Preface to the Second Edition

Distributed systems have witnessed phenomenal growth in the past few years. The declining cost of hardware, the advancements in communication technology, the explosive growth of the Internet and our ever-increasing dependence on networks for a wide range of applications ranging from social communication to financial transactions have contributed to this growth. The breakthroughs in embedded systems, nanotechnology, and wireless communication have opened up new frontiers of applications like sensor networks and wearable computers. The rapid growth of cloud computing and the growing importance of big data have changed the landscape of distributed computing.

Most applications in distributed computing center around a set of core subproblems. A proper understanding of these subproblems requires a background of the underlying theory and algorithmic issues. This book provides a balanced coverage of the foundational topics, and their relationship to real-life applications. The language has been kept as unobfuscated as possible – clarity has been given priority over formalism. The second edition fixes many of the problems in the first edition, adds new topics, and significantly upgrades the contents. The twenty-one chapters have been divided into five sections:

Section I (Chapters 1-2) deals with *background materials* that include various cloud computing platforms.
Section II (Chapter 3-6) presents *foundational topics*, which address system models, correctness criteria, and proof techniques.
Section III (Chapter 7-11) presents the core paradigm*s* in distributed systems – these include logical clocks, distributed snapshots and debugging, deadlock and termination detection, election, and distributed graph algorithms.
Section IV (Chapters 12-17) addresses failures and fault-tolerance techniques in various applications – it covers consensus, transactions, group communication, replicated data management, and self-stabilization. Group communication and consensus have been included in this section since they are two of the primary beneficiaries of fault-tolerant designs. Finally,
Section V (Chapters 18-21) addresses a few real-world issues: these include distributed discrete-event simulation, security, sensor networks, social and peer-to-peer networks. Each chapter has a list of exercises that will challenge the readers (those tagged with * are the more challenging ones). A small number of these are programming exercises. Some exercises will encourage the readers to learn about outside materials.

The book is intended for use in a one-semester course at the senior undergraduate or the first-year graduate level. About 75% of the materials can be covered in one semester. Accordingly, the chapters can be picked and packaged in several different ways. A theory oriented offering is possible using Chapters 1-17. For a more practical flavor, use chapters 1-2, selected topics from chapters 3-16, and chapters 18-21, supplemented by a semester-long project chosen from replicated data management, sensor networks, group communication, discrete-event simulation, social or P2P networks. Here is a disclaimer: this book is *not* about programming distributed systems. Chapter 2 is only a high level description that we expect everyone to know, but is *not* an introduction to programming. If programming is the goal, then I encourage readers to look for other materials. There are several good books available for this.

Iowa City
December 4, 2013

Sukumar Ghosh

Table of Contents

SECTION I. BACKGROUND MATERIALS

SECTION III. IMPORTANT PARADIGMS

SECTION IV. FAULTS AND FAULT-TOLERANT SYSTEMS

SECTION V. REAL WORLD ISSUES