

# 分布式算法入门讲义

黄 宇

2021 年 5 月 29 日

# 目录

<b>第一部分 基础知识</b>	<b>4</b>
<b>第一章 引言</b>	<b>5</b>
<b>第二章 分布式计算模型</b>	<b>6</b>
2.1 系统建模 . . . . .	6
2.1.1 计算模型的基本概念 — 一个二维框架 . . . . .	6
2.1.2 模型间模拟的概念 . . . . .	6
2.1.3 失效模型的基本概念 . . . . .	6
2.2 特定领域分布式系统的建模 . . . . .	6
2.2.1 分布式数据库系统 . . . . .	6
2.2.2 分布式消息分发中间件 . . . . .	7
2.3 问题规约 . . . . .	7
2.3.1 问题间的归约 . . . . .	7
2.4 消息传递模型 . . . . .	7
2.4.1 同步 . . . . .	7
2.4.2 异步 . . . . .	7
2.5 共享内存模型 . . . . .	7
<b>第二部分 消息传递</b>	<b>8</b>
<b>第三章 分布共识问题及基本算法</b>	<b>9</b>
<b>第四章 Paxos</b>	<b>10</b>
<b>第五章 Raft</b>	<b>11</b>
<b>第六章 原子性广播协议Zab</b>	<b>12</b>
<b>第七章 分布共享内存</b>	<b>13</b>
7.1 原子寄存器 . . . . .	13

目录	2
7.1.1 单写 . . . . .	13
7.1.2 多写 . . . . .	13
第八章 分布事务的原子性提交	14
第三部分 共享内存	15
第九章 互斥	16
第四部分 理论专题	17
第十章 不可区分性的构造	18
第五部分 系统专题	19
第十一章 Paxos协议的系统实现	20
第十二章 Raft协议的系统实现	21
第十三章 容错与可靠性	22
13.1 理解failure . . . . .	22
13.2 发现failure . . . . .	22
13.3 容忍failure . . . . .	22
第六部分 验证专题	23
第十四章 分布式协议的模型检验	24
第十五章 分布式系统实现的模型检验	25
第十六章 分布式协议的定理证明	26
第十七章 分布式系统实现的定理证明	27

# 前言

每年招新学生的时候，分布式算法知识的入门是一个反复出现的任务。一般而言，分布式算法知识和大家的本科课程距离较远，难以自然地衔接。分布式算法的经典课本、论文在本科毕业的时候直接阅读往往比较困难。另外一方面，网上相关技术内容不少，但是质量堪忧，有误导性的“负作用”。为此陆续整理这一本讲义，希望能将每年重复给新学生罗列的学习内容沉淀下来，复用起来。

这本讲义完全是“on demand”式地逐步添加的，而且论述非常简要，更像是向学生讲解所需阅读的入门文献时，所做的注解。未来有一天，如果这本讲义的内容充实到一定程度，或许它能成为一本合格的《分布式算法入门》课本。

# 第一部分

## 基础知识

# 第一章 引言

分布式系统的基础知识。

🍃 分布式系统的经典教科书 [TS06]。

形式化验证的基础知识

🍃 数理逻辑与软件形式化验证基础知识 [HR04]。

TLA+

🍃 TLA+的基础知识 <sup>1</sup>。

🍃 TLA+教程 [Hil18]。

分布式系统的形式化验证的工业界实践。

🍃 Amazon的实践 [NRZ<sup>+</sup>15]。

面向教学的分布式算法快速实现。

🍃 DSLabs [MWA<sup>+</sup>19]。

---

<sup>1</sup>Leslie Lamport, The TLA+ Home Page: <http://lamport.azurewebsites.net/tla/tla.html>

## 第二章 分布式计算模型

### 2.1 系统建模

#### 2.1.1 计算模型的基本概念 — 一个二维框架

首先了解计算模型的基本概念。可以从两个维度来了解计算模型的基本构成。一个维度是时间，相应的有同步模型和异步模型。一个维度是(空间)通信方式，包括消息传递(简称为MSG)模型和共享存储(简称为SHM)模型。

📖 阅读[AW04]的第1、2、4章。阅读[Asp19]的第2、15章。

📖 计算模型的二维构成，参加L1胶片中的图。

#### 2.1.2 模型间模拟的概念

其次要了解计算模型之间的模拟(simulation)这一引申概念。了解模拟本身之外，还需要辨析、深入理解不同计算模型的优劣、权衡。

📖 阅读[AW04]的第7、9章。阅读[Asp19]的第16章。

#### 2.1.3 失效模型的基本概念

首先了解crash failure、Byzantine failure、link failure的基本定义和大致含义，后面结合具体的问题、算法、系统，做进一步了解。

### 2.2 特定领域分布式系统的建模

#### 2.2.1 分布式数据库系统

数据划分(partition, sharding)的问题。

数据复制(replication)的问题。

对上层应用编程提供事务(transaction)的支持。

分布式数据库中的容错问题。

### 2.2.2 分布式消息分发中间件

分布式消息分发系统，分布式消息中间件(DMS, Distributed Messaging System)。  
发布-订阅(Pub-Sub, Publish-Subscribe)系统。

## 2.3 问题规约

问题的规约(specification)。

安全性(safety)。

活性(liveness)。

### 2.3.1 问题间的归约

问题之间的归约(reduction)。

## 2.4 消息传递模型

### 2.4.1 同步

### 2.4.2 异步

## 2.5 共享内存模型



## 第二部分

## 消息传递

## 第三章 分布共识问题及基本算法

分布共识问题。相关工作很多，此处先介绍基本问题，和朴素的算法。后面分多个专题讨论分布共识方面的经典工作。

📖 蛮力算法：[AW04]第5章，问题的定义，和一个容忍crash failure的同步系统的 $f + 1$ 轮蛮力算法。

📖 经典算法：经典的共识协议Paxos[Lam01]、VR[OL88]。

📖 后续变体：ZK [JRS11]，Raft [OO14]。

## 第四章 Paxos

## 第五章 Raft

## 第六章 原子性广播协议Zab

Zab [JRS11, Med12]。

## 第七章 分布共享内存

分布共享内存(Distributed Shared Memory, DSM)。

### 7.1 原子寄存器

#### 7.1.1 单写

#### 7.1.2 多写

## 第八章 分布事务的原子性提交

原子性提交(atomic commitment)。

## 第三部分

## 共享内存



## 第九章 互斥

## 第四部分

## 理论专题

## 第十章 不可区分性的构造

经典的chain argument。

🍃 一个关于fast read-and-write atomic register的例子，参考[DGLV10]。

更复杂、更精细的构造。

🍃 一个关于fast write atomic register的例子，参考[HHW20]。

## 第五部分

## 系统专题

## 第十一章 Paxos协议的系统实现

## 第十二章 Raft协议的系统实现

etcd<sup>1</sup>中的Raft实现。

BRaft<sup>2</sup>。

MongoDB<sup>3</sup>中的Raft实现。

---

<sup>1</sup><https://etcd.io/>

<sup>2</sup><https://github.com/baidu/braft>

<sup>3</sup><https://www.mongodb.com/>

## 第十三章 Zookeeper：Zab协议的系统实现

# 第十四章 容错与可靠性

## 14.1 理解failure

从节点的视角。crash failure。Byzantine failure。

🍃 经典的共识协议Paxos[Lam01]、VR[OL88]，它们容忍的是crash failure。其中，VR扩展出了经典的拜占庭容错的共识协议PBFT [CL02]。

从网络连接的视角。

🍃 [Lyn96]第5章，容忍link failure的协同攻击问题。

🍃 大量真实但是anecdotal案例的精彩讲述 [BK14]。基于大量开源项目bug report分析partial partition [AAAAK20]。

## 14.2 发现failure

## 14.3 容忍failure

错误自动注入技术。

🍃 [AT17]。



## 第六部分

### 验证专题

## 第十五章 分布式协议的模型检验

主要以TLA+规约与模型检验为例。

## 第十六章 分布式系统实现的模型检验

✎ [GWZ<sup>+</sup>11]。

## 第十七章 分布式协议的定理证明

## 第十八章 分布式系统实现的定理证明

🍃 IronFleet [HHK<sup>+</sup>15]。

## 参考文献

- [AAAAK20] Mohammed Alfatafta, Basil Alkhatib, Ahmed Alquraan, and Samer Al-Kiswany. Toward a generic fault tolerance technique for partial network partitioning. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI'20)*, pages 351–368. USENIX Association, November 2020.
- [Asp19] James Aspnes. *Notes on Theory of Distributed Systems*. Yale University, CPSC 465/565, 2019.
- [AT17] Peter Alvaro and Severine Tymon. Abstracting the geniuses away from failure testing. *Commun. ACM*, 61(1):54–61, December 2017.
- [AW04] Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons, 2004.
- [BK14] Peter Bailis and Kyle Kingsbury. The network is reliable. *Commun. ACM*, 57(9):48–55, September 2014.
- [CL02] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Trans. Comput. Syst.*, 20(4):398–461, November 2002.
- [DGLV10] Partha Dutta, Rachid Guerraoui, Ron R. Levy, and Marko Vukolić. Fast access to distributed atomic memory. *SIAM J. Comput.*, 39(8):3752–3783, December 2010.
- [GWZ<sup>+</sup>11] Huayang Guo, Ming Wu, Lidong Zhou, Gang Hu, Junfeng Yang, and Lintao Zhang. Practical software model checking via dynamic interface reduction. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, SOSP '11*, pages 265–278, New York, NY, USA, 2011. ACM.
- [HHK<sup>+</sup>15] Chris Hawblitzel, Jon Howell, Manos Kapritsos, Jacob R. Lorch, Bryan Parno, Michael L. Roberts, Srinath Setty, and Brian Zill. Ironfleet: Proving practical distributed systems correct. In *Proceedings of the 25th Symposium on Operating Systems Principles, SOSP '15*, pages 1–17, New York, NY, USA, 2015. ACM.

- [HHW20] Kaile Huang, Yu Huang, and Hengfeng Wei. Fine-grained analysis on fast implementations of distributed multi-writer atomic registers. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, PODC'20, page 200–209, New York, NY, USA, 2020. Association for Computing Machinery.
- [Hil18] Hillel Wayne. *Practical TLA+: Planning Driven Development*. Apress, 2018.
- [HR04] Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge University Press, USA, 2004.
- [JRS11] Flavio P. Junqueira, Benjamin C. Reed, and Marco Serafini. Zab: high-performance broadcast for primary-backup systems. In *Proc. DSN'11, IEEE/IFIP Conference on Dependable Systems and Networks*, pages 245–256. IEEE, 2011.
- [Lam01] Leslie Lamport. Paxos made simple. *ACM SIGACT News (Distributed Computing Column)* 32, 4 (Whole Number 121, December 2001), pages 51–58, December 2001.
- [Lyn96] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.
- [Med12] Andre Medeiros. Zookeeper's atomic broadcast protocol: Theory and practice. Technical report, 2012.
- [MWA<sup>+</sup>19] Ellis Michael, Doug Woos, Thomas Anderson, Michael D. Ernst, and Zachary Tatlock. Teaching rigorous distributed systems with efficient model checking. In *Proceedings of the Fourteenth EuroSys Conference 2019*, EuroSys '19, pages 32:1–32:15, New York, NY, USA, 2019. ACM.
- [NRZ<sup>+</sup>15] Chris Newcombe, Tim Rath, Fan Zhang, Bogdan Munteanu, Marc Brooker, and Michael Deardeuff. How amazon web services uses formal methods. *Commun. ACM*, 58(4):66–73, March 2015.
- [OL88] Brian M. Oki and Barbara H. Liskov. Viewstamped replication: A new primary copy method to support highly-available distributed systems. In *Proceedings of the Seventh Annual ACM Symposium on Principles of Distributed Computing*, PODC '88, page 8–17, New York, NY, USA, 1988. Association for Computing Machinery.
- [OO14] Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm. In *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, USENIX ATC'14, pages 305–320, Berkeley, CA, USA, 2014. USENIX Association.
- [TS06] Andrew S. Tanenbaum and Maarten van Steen. *Distributed Systems: Principles and Paradigms (2nd Edition)*. Prentice-Hall, Inc., USA, 2006.