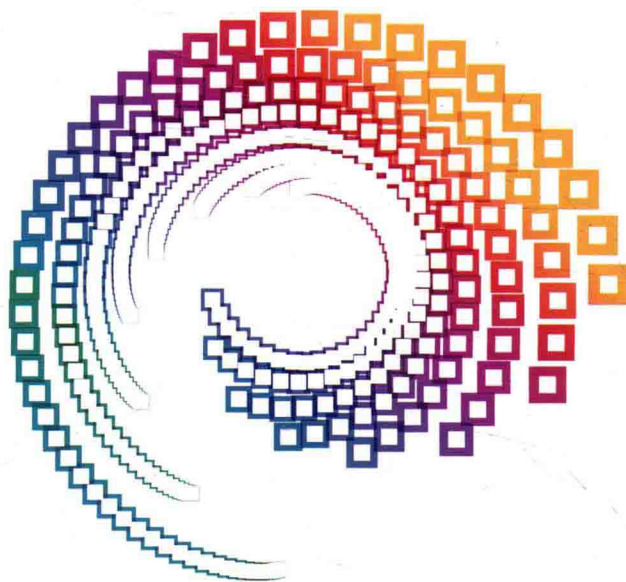




多位大数据专家联袂推荐，资深系统研发工程师撰写，参透大规模分布式实时处理系统  
抽丝剥茧，从概念、原理到分布式实时计算框架实现，兼顾理论与实践，带领读者逐步  
实现一个基于C++ 11、高性能的分布式实时处理系统



技术丛书



Distributed Real-time Processing System  
Principle, Architecture and Implementation

# 分布式实时处理系统

## 原理、架构与实现

卢誉声◎著



机械工业出版社  
China Machine Press



技术丛书

Distributed Real-time Processing System  
Principle, Architecture and Implementation

# 分布式实时处理系统

## 原理、架构与实现

卢誉声◎著



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

分布式实时处理系统：原理、架构与实现 / 卢誉声著. —北京：机械工业出版社，2016.6  
(大数据技术丛书)

ISBN 978-7-111-53996-4

I. 分… II. 卢… III. 分布式处理系统 IV. TP338

中国版本图书馆 CIP 数据核字 (2016) 第 128764 号

# 分布式实时处理系统：原理、架构与实现

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）

责任编辑：高婧雅

责任校对：殷虹

印刷：北京市荣盛彩色印刷有限公司

版次：2016 年 6 月第 1 版第 1 次印刷

开本：186mm × 240mm 1/16

印张：31.5

书号：ISBN 978-7-111-53996-4

定价：99.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88379426 88361066

投稿热线：(010) 88379604

购书热线：(010) 68326294 88379649 68995259

读者信箱：hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

## Praise 本书赞誉

本书不但讲解高性能分布式实时处理系统编程的原理，特别对内存资源管理、编码解决方案、并发与异步处理、线程模型、批处理与实时处理的区别、消息队列、动态装载等作了详细介绍，还深入分析了实时处理系统的架构以及内部实现，最后详细分析了怎样实现一套分布式实时处理系统 Hurricane。本书非常适合大数据开发人员和架构师阅读，同时可以解决性能优化的很多问题。

——卢亿雷，AdMaster 技术副总裁兼总架构师

分布式系统可以追溯到 20 世纪 60 年代的 ARPANET。随着物联网、边缘计算和其他相关领域的蓬勃发展，对高性能分布式实时处理系统的需求日益增多。这本技术专著着眼于实际分布式框架的编程应用，将助力有志于该领域的软件开发人员。

——安宁，Oracle 空间数据部门首席工程师、IEEE 高级会员和 ACM 终身会员

本书不仅仅是一部讲授代码编写的书籍，还是一个开源社区的星星火种，我特别推荐所有对实时大数据分析感兴趣的同业中人阅读此书，并以此作为迈入下一个开源大时代的第一步。

——徐立冰，思科系统高级客户经理

通常来说，在互联网+、大数据时代盛行拿来主义；有开源的 Spark、Strom，绝对不会动自己构建分布式系统的心思。但是作者不然，深厚的 C++ 功底以及对分布式计算框架的深度理解，构建了高性能分布式、实时处理系统 Hurricane，进入了更高层次的追求。

静下心来，这本书将对自己的 C++ 编程、分布式存储系统、分布式计算框架、分布式通信等知识进行了洗礼，不知不觉间就提升了自己的层次，期待阅读本书之后，也可以构建一个全新的分布式实时处理系统，变成分布式实时计算领域的贡献者。

路已铺好，如何贡献，那是追求。

——于俊，科大讯飞大数据专家

“天下武功，无坚不摧，唯快不破”，如果说以 MapReduce、Spark 等为代表的批处理方式，是数据处理的“坚”，则 Storm、Hurricane 等流处理系统，充分体现了数据处理的“快”。本书由浅到深，从基础的分布式系统的概念、网络通信和 C++ 11 高性能编程，到流处理中的消息、消息源、处理单元、收集器、计算拓扑等重要抽象的实现，到日志处理、频繁组合等典型应用，深入研究了一个典型分布式流处理系统的各个重要方面，推荐。

——蔡斌，腾讯科技高级工程师

分布式实时处理系统，难就难在把分布式和实时处理结合起来。本文从拆解留言板系统开始节节升级，直至推出 Hurricane 实时处理系统，旁征博引，纲举目张。充分体现了分布式构建和实时处理的细节考虑。

——彭敏，思科系统（中国）研发有限公司服务器与平台架构部门研发经理

作者以他参与 Cisco 和 Autodesk 诸多大型系统开发时所积累的经验以及对开源技术多年的钻研铸就本书。本书由理论到应用，由实现到优化，由浅入深，抽丝剥茧地把这么浩繁的概念在本书中讲得十分清晰。

——张洋，Autodesk（中国）研发中心平台架构研发部门经理

正如“如何阅读本书”中说的一样，“本书从最基本概念作为引子，逐步引入高性能分布式实时处理系统编程所需要的方方面面，抽丝剥茧把实时处理系统的架构以及实现娓娓道来”，有概念、有理论，有本身知识体系的交代，也有周边必要知识内容的说明，有实战、有案例，不空谈，能落地，是一本不可多得的学习分布式实时架构的好书。

——贾锋，知名大数据布道者和践行者

2016年春节刚过，此时距离我出版《腾云》已经过去三年，而写作一本技术书籍的艰辛仍然历历在目。我身边有不少大拿级别的牛人，不少都被出版社试探过，但真正动笔的寥寥可数，主要原因是写书实在不是一件吃饭喝酒般轻描淡写的事情。在被本职工作和家庭琐事折腾得死去活来的间隙，强打起精神一砖一瓦地堆砌出几十万字，想想就让人却步，而这个过程往往历时数月，这意味着你要过上小一年的苦行僧式生活。若不是对书中主题保持着异乎寻常的兴趣和坚定不移的信念，一般人不会轻易开始这段苦旅。正是因为这个原因，我相信卢誉声的这本书一定是个干货满满的好东西。

优秀的技术书籍都有一个特点，那就是自下而上的阐述方式，从最底层、最实际的操作层面入手，而不是形而上的口号式概念，而本书则正是这样一本书。读者从第1章就能实际触摸到实操的快感，书中讲解了大量实例，我常说“一百字的定义，不如一句话的实例”，通过这些范例，读者可以快速获得感性认识，进而随着章节的推进把这些认知归纳总结为自己能够掌握的方法论。在游泳中学习游泳，说的就是这种简单、实用的学习思路。而考验一个作者功力的地方，就是他能否针对每个关键概念提出恰当清晰的例子，并阐述清楚。从我读到的篇章来看，卢誉声的这个工作完成得相当不错。这样一本朴实而扎实的技术书籍反映的是作者的态度，作为读者，在阅读过程中我可以感受到字面下作者热切希望跟广大同行分享的激情，那种把自己的认识和盘托出的诚恳是每一个痴迷于技术的工程师都有过的感受，卢誉声体会过，我体会过，这本书未来的大多数读者都体会过。正是这种对技术的诚恳让我相信本书值得一读。

回过头来我想说说对开源的认识。本书由 Apache Storm 说开，Apache Storm 作为一个顶级开源项目在业界的影响力无需多言，为了了解项目的背景，我特意读了 Storm 项目的孕育者 Nathan Marz 的长文《History of Apache Storm and lessons learned》。我发现，虽然 Nathan 充分认识到 Storm 能够带来的商业潜力（赶在 Twitter 收购前公开展示 Storm 的

效果),但他从来没有动摇过将这个项目完全开源的打算,甚至应该反过来说,他从来没想到要将 Storm 作为一个私有计划保持下去。2011 年 7 月, Nathan 所在的 BackType 正式被 Twitter 收购,几乎毫不犹豫的,他旋即开始着手将 Storm 开源,这之后便是大家熟悉的故事, Storm 以令人炫目的速度吸收开发者,并在短短三年后,于 2014 年 9 月 17 日正式成为 Apache 顶级项目。

为什么会有开源运动?这已经是一个无法再吸引注意力的老旧话题,无数人从商业、技术、社会等领域给出了无数严密的解答,但我们真的从心里认可这种行为了吗?特别是在国内的商业环境下,广大工程师每天享受开源项目的成果之余,真的理解开源运动的深意了吗?至少对我来说,花了很长一段时间才得出能够说服自己的解释。

第一个我无法理解的现象就是,开源并没有带来可量化的商业价值。作为开源世界的老大哥, Red Hat 的管理层向股东保证在 2016 财年达到 20 亿美元销售额,此时距离 Red Hat 成立已经超过 20 年。作为对比,2015 财年 Microsoft 的销售额超过 930 亿美元,如果说今日的微软已经包括了游戏机硬件、搜索等与软件不相关的业务,那么另一个传统软件领域的代表 Oracle 在 2015 年营收达到 382 亿美元,而 SAP 也有 200 亿美元。这些被嘲笑成“史前恐龙”的传统软件厂商在不同场合被描述为落后生产力的代表,它们站在开源潮流的对立面,出于狭隘的商业利益,沿着封闭、自我的路线一意孤行。可事实是,软件产业说到底同卖手机、卖汽车一样,仍然是一门生意,遵循用户用脚投票的商业规律,既然开源这么美好,为什么没有像苹果手机一样,通过巨大的商业成功快速颠覆原有模式呢?

另一个一直以来我没想明白的问题是,作为最终用户的工程师为什么要拥抱开源呢?开源固然可以给程序员带来莫大的快感,不管是个人成就还是物质回报,以往依附于大型软件公司的独立程序员现在有机会在社区通过个人贡献树立更大的影响力;可是对于绝大使用这些产品的工程师来说,开源和商业产品的区别就没有那么大了,而使用开源还伴随着学习成本和不稳定的后期支持。即使如此,我们周围抵触开源的声音却越来越小,即使那些最老资历、最忠诚的 Oracle DBA 也开始接触 MySQL,很有意思?

把时间拉回 20 世纪 90 年代,彼时个人电脑还是黑科技代表之一,学校还会开设“电脑课”教授基本操作,这类课程中往往很重要的一个章节就是“五笔打字”法。打字这个技巧放在今天几乎是跟走路、吃饭一样的基本生存技能,90 后一代已经很难回忆起来自己是在哪个时刻“学”会了打字,大多数都是自然而然在日常生活中磨练出来的。20 年前的一门专业技巧现在已经完全融入大众生活,这其中蕴含了一个有意思的规律,即任何一种技能都会随着时间的推移失去门槛,同时在这个过程中经过无数人的实践和磨练,这项技能已经进化出一套最有效率的模式,后来者可以跳过探索、试验的过程,用最短的时间直接掌握这套模式

就能实现之前高手级别才能达到的效果。在打字这个例子中，对于中国人来说目前在高效与易学间取得最佳平衡的是具备联想功能的拼音输入法，因此年轻的电脑用户只需听从朋友推荐下载正确的软件，两三天内就能练就足够应付日常交流的打字能力。

如果把视线拉远，欣欣向荣的新型操作系统和数据库正是这种技能门槛不断拉低的现象在软件领域的投射。操作系统、大型数据库这些领域在 20 世纪 90 年代是皇冠顶上的宝石，全世界也只有那么一小簇顶尖专家能够弄明白其中的奥妙，而时间过去 20 年，Microsoft 们已经培养出一大群熟悉这些大型系统的专家，人力门槛不复存在，而搭建一个操作系统或数据库的基本方法论现在已经非常成熟，因此开发操作系统不再有那么耀眼的光环，越来越多的政府机构、企业、科研机构甚至个人进入这个领域，并且取得不错的成果。这完全是因为最初的那一批精英已经填平了这条路上的大坑，并将他们提炼出来的最优方法论形成“可复制的经验”，而后来人能够直接利用这些“可复制的经验”，快速经过基础知识积累阶段，直接针对当下的难点攻坚，从而令一些出色的后来者能够进一步推高整个领域的高度。

在没有开源运动的时代，“可复制的经验”的传承是受到严格限制的，要么在企业内部形成专利，只有技术团队的核心成员能够接触到，要么在科研机构的高墙后，少部分有能力进入高墙后的精英得以一窥究竟。开源几乎是以几何倍数放大了“可复制的经验”的传播速度，这种方式在技术领域带来的后果是极大地加快了技术本身的演进，这很好理解，因为参与的人多了，众人拾柴火焰高，自然比小团体的做法有效率。而在商业领域，开源则令资源配置更有效率，开源行为本身会大量产生“可复制的经验”，从而反过来进一步拉低特定技术领域的门槛，加速技术的演进，企业的决策者发现新技术的成熟速度大大加快，因此他们必须更加积极地把资源配置到更前沿的领域以保持竞争力。回到 Red Hat 的例子，虽然这家公司本身的销售额永远不可能达到 Microsoft 或 Oracle 的高度，但 Red Hat 以远比 Microsoft 小得多的规模提供了一个同样可靠并更加灵活的操作系统，为整个行业释放出大量优秀工程师资源，这些人才将进入云计算、大数据等新兴行业，在新的山头攻坚。如果没有 Red Hat 这样的企业以最有效率的方式为行业提供基础设施，新的技术领域很难建立足够的人才队伍，整个行业的发展速度也会缓慢下来。因此，我们不能只看 Red Hat 的销售额，还应该看看 AWS、Salesforce 这样的新兴玩家，正是因为有了 Red Hat，才有后者的高速发展。对于个人而言，这种大趋势是不可阻挡的，聪明的老专家们自然会即时调整方向，拥抱开源。

开源运动近年来已经逐渐突破计算机软件领域，开始向其他行业扩展，例如开源服务器硬件、开源网络设备，甚至开源的 IT 管理流程。说白了，开源是一种新时代的知识传承模



式，未来的世界将处处开源，竞争的壁垒将体现在高效协调资源的能力，而不是对特定知识的独占。当我知道卢誉声将把书中提到的 Hurricane 完全开源时，我非常赞同他的做法。因为这个动作，本书不仅仅是一部讲授代码编写的书籍，还是一个开源社区的星星火种，我特别推荐所有对实时大数据分析感兴趣的同业中人阅读此书，并以此作为迈入下一个开源大时代的第一步。

徐立冰

思科系统高级客户经理

最近，我陷入纷繁复杂的产品问题中，难以自拔，而当我看到这本书时，感想良多，思绪万千。想到我刚工作时非常喜欢一类书，“xx 技术 yy 天从入门到精通”，一般都是从基本概念到实际操作全程指导，非常适合入门新手。随着工作的深入和扩展，百度、谷歌和各种技术论坛、知识库又成为更常用的查询工具。

然而，正如武打小说里的速成功夫或者一招鲜，始终不能让我们到达真正高手的境界。系统的学习和实践一门技术才是攀上高阶的王道。

我和作者誉声共事多年，多数时间从事分布式实时处理系统设计和实现，誉声举一反三，经常通过具体案例和问题，总结、归纳出规律和特点，继而演绎出适应性更强的结构，我受益良多。

现在这份关于分布式实时处理系统的佳作由浅入深，由易到难。从头读来，可以感受到完整的分布式系统的清晰脉络。尤其难得的是，既有高屋建瓴的架构设计，又有 C++ 最细、最艰深的要点剖析；既有明确的理论指导，又有细节到代码行的具体实现。这就是一套进阶大规模计算领域的秘笈。

分布式实时处理系统可以说是软件发展中里程碑式的智慧结晶，凝结了无数高手的心血，也是大规模业务持续推动的结果。一般只有很大的软件公司或者开源组织有能力投资发展。能有长期深入的实际工作经验很不容易，可以看出本书都是实战经验总结。

正如文中所提及的各种分布式系统，不仅要求架构合理、安全可靠、性能优异、开放标准、弹性互联，而且对于每行代码都有严格要求，更难的是需要有一支强大的团队才能坚持到成功。而作为一个团队，正需要每个人都要有本文所体现出的统一意志。

分布式实时处理系统，难就难在把分布式和实时处理结合起来。本文从拆解留言板系统开始节节升级，直至推出 Hurricane 系统，旁征博引，纲举目张。充分体现了分布式构建和实时处理的细节考虑。

本人也仍然战斗在一线的分布式实时处理系统中，深感个中艰难，尤其是带领一支强大战队不停前行。时刻需要类似本文的“系统综述”保障团队的思想统一、步调一致，不断创造新高度。

在这里，我愿意推荐好文与君共绘。在软件海洋的最精彩之路互进共勉。

彭敏

思科系统（中国）研发有限公司服务器与平台架构部门研发经理

并行计算和分布式系统一直都是个迷人的主题。近十年来，随着云计算和大数据的发展，它也逐渐从云端走下。如今，几乎所有人都可以尝试使用各种编程语言去构建各种规模的分布式应用。广大的 C++ 开发者也被深深地吸引，为之兴奋。C++ 11 的发布让这门已有三十多年历史的语言充满了活力，我们不禁想问：这百多项 C++ 11 的新特性如何能被高效地应用于分布式系统的开发？是不是该有一本书来谈谈这个问题呢？

能在第一时间看到本书的原稿，我感到很高兴。因为这就是一本关于实时分布式系统的书，一本关于 C++ 11 的书，更是一本关于实战的书。全书以 Hurricane 这个自制系统为例，清晰地将实时分布式系统从概念，到实现，再到应用和部署完全地呈现在了读者面前。并将 C++ 11 的诸多新特性贯穿其中。本书的主题是探讨互联网时代之下的云计算和大数据处理的高效实现。书中所述的高性能编程思想和 C++ 11 应用放在其他类型系统里同样适用。

这是一本讲技术的书，而且写得很好。由理论到应用，由实现到优化，由浅入深，抽丝剥茧地把这么浩繁的概念在本书中讲得十分清晰。这也是一本讲技巧的书，在实现高性能系统和遇到困难时，定会派上很大用场。这还是本有活力的书，无论是 C++ 11、大数据，还是 Samuel、Lionel 这两个书中的年轻人，都让这本书充满了朝气。

读完本书，尚感意犹未尽。作者以他参与 Cisco 和 Autodesk 诸多大型系统开发时所积累的经验以及对开源技术多年的钻研铸就本书。相信作者的写作和分享一定会带给你们一段充满乐趣的阅读之旅。

张洋

Autodesk (中国) 研发中心平台架构研发部门经理

# 前 言 *Preface*

## 为什么要写这本书

云计算与大数据为计算机科学领域注入了前所未有的活力，而大数据的实时处理更是为海量数据处理和数据挖掘带来了崭新的契机。从某种意义上讲，传统大数据的批处理方式已经一定程度上解决了我们所面临的问题，由 MapReduce、Bulk Synchronous Parallel 以及其他计算范式衍生而来的大规模计算集群已经广泛运用于生产环境中。然而，客户的要求是永不满足的。他们想要更多的数据、服务、价值以及更多的便利。随着数据量的增加，对实时响应时间的需求也在提高，原本承载着海量数据处理任务的批处理系统在实时计算处理方面越发显得乏力。这么说的原因很简单，像 Hadoop 使用的 MapReduce 这样的数据批处理技术，其设计初衷并不是为了满足实时计算的需求。数据批处理系统与实时处理系统在需求上存在着本质的区别。要做到实时性，不仅需要及时地推送数据以便处理，还要将数据划分成尽可能小的单位，而像 HDFS 存储推送数据的能力已经远不能满足实时性的需求。

虽然目前 Apache Spark 等新平台和框架越来越为流行，同时极大提升了批处理的性能，但是由于这些传统批处理系统的运行机制本身，无法从根本上解决实时计算的问题。

因此，随着业务数据规模的爆炸式增长，对数据实时处理能力的需求也越来越大。专注于实时性、规模化的计算平台新时代已经悄然来临。

Apache Storm 的出现扭转了传统数据批处理系统的劣势，成为了真正意义上的实时数据处理系统。Storm 实时处理系统实现了一个可靠的、高容错性的实时分布式处理平台，那么这些听起来十分抽象的概念到底是如何实现的呢？这其实就是笔者写作本书的目的之一。当笔者深入研究和学习了 Storm 的内部结构和代码实现之后，就在思考有没有什么办法让其他更多的人少走弯路，并能在实践中对这个看似“神秘”的实时处理系统的设计与实现融会贯通呢？

## 本书特色

本书是一本由浅入深并详细讲解编写一套全新的基于 C/C++ 的实时处理系统的编程实战书。本书从基础知识开始，到实时数据系统的架构设计，到代码的实际编写，逐步实现一个完整的实时数据处理系统。本书把这套全新的高性能分布式实时处理系统命名为 Hurricane，该单词与 Storm 涵义类似，但略有不同，其中维基百科对 Hurricane 的解释是“A storm that has very strong fast winds and that moves over water”，即“在水面高速移动的飓风 (storm)”。

同时，为了支持高性能的实时处理系统，我们必须提供高性能的网络层，能够支持大量的并发，因此本书设计实现了一套跨平台的网络库 Meshy，并将其作为 Hurricane 实时处理系统的传输层。

为了编写更清晰、易于移植、易于维护的现代化 C++ 代码，我们在书中大量使用了 C++ 11 的特性，从一些小的语法点（如 auto、override）到 C++ 11 中新增加的库（如 thread、chrono、functional）到一些翻天覆地的语法特性（如统一初始化、Lambda 表达式）都有所涉及。每当遇到新的 C++ 11 知识时，我们都会着重向读者介绍。由于目前 C++ 14 还不够普及成熟，因此在本书中暂不考虑 C++ 14 的特性。

为此，本书一开始将会花费大量篇幅介绍分布式计算存储的概念以及网络通信的基础知识。接着阐述和分布式计算存储相关的网络高层抽象知识，为构建分布式网络应用打下坚实基础。接着集中介绍本书需要运用的 C++ 相关知识，包括 C++ 11 的语言特性以及需要了解的底层知识。之后就开始介绍 Hurricane 实时处理系统的设计方案，并引导读者一步步自己实现 Hurricane 实时处理系统。

完成 Hurricane 实时处理系统的主体功能部分后，我们转而介绍 Meshy，阐述如何实现 Meshy 这一跨平台的网络框架，并与 Hurricane 实时处理系统进行对接。为了实现跨平台的高性能网络通信库，我们必须学习使用 epoll、IOCP 等与平台密切相关的技术来保证系统性能。同时，我们也要学会如何编写管理一个需要考虑移植和平台兼容性的系统的技巧与实践方法。最后辅以实战用例讲解如何将该系统应用于实际的生产环境中。

总之，Hurricane 实时处理系统是一个使用 C++ 11 编写的，以高性能为关注点的分布式实时计算框架，使用流模型作为计算模型，同时提供更易于理解的高层接口。

希望读者能够从本书中或多或少学到点新的知识，能够对 C++ 语言以及网络通信有更加深入的认识，了解如何构建一个可应用于生产环境的分布式实时处理系统。

## 如何阅读本书

本书以最基本的云计算与大数据概念作为引子，逐步引入高性能分布式实时处理系统编

程所需要的知识，抽丝剥茧地把有关实时处理系统的架构以及内部实现娓娓道来。

**第 1 章** 介绍分布式系统的一些基本概念，以及开发实时处理系统所需要具备的一些重要知识点。

**第 2 章** 介绍分布式系统通信基础，包括 TCP/IP 以及 Socket 方面的基本概念，为后续开发网络库 Meshy 做知识储备。

**第 3 章** 介绍分布式系统通信所需的高层抽象，包括 RPC 远程过程调用、RESTful、消息队列等常用的通信模型。同时介绍基本的序列化概念与解决方案，并使用 Thrift 开发简单的公告牌服务，为 Hurricane 的开发建立通信抽象与框架上的基础概念。

**第 4 章** 介绍 C++ 高性能编程所需的基础与进阶知识，包括 C++ 中的内存资源管理、编码解决方案、并发与异步处理以及内存管理技巧，以及 C++ 11 中与内存管理、编码处理、线程模型相关的内容。

**第 5 章** 介绍分布式处理系统的基本概念，包括批处理与实时处理的区别，Hadoop 与 Storm 的基本介绍及基本模型。最后介绍可靠消息处理的基本思想。

**第 6 章** 介绍实时处理系统的总体架构与接口设计，包括消息源、消息处理器、数据收集器、元组以及序列化接口。

**第 7 章** 介绍服务组件的设计与实现，包括 Executor 及其消息队列、动态装载以及 Task 的设计与实现等。

**第 8 章** 介绍管理服务的设计与实现，其中包括集群管理器 President 以及节点管理器 Manager 的架构设计与编程实现。

**第 9 章** 介绍实时处理系统中各部分接口的实现，包括消息源、消息处理单元以及数据收集器的实现。

**第 10 章** 介绍可靠消息处理的概念、接口设计与具体实现，包括简单和高效的实现方案。

**第 11 章** 介绍底层数据传输层及 Meshy 的设计与实现，包括 I/O 多路复用的概念与实现方法、所需的基础工具，以及跨平台的实现方案。最后辅以实战用例来展示集成与使用 Meshy 的方法。

**第 12 章** 介绍事务性计算拓扑的概念、实现方案与编程实现，并介绍相关 API，以及如何使用 Cassandra 存储元数据。

**第 13 章** 介绍在不同的编程语言中实现计算拓扑的方法，并在现有技术基础上增加一些新的技术。

**第 14 章** 介绍基于 Hurricane 实时处理系统的高级抽象元语、分布式远程过程调用

(DRPC) 的设计、实现方案及编程实战。

第 15 章 介绍了基于 Hurricane 实时处理系统开发的日志流处理实例，其中包含日志流处理的整体流程、使用 Hurricane 处理日志的具体实现思路，以及使用 Hurricane 处理日志的具体实现。

第 16 章 介绍了基于 Hurricane 实时处理系统开发的频繁组合查找实例，其中包含频繁项集挖掘概念与方法、频繁二项集挖掘算法原理与实现分布式统计方法。并介绍如何使用 Hurricane 实现自己的频繁二项集挖掘系统。

第 17 章 介绍在 AWS 和阿里云上部署 Hurricane 实时处理系统，首先介绍在 AWS 上创建私有云和 EC2 实例的方法，接着介绍在阿里云上创建私有云和 ECS 实例的方法，最后介绍 Hurricane 的分布式部署原理与方法。

## 阅读前提

本书采用 Ubuntu 或 Debian 操作系统以及 Windows 操作系统作为基本的开发环境。此外，本书并不准备对基础的编程概念进行理论介绍。我们假定你在阅读本书之前已经达到基本的编程技术水平以及具备一定的 C++ 编程经验和功底。如果不是，笔者建议阅读《C++ 编程思想》作为基础来了解编程的基本概念，并阅读《高级 C/C++ 编译技术》作为提高。

## 本书版式约定

在本书中，读者会发现针对不同信息类型的文本样式。下面是这些样式的示例和解释。

所有命令行输入和输出如下所示：

```
mkdir Hurricane
cd Hurricane
```

代码清单通常以以下格式展现：

```
1 #include <iostream>
2 #include <cstdlib>
3
4 int main()
5 {
6     std::cout << "Welcome to Hurricane" << std::endl;
7
8     return EXIT_SUCCESS;
9 }
```

在正文中时常会用以下方式拓展所讲解的内容：





这里是相关提示的文字。

## 读者对象

(1) **大数据系统研发工程师**。本书不但讲解高性能分布式实时处理系统编程所需要的方方面面，抽丝剥茧地呈现出实时处理系统的架构以及内部实现，还带领大家自己编写一套分布式实时处理系统。

(2) **研发人员**。本书是一本深入剖析分布式实时处理系统编号的指南。

(3) **架构师**。本书是一本层次化分布式系统架构设计的实战书。读者可以深入理解分布式实时处理系统的内部构造以及重要组成部分，并自己设计分布式系统的各个层次。

(4) **编程初学者**。学习实战技术，掌握分布式系统开发中惯用的编程技巧。

## 勘误和支持

虽然笔者在编写本书的过程中经过反复审校，全力确保本书内容的准确性，但疏漏在所难免。书中难免可能会出现一些不妥或不准确的描述，恳请读者批评指正。本书所涉及的所有源代码及工程都可以从华章官网 ([www.hzbook.com](http://www.hzbook.com)) 下载，同时这些项目也都是开源项目。现在我怀着期盼和忐忑的心情，将这本著作呈献给大家，我渴望得到你的认可，更渴望和你成为朋友，如果你有任何问题和建议，请与我联系 (电子邮箱: [samblg@me.com](mailto:samblg@me.com))，期待能够得到你的真挚反馈。

## 致谢

在创作本书的过程中，我得到了很多人的帮助，这里必须要一一感谢，聊表寸心。感谢鲁昌华教授，在我的成长道路上给予了很大的支持和鼓励。感谢我在 Autodesk 中国研究院 (ACRD) 的同事和思科系统 (中国) 研发中心的朋友。特别是我的良师益友彭敏、旷天亮和徐立冰，在我的学习工作中给予了很大帮助。感谢我的好友金柳颀，感谢你在写作本书过程中的通力合作以及技术问题上的共同探讨。还要感谢机械工业出版社的高婧雅编辑对我的信任与支持。

谨以此书献给我最亲爱的家人与朋友，你们是我奋斗路上坚强的后盾。

卢誉声  
于上海