

# Distributed systems

for fun and profit

[Previous Chapter](#) | [Home](#) | [Next Chapter](#)

## 6. Further reading and appendix

If you've made it this far, thank you.

If you liked the book, follow me on [Github](#) (or [Twitter](#)). I love seeing that I've had some kind of positive impact. "Create more value than you capture" and all that.

Many many thanks to: logpath, alexras, globalcitizen, graue, frankshearar, roryokane, jpfuentes2, eerror, cmeiklejohn, stevenproctor eos2102 and steveloughran for their help! Of course, any mistakes and omissions that remain are my fault!

It's worth noting that my chapter on eventual consistency is fairly Berkeley-centric; I'd like to change that. I've also skipped one prominent use case for time: consistent snapshots. There are also a couple of topics which I should expand on: namely, an explicit discussion of safety and liveness properties and a more detailed discussion of consistent hashing. However, I'm off to [Strange Loop 2013](#), so whatever.

If this book had a chapter 6, it would probably be about the ways in which one can make use of and deal with large amounts of data. It seems that the most common type of "big data" computation is one in which [a large dataset is passed through a single simple program](#). I'm not sure what the subsequent chapters would be (perhaps high performance computing, given that the current focus has been on feasibility), but I'll probably know in a couple of years.

# Books about distributed systems

## Distributed Algorithms (Lynch)

This is probably the most frequently recommended book on distributed algorithms. I'd also recommend it, but with a caveat. It is very comprehensive, but written for a graduate student audience, so you'll spend a lot of time reading about synchronous systems and shared memory algorithms before getting to things that are most interesting to a practitioner.

## Introduction to Reliable and Secure Distributed Programming (Cachin, Guerraoui & Rodrigues)

For a practitioner, this is a fun one. It's short and full of actual algorithm implementations.

## Replication: Theory and Practice

If you're interested in replication, this book is amazing. The chapter on replication is largely based on a synthesis of the interesting parts of this book plus more recent readings.

## Distributed Systems: An Algorithmic Approach (Ghosh)

## Introduction to Distributed Algorithms (Tel)

## Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery (Weikum & Vossen)

This book is on traditional transactional information systems, e.g. local RDBMS's. There are two chapters on distributed transactions at the end, but the focus of the book is on transaction processing.

## Transaction Processing: Concepts and Techniques by Gray and Reuter

A classic. I find that Weikum & Vossen is more up to date.

# Seminal papers

Each year, the [Edsger W. Dijkstra Prize in Distributed Computing](#) is given to outstanding papers on the principles of distributed computing. Check out the link for the full list, which includes classics such as:

- ["Time, Clocks and Ordering of Events in a Distributed System"](#) - Leslie Lamport
- ["Impossibility of Distributed Consensus With One Faulty Process"](#) - Fisher, Lynch, Patterson
- ["Unreliable failure detectors and reliable distributed systems"](#) - Chandra and Toueg

Microsoft Academic Search has a list of [top publications in distributed & parallel computing ordered by number of citations](#) - this may be an interesting list to skim for more classics.

Here are some additional lists of recommended papers:

- [Nancy Lynch's recommended reading list](#) from her course on Distributed systems.
- [NoSQL Summer paper list](#) - a curated list of papers related to this buzzword.
- [A Quora question on seminal papers in distributed systems.](#)

## Systems

- [The Google File System](#) - Ghemawat, Gobioff and Leung
- [MapReduce: Simplified Data Processing on Large Clusters](#) - Dean and Ghemawat
- [Dynamo: Amazon's Highly Available Key-value Store](#) - DeCandia et al.
- [Bigtable: A Distributed Storage System for Structured Data](#) - Chang et al.
- [The Chubby Lock Service for Loosely-Coupled Distributed Systems](#) - Burrows
- [ZooKeeper: Wait-free coordination for Internet-scale systems](#)

[Previous Chapter](#) | [Home](#) | [Next Chapter](#)

"Distributed systems: for fun and profit" by Mikito Takada.

0 Comments

Distributed Systems for fun and profit

 Login

Sort by Best

Share  Favorite 



Start the discussion...

Be the first to comment.



Subscribe



Add Disqus to your site



Privacy