maarten van Steen (Vrije Universiteit Amsterdam), who talked about very large, self-managing distributed systems. Abstracts of the contents of the keynote talks are included in this volume.

Apart from the technical program, OPODIS 2003 also offered a set of satellite events in the form of tutorials, with the themes: Self-stabilization, by

Joffroy Beauquier (Université de Paris 11); Distributed Computing and Information Security, by Roberto Gomez Cárdenas (ITESM-CEM); and Non-blocking Synchronization, by Philippas Tsigas (Chalmers University of Technology).

It is impossible to organize a successful program without the help of many individuals. We would like to express our appreciation to the authors of the submitted papers, and to the program committee members and external referees, who provided useful reviews. Furthermore, we would also like to thank the OPODIS steering committee members, who supervise and support the continuation of the event. We owe special thanks to Yi Zhang for his assistance with the electronic submissions and reviewing system. Finally, one more special thanks to all the other organizing committee members for their precious efforts that contributed to making OPODIS 2003 a successful conference.

Marina Papatriantafilou
Philippe Hunel
OPODIS 2003 Program Co-chairs

Program Committee

Mustaque Ahamad	Georgia Inst. of Technology, USA
Joffroy Beauquier	Univ. Paris 11, France
Alain Bui	Univ. Reims Champagne-Ardenne, France
Marc Bui	Univ. Paris 8, France
Osvaldo Carvalho	Univ. Fed. Minas Gerais, Brazil
Bernadette Charron-Bost	Lab. d'Informatique, LIX, France
Hacene Fouchal	Univ. Reims Champagne-Ardenne, France
Roberto Gomez-Cardenas	CEM-ITESM, Mexico
Hans Hansson	Mälardalen Univ., Sweden
Ted Herman	Univ. of Iowa, USA
Teruo Higashino	Osaka Univ., Japan
Philippe Hunel (Co-chair)	Univ. of Antilles-Guyane, French West Indies
Colette Johnen	LRI, CNRS-Univ. Paris-Sud, France
Christian Lavault	LIPN, CNRS UMR 7030, Univ. Paris 13, France
Toshimitsu Masuzawa	Osaka Univ., Japan
Jean-Franois Mehaut	Univ. of Antilles-Guyane, French West Indies
Dominique Mery	Univ. Henri Poincaré and LORIA, France
Marina Papatriantafilou (Co-chair)	Chalmers Univ. of Technology, Sweden
Luis Rodrigues	Univ. of Lisbon, Portugal
Nicola Santoro	Carleton Univ., Canada
Alex Shvartsman	Univ. of Connecticut/MIT, USA
Philippas Tsigas	Chalmers Univ. of Technology, Sweden
Vincent Villain	Univ. Picardie Jules Verne, France

Organization

OPODIS 2003 was organized by the Université des Antilles et de la Guyane, La Martinique, French West Indies and by Chalmers University of Technology, Gothenburg, Sweden.

Organizing Institutes

L'Université des Antilles et de la Guyane

CHALMERS
Chalmers University of Technology, Sweden

Organizing Committee

Hacene Fouchal	Univ. Reims Champagne-Ardenne, France
Philippe Hunel	Univ. of Antilles-Guyane, La Martinique, French West Indies
Richard Nock	Univ. of Antilles-Guyane, La Martinique, French West Indies
Marina Papatriantafilou	Chalmers Univ. of Technology, Sweden
Jean-Emile Symphor	Univ. of Antilles-Guyane, La Martinique, French West Indies
Yi Zhang	Chalmers Univ. of Technology, Sweden

Steering Committee

During 2003 the Steering Committee of OPODIS consisted of:

Alain Bui	Univ. Reims Champagne-Ardenne, France
Marc Bui	Univ. Paris 8, France
Roberto Gomez-Cardenas	CEM-ITESM, Mexico
Philippas Tsigas	Chalmers Univ. of Technology, Sweden
Vincent Villain	Univ. Picardie Jules Verne, France

Other Supporting and Sponsoring Organizations

The conference was supported and sponsored by the Université des Antilles et de la Guyane, Chalmers University of Technology, Springer-Verlag (publication of this official, postconference proceedings volume), Canon Martinique (preliminary proceedings volume, available during the conference), the Research Ministry of France, the Research Council of Sweden, the French National Centre for Scientific Research (CNRS, GdR Architecture, Réseaux et systèmes, Parallélisme), Microsoft Research, the Department of Tourism in Martinique (Office Départemental du Tourisme de la Martinique, ODTM), the Regional Agency for the Touristic Development of Martinique (Agence Régionale pour le Développement Touristique de la Martinique, ARDTM), the Municipality of Schoelcher, and the ACM French Chapter.

The electronic submission and reviewing system used for OPODIS 2003 was the CyberChair system, authored by Richard van de Stadt.

Referees

Ahmed Ainouche
Johan Andersson
Filipe Araújo
Anish Arora
Hichem Baala
Alina Bejan
Simon Bloch
Olivier Bournez
Jean-Michel Bruel
Franck Butelle
Ken Calvert
Antonio Casimiro
Pranay Chaudhuri
Nawal Cherfi
Bruno Codenotti
Alain Cournier
Ivica Crnkovic
Sivarama Dandamudi
Xavier Defago
Carole Delporte
Stéphane Devismes
Andreas Ermedahl
Hugues Fauconnier
Olivier Festor
German Finez
Lucian Finta
Olivier Flauzac
Pierre Fraigniaud
Johan Fredriksson
Eduardo Garcia
Philippe Gauron
Chryssis Georgiou
Sukumar Ghosh
Anders Gidenstam
Jens Gustedt
Phuong-Hoai Ha
Thomas Herault
Lisa Higham
Taisuke Izumi
Raul Jacinto
Mehmet-Hakan Karaata
Boris Koldehofe
Kishori Konwar
Michael Krajecki
Mikel Larrea-Alava
Victor M. Larios-Rosillo
Patrice Laurencot
Fabrice Le Fessant
Pierre Lemarinier
Erika Mata-Sanchez
Stephan Merz
Hugo Miranda
Lynda Mokdad
Peter Musial
Anders Möller
Yoshihiro Nakaminami
Mikhail Nesterenko
Richard Nock
Mikael Nolin
Florent Nolot
Thomas Nolte
Rui Oliveira
Fukuhito Ooshita
Gabriel Paillard
Catuscia Palamidessi
José-Orlando Pereira
Paul Pettersson
Scott M. Pike
Laurence Pilard
Imran Pirwani
Stefan Pleisch
Vahid Ramezani
Sylvain Rampaceck
Xavier Rebeuf
Antoine Rollet
Launrent Rosaz
Brigitte Rozoy
Sebastien Salva
Kristian Sandström
Pierre Sens
Devan Sohier
Olivier Soyez
Gerard Tel
Henrik Thane
Sébastien Tixeuil
Luis Trejo
Tatsuhiro Tsuchiya
Hasan Ural

Peter Urban
Thierry Val
Edgar Vallejo
Krishnamurthy Vidyasankar
Ramesh Viswanath
Anders Wall
Josef Widden
Mark Wineberg
Wang Yi
Chen Zhang
Hongwei Zhang
Mikael Åkerholm

Author Index

Table of Contents

Invited Talk 1

Invited Talk 2

Invited Talk 3

Distributed and Multiprocessor Algorithms I

Peer-to-Peer Systems, Middleware I

Peer-to-Peer Systems, Middleware II

Real-Time and Embedded Systems

Verification, Models, Performance of Distributed Systems

Distributed and Multiprocessor Algorithms II

Distributing Bits and Atoms

Neil Gershenfeld

Director
MIT Center for Bits and Atoms
Cambridge, MA 02139, USA

Abstract. The principles used today for developing distributed systems will not scale to the limit of thermodynamically complex engineered systems. The great insight of statistical mechanics is that it is possible to make precise statements about the macroscopic behavior of a system based on knowledge of its microscopic governing equations, without requiring a specification of its internal confirguration. A scalable theory of distributed system design must likewise be able to allocate available local degrees of freedom to accomplish a global goal, without demanding a detailed description of their configuration. Towards that end, I discuss the role of physical error correction in building reliable systems out of unreliable components, and the use of principles from mathematical programming as a language for expressing algorithms in this statistical-mechanical limit.

M. Papatriantafilou and P. Hunel (Eds.): OPODIS 2003, LNCS 3144, p. 1, 2004.

Circuits Without Clocks: What Makes Them Tick?

Jo Ebergen

Asynchronous Design Group
Sun Microsystems Laboratories
Mountain View CA 94043, USA
jo.ebergen@sun.com
http://research.sun.com/projects/async/

Abstract. Most digital circuits have a global clock that dictates when all circuit components execute their basic computation steps. The clock is a convenience for the designer, because the clock synchronizes all basic computations to its ticks. On the other hand, the clock can be a serious inconvenience with respect to speed, power consumption, modularity of design, and reduced electro-magnetic radiation. A clockless circuit is essentially a distributed system in-the-small, where the main challenge is the coordination of all basic computations in a fast and energy-efficient manner. A growing research community is exploring the benefits of circuits without clocks. In this talk I will give a brief overview of clockless circuits, illustrate their potential by means of some 'live' demos, and discuss current challenges.

M. Papatriantafilou and P. Hunel (Eds.): OPODIS 2003, LNCS 3144, p. 2, 2004.

Towards Very Large, Self-Managing Distributed Systems
Extended Abstract

Maarten van Steen

Vrije Universiteit Amsterdam

1 Introduction

As distributed systems tend to grow in the number of components and in their geographical dispersion, deployment and management are increasingly becoming problematic. For long, there has been a tradition of developing architectures for managing networked and distributed systems [2]. These architectures tend to be complex, unwieldy, and indeed, difficult to manage. We need to explore alternative avenues if we want to construct a next generation of distributed systems.

Recently, solutions have been sought to develop self-managing systems. The basic idea here, is that a distributed system can continuously monitor its own behavior and take corrective action when needed. As with many new, or newly introduced, concepts, it is often difficult to separate hype from real content. In the case of self-management (or other forms of *self-∗-ness*), the low signal-to-noise ratio can be partly explained by our poor understanding of what self-management actually means.

2 A Self-Managing User-Centric CDN

In our own research on large-scale distributed systems at the Vrije Universiteit Amsterdam, we have been somewhat avoiding the problem of systems management. However, one of the lessons we learned from building Globe [8], is that supporting easy deployment and management is essential. Partly based on our experience with Globe, we are currently developing a user-centric Content Delivery Network to further explore facilities for self-management. This CDN, called Globule, is designed to handle millions of users, each providing Web content by means of a specially configured Apache Web server.

An important aspect of Globule is that a server can automatically replicate its Web documents to other servers. For each document, a server evaluates several replication strategies, and selects the best one on a per-document basis. This approach allows for near-optimal performance in terms of client-perceived delays as well as total consumed bandwidth [6]. In a recent study, we have also demonstrated that continuous re-evaluation of selected strategies is needed, and that this can be done efficiently [7]. The approach we follow is to regularly perform trace-driven simulations for a specific document, where each simulation entails a single replication strategy. Using a linear cost function defined over performance metrics such as client-perceived latency and consumed bandwidth, we can then compare the effects of applying different strategies. These simulations take in the order of tens of milliseconds in order to select the best strategy for a given document.

M. Papatriantafilou and P. Hunel (Eds.): OPODIS 2003, LNCS 3144, pp. 3–6, 2004.

Clearly, Globule should be able to manage itself when it comes to replicating documents, and as far as static content is concerned, such self-management appears to be feasible.

However, much more is needed to develop a CDN such as Globule. For one thing, if we are to replicate documents to where they are needed, it is mandatory that we can locate clients and replica servers in the proximity of those clients. One problem that needs to be solved is letting a Web server determine how close two arbitrary nodes in the system actually are. Fortunately, it turns out that if we consider latency as a distance metric, we can represent the nodes of a widely dispersed distributed system in an N-dimensional Euclidean space [4,5]. In this way, estimating latency is nothing more than a simple computation. In contrast to existing systems, latency estimations in Globule can be obtained in a fully decentralized manner, which, in turn, simplifies overall system management.

By introducing locations and easy-to-compute distances, it becomes feasible to automatically partition the set of nodes comprising a distributed system into manageable parts. For example, by grouping nodes into geographical zones (where the geography is fully determined by the Euclidean space mentioned before), we can assign special nodes to zones in order to manage services, resources, etc. These special nodes, called brokers in Globule, are elected as *super peers* from all available nodes, and together form a separate overlay network using their own routing protocol. Whenever a node in zone A requires services from a zone B (such as, for example, a list of potential replica servers) it simply sends a request to a broker in A which will then forward the request to a broker for B. In Globule, a zone is defined implicitly: it consists of the servers that are closest to a given broker. As a consequence, zones do not overlap. Moreover, adding and removing servers, be they brokers or not, is fully decentralized.

There are many variations on this theme, but it should be clear that grouping nodes into zones and electing brokers for zones are things that can be done in a fully decentralized fashion. There is no need for manual intervention, although there are many unresolved details concerning how this organization can be automatically done.

3 Epidemic-Based Solutions

One could argue that the description of a self-managing system given so far is largely dictated by automating tasks that are currently handled manually. In this sense, self-management is just a next step in the evolution of distributed systems. The question comes to mind if there are radically different alternatives. We are currently exploring epidemic-based systems for management tasks.

In an epidemic-based system, we are generally concerned with reaching eventual consistency: in the absence of any further updates, all nodes should eventually reach the same state. Data are spread by letting each node regularly contact an arbitrary other node, after which the two exchange updates [1]. The problem with this approach for very large systems, is that, in principle, every node should know the entire set of nodes in order to guarantee random selection of a peer. One solution is to maintain, per node, a small list of peers that represents a random sample from all nodes. Maintaining this list is now the key to successfully applying epidemics in very large systems.