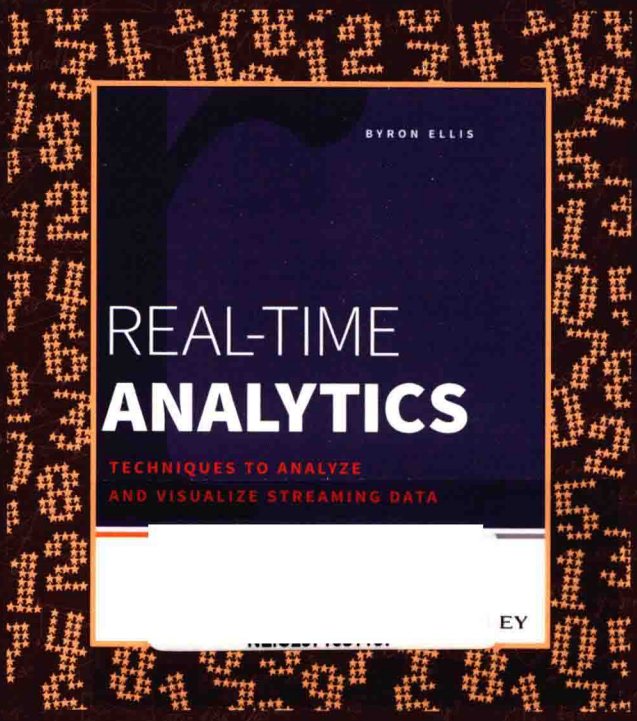
HZ BOOKS 华章IT
WILEY
数据科学与工程技术丛书

实时分析

流数据的分析与可视化技术

[美] 拜伦·埃利斯（Byron Ellis）著
王晓伟 译



REAL-TIME ANALYTICS
TECHNIQUES TO ANALYZE AND
VISUALIZE STREAMING DATA

机械工业出版社
China Machine Press

数据科学与工程技术丛书

REAL-TIME ANALYTICS

TECHNIQUES TO ANALYZE AND
VISUALIZE STREAMING DATA

实时分析

流数据的分析与可视化技术

[美] 拜伦·埃利斯（Byron Ellis）著

王晓伟 译

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

实时分析：流数据的分析与可视化技术 /（美）埃利斯（Ellis, B.）著；王晓伟译 . —北京：机械工业出版社，2016.4
（数据科学与工程技术丛书）
书名原文：Real-Time Analytics: Techniques to Analyze and Visualize Streaming Data

ISBN 978-7-111-53216-3

I. 实… II. ① 埃… ② 王… III. 数据处理 IV. TP274

中国版本图书馆 CIP 数据核字（2016）第 051704 号

本书版权登记号：图字：01-2014-7252

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：秦　健	责任校对：董纪丽
印　　刷：三河市宏图印务有限公司	版　　次：2016 年 4 月第 1 版第 1 次印刷
开　　本：185mm × 260mm　1/16	印　　张：19.75
书　　号：ISBN 978-7-111-53216-3	定　　价：79.00 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) 88378991　88361066　　投稿热线：(010) 88379604
购书热线：(010) 68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

译 者 序

大概十年前，正是学术界对流数据的研究方兴未艾之时，我刚开始攻读博士学位。当时，数据库领域正致力于研究流数据管理系统（Data Stream Management System，DSMS），并发展出 Stream、TelegraphCQ、Aurora 等 DSMS 原型系统，数据挖掘领域则致力于开发各种经典数据挖掘算法的流数据版本。

然而，在工业界，Storm、Kafka、ZooKeeper 等流分析基础架构的主要组件大多还处于内部开发的起步阶段，几乎不为人知。当时，除了 Gigascope 这个专用于网络监控的流数据管理系统，很少听说大规模实际应用流数据系统的案例。后来，在参与几个大规模网络安全监控项目的过程中，我也曾尝试应用学术界提出的几个 DSMS 来解决一些流分析问题，但效果均不理想。因此，从当时的情况来看，流数据的学术研究似乎离实际应用遥不可及，也正是由于这种悲观预期，我放弃了数据流这个研究方向，转而研究其他课题。

令人意想不到的是，近年来，随着 Storm、Samza 等流处理系统被贡献给开源社区，ZooKeeper、Kafka、Flume、Redis、MongoDB、Cassandra 等相关系统也纷纷在流分析领域占据一席之地。短短几年间，一个完整的流分析基础架构似乎已经成型。在以 Twitter、LinkedIn、Google 为代表的大数据技术和应用的先行者中，这些系统已经各司其职，担负起艰巨的流分析任务。这样令人眼花缭乱的进步，怎能不让人感叹大数据浪潮的威力！

正是因为自己以前的研究经历，当看到本书目录时，我真的有眼前一亮的感觉。本书不仅详细讨论流数据的采集、传输、处理、交付等过程中涉及的各个主流系统，还介绍略图、Bloom Filter、抽样等重要数据结构和算法的基本原理，以及流分析中的应用。在流数据分析如火如荼的今天，这样一本具有很强实用性，同时又有一定理论深度的书实属不可多得。我相信，无论是对流分析感兴趣的入门级读者，还是在企业从事了多年流分析业务的研发人员和管理人员，都能从这本书中找到自己想要的东西。

最后，衷心感谢机械工业出版社对本书的翻译和出版所给予的大力支持，感谢编辑和审校老师付出的辛勤劳动。由于本人水平有限，翻译过程中难免有错误或疏漏。如果您发现翻译有误或者有其他疑问，欢迎您通过邮件批评指正。我的邮箱是 gfkdwxw@aliyun.com.cn。

王晓伟

前　言

概述及组织结构

流数据处理涉及软件开发和工程领域的许多不同问题。一方面，它需要一个灵活的基础架构，能够迅速便捷地移动数据；另一方面，处理速度要“跟得上”数据采集的速度，还要能扩展，以适应源源不断的数据流，由于这个限制，流数据处理很青睐从其他领域借鉴而来的数据结构。最后，一旦数据采集和处理完毕，应该利用数据做些什么？对于这一点，有一批可以直接在流数据上运行的应用，这些应用已经在大多数相关企业中发挥作用，还有更多的应用一直处于企业的考虑之中。本书将流数据的所有这些方面糅合在一起，既可以充当大众读者的入门书籍，又对更专业的技术人员有参考价值。我们希望，通过阅读本书，你能够建立足够的自信，在企业中从头到尾实施一个流数据的概念验证项目，并尝试将其应用到生产环境。为了实现以上目标，不仅需要建立基础架构，还需要实现相关算法，为此我们将本书划分为两个独立的部分。

第一部分介绍流数据系统本身的基础架构，以及系统的运营问题。如果数据是流式的，但是仍然以批量的方式处理，那就不再是流数据，这些只是碰巧连续采集得到的批量数据，这样的数据对许多用例完全够用。但是，本书的假设是，人们已经认识到，如果数据生成后不久就可以由最终用户使用，这样会大有裨益。鉴于此，本书涵盖了完成这种任务所需要的工具和技术。

我们首先介绍流框架底层的概念和特性，包括流处理系统的各种组件。并不是所有项目在起始阶段都会用到这些组件，但所有成熟的流基础架构最终都会用到它们。然后我们在流基础架构的关键特性，即可用性、可扩展性和延迟的背景下讨论这些组件。

第一部分剩下的内容聚焦于实现或配置每个组件的具体细节。由于组件框架的广泛应用，多半已经不需要编写多少代码来实现组件。取而代之的是安装、配置，或者定制工作。

第 3 章和第 4 章介绍构建和协调数据移动系统所需要的工具。根据所处环境的不同，可能需要开发软件来直接与该系统集成，或者修改已有软件使之适应该系统。我们将讨论这两种途径的利弊。

数据一旦被移动，就必须得到处理，并最终存储起来。这是第 5 章和第 6 章的内容。这两章介绍主流的流处理软件，以及对数据存储系统的选择。

第二部分用流基础架构解决各种问题。仪表板与预警系统是流数据采集最早的应用，也是第二部分介绍的第一个应用。

第 7 章讨论从流环境向最终用户的数据交付问题。这是构建仪表板以及其他监控应用

所使用的核心机制。一旦交付，数据就必须呈现给用户，因此，该章还包括一节内容，专门讨论如何在基于 Web 的环境中建立仪表板可视化。

当然，所交付的数据经常要由处理系统进行聚集计算。第 8 章涵盖流环境下的数据聚集计算问题，特别是对多分辨率时间序列数据的聚集计算问题，该结果数据最终会被交付给第 7 章讨论的仪表板应用。

对数据进行聚集计算之后，数据中有什么样的模式这个问题就浮出了水面。是否有随时间变化的趋势？某个行为是否与先前观察到的行为有明显不同？要回答这些问题，需要一些统计学和随机过程方面的知识（一般来讲，这两方面的知识可以解答关于大规模数据采集的任何问题）。第 9 章对统计学和概率论的基础知识进行了简要介绍。其中还介绍了统计抽样的概念，它可以用来计算比简单的聚集更加复杂的度量。

抽样是对复杂度量进行近似的传统机制，但我们还可以通过其他机制更好地计算某些度量，略图（sketch）概率数据结构就是这样的一种机制。第 10 章将讨论略图，这需要大量使用第 9 章的概率论知识。略图通常具有更快的更新速度并占用更少的内存，因此特别适合流环境。

最后，第 11 章讨论聚集计算之外能够应用于流数据的一些更深入的话题。篇幅所限，这一章仅对所涉及的众多话题作简要介绍，实际上每个都可以单独成书。第一个话题是来自统计学和机器学习领域的流数据模型。这些模型是预测等许多应用的基础，用来估计未来的数据值。由于数据是流式的，可以将这些预测值与实际值相比较，据此再对模型进行修正。预测的应用很广泛，其中包括第 11 章中讨论的异常检测。

第 11 章还简要介绍优化和 A/B 测试。如果你可以预测用户对两个不同网站设计的反应，就可以利用这些信息，向用户展示具有较好预期反应的网站设计。当然，预测并不是完美的，也有可能对各个网站的设计效果给出糟糕的估计。要改进对特定网站设计的预测，唯一途径是收集更多的数据。这样，你就需要确定各网站设计要多久展示一次，从而保证在改进预测的同时，不会由于展示了非主流的设计，而牺牲展示效果。第 11 章给出一种常用于解决这类问题的简单机制，称为多臂赌博机（multi-armed bandit）。利用这种机制，你可以站在一个比较高的起点上对优化问题进行更深入探索。

本书面向的读者

我们曾在开篇提到过，本书意在吸引从事软件行业的广大读者，针对的是有兴趣开始涉足流数据及其管理的技术人员。正因如此，我们希望读者按顺序阅读，最终可以全面掌握流数据分析的基础知识。

即便如此，对于只熟悉这个领域的一部分内容，而对其他部分不甚了解的专家来说，本书也是值得一看的。例如，数据分析师或数据科学家可能在第 9 章和第 11 章的统计学方法方面有很强的专业背景，也可能对第 7 章的仪表板应用以及第 8 章的聚集计算技术有一定的经验，但对第 10 章的概率数据结构可能没有太多了解。如果不是要实际实现基础架构，而是想理解对设计的权衡是怎样影响所分析的数据的交付的，他们也可能会对前六章感兴趣。

与之类似，关注运营和基础架构方面的读者可能对第 1 ～ 6 章所讨论的话题颇有了解。他们可能没有接触过特定的软件，但肯定处理过许多类似的问题。本书第二部分，即

第 7 ～ 11 章可能对他们更有吸引力。系统监控是流数据的首批应用之一。异常检测这样的工具可以在诸如健壮的错误检测机制这样的开发中派上用场。

需要的工具

不管你喜欢与否，数据基础架构领域大多以 Java 虚拟机为基础。这里的原因比较复杂，但不管怎样，它始终是本书所需要的工具。本书中使用的软件和示例是基于 Java 7 开发的，通常也可以支持 Java 6 或 Java 8。读者应确保操作系统安装了合适的 Java 开发工具包。

由于要使用 Java，有必要安装一个编辑器。本书中的软件是用 Eclipse 编写的，项目也是用 Maven 构建系统来组织的。安装这两个软件将帮助你构建本书包含的示例。

本书也使用了其他软件包，具体安装方法会在相应的章节中讲到。

书中使用了一些基础的数学术语和公式。如果你的数学知识有些生疏，觉得这些概念有点难，Sheldon Ross 的《A First Course in Probability》会对你有所帮助。

配套网站

本书配套的网站中包含了每一章所有例子的代码包。每章的代码都划分成独立的模块。

有的代码在多章中共用。这种情况下，代码被复制到每个模块，以保证这些模块是自包含的。

网站还包含 Samza 源代码的拷贝。Samza 是一个新兴的项目，其代码库更新很快，这导致在编写和编辑相关章节的时候，本书中的示例可能已经难以运行。为了避免对读者造成困扰，我们在网上放了一个能与本书中代码良好兼容的项目版本。请参考 http://www.wiley.com/go/realtimeanalyticsstreamingdata。

扬帆起航

闲话少说，是时候踏上实际构建流数据系统的精彩旅程了。本书所讲的技术虽不是流数据处理的独门秘籍，但却都是我付出多年心血摸索出来的经验。我个人认为这些技术都非常好用，虽然在这个激动人心的时代，新事物始终会层出不穷。不管怎样，我希望，通过阅读这本书，你至少能少走弯路。

去找一些数据，让我们扬帆起航。

致　谢

在写这本书之前，每当在致谢部分看到“有太多要感谢的人”这句话，我都觉得是陈词滥调。现在我才明白，这样的话真的是发自肺腑。对于这本书，我确实有太多需要感谢的人，实在难以在这里对他们一一列举。

不过，我还是要为其中一些人的贡献专门致谢，尽管他们有的人未必知道这本书。第一个当然是 Wiley 出版社的 Robert Elliot，他欣赏我的演讲并认为可以写成一本近 400 页的书。没有他，就没有本书的面世。我还要感谢 Justin Langseth，他是我那次演讲的合作者，很遗憾他没有能与我合作编写本书。希望我们还有机会再次合作。还应该感谢由 Kelly Talbot 领导的编辑，Charlotte、Rick、Jose、Luke 和 Ben，是他们帮助我找到和纠正了许多错误，使项目始终按计划推进。如果书中还有什么错误，那绝对是我的问题。

我要感谢 DDG 所有的常客，本书中的软件至少有一半，甚至可能超过 80%，都是我端着啤酒杯聊天得来的。对于一个非正式的松散聚会，这真的很值得。感谢 Mike 第一个邀请我一同前往，感谢 Matt 和 Zack 这些年来举办这么多活动。

最后，我要感谢我这些年来的同事们。应该为你们忍受我各种轻率计划和笨拙补救颁发奖牌。特别感谢 adBrite 的全体成员，我们一起做出了许多很酷的东西，这些东西仍然处于领域的前沿。感谢 Caroline Moon，允许分析人员使用新奇的“ Hadoop”，以及收集更多的服务器。特别感谢 Daniel Issen 和 Vadim Geshel。我们并不是经常看法一致（可能仍然无法做到），但本书的内容很多都来源于我与他们两人的争论。

作者简介

Byron Ellis 是 Spongecell 公司的 CTO，该公司是一个总部位于纽约的广告技术公司，在旧金山、芝加哥和伦敦设有办事处。他负责公司的研发和计算基础设施的维护工作，在加盟 Spongecell 之前，他是在线交互技术“领头羊”企业 Liveperson 公司的首席数据科学家。他还在当时世界最大的广告交换公司之一 adBrite 担任过多项职务。他拥有哈佛大学统计学博士学位，攻读博士学位期间主要研究高吞吐量生物学实验数据中网络结构的学习方法。

技术编辑简介

Jose Quinteiro 有 20 年技术经验，参与过许多终端用户、企业、Web 软件系统和应用的设计与开发工作。他对于包括前后端的设计和实现在内的全套 Web 技术有着丰富经验。Jose 在威廉玛丽学院获得化学学士学位。

Luke Hornof 拥有计算机科学博士学位，曾参与创建了多个成功的高科技初创企业。他在编程语言方面发表了十多篇同行评审的论文，曾为微处理器、广告和音乐行业开发过商用软件。他目前的兴趣之一是使用数据分析技术来改善 Web 和移动应用。

Ben Peirce 在 Spongecell 广告技术公司负责研究工作和基础设施的管理。加盟 Spongecell 之前，他在医疗健康技术初创企业担任过多项职务，他还是 SET Media 公司的联合创始人之一，该公司是一个视频广告技术公司。他在哈佛大学工程与应用科学学院获得博士学位，研究方向是控制系统和机器人。

目　录

译者序
前言
致谢
作者简介
技术编辑简介

第 1 章　流数据简介……1

1.1　流数据的来源……2
1.1.1　运行监控……2
1.1.2　Web 分析……2
1.1.3　在线广告……3
1.1.4　社交媒体……3
1.1.5　移动数据和物联网……4
1.2　流数据的特别之处……5
1.2.1　始终在线，持续流动……5
1.2.2　松散结构……5
1.2.3　高基数的存储……6
1.3　基础架构和算法……6
1.4　总结……7

第一部分　流分析架构

第 2 章　实时流架构设计……10

2.1　实时架构的组件……10
2.1.1　数据采集……11
2.1.2　数据流程……11
2.1.3　数据处理……13
2.1.4　数据存储……13
2.1.5　数据交付……14
2.2　实时架构的特性……16
2.2.1　高可用性……16
2.2.2　低延迟……17
2.2.3　水平可扩展性……17
2.3　实时编程语言……18
2.3.1　Java……18
2.3.2　Scala 和 Clojure……19
2.3.3　JavaScript……19
2.3.4　Go 语言……20
2.4　实时架构概览……20
2.4.1　数据采集……20
2.4.2　数据流程……21
2.4.3　数据处理……21
2.4.4　数据存储……21
2.4.5　数据交付……22
2.5　总结……22

第 3 章　服务配置和协调……24

3.1　配置和协调系统的研发动机……24
3.2　维护分布式状态……25
3.2.1　不可靠的网络连接……25
3.2.2　时钟同步……25
3.2.3　不可靠环境下的一致性……25
3.3　Apache ZooKeeper……26
3.3.1　znode……27
3.3.2　监视和通知……28
3.3.3　保持一致性……28

3.3.4 创建 ZooKeeper 集群……28
3.3.5 ZooKeeper 本地 Java 客户端…33
3.3.6 Curator 客户端……39
3.3.7 Curator Recipes 组件……45
3.4 总结……50

第 4 章 流分析中的数据流程管理……52

4.1 分布式数据流程……52
4.1.1 至少交付一次……52
4.1.2 “n + 1”问题……53
4.2 Apache Kafka：高吞吐量分布式消息机制……54
4.2.1 设计与实现……54
4.2.2 配置 Kafka 环境……57
4.2.3 与 Kafka 代理交互……65
4.3 Apache Flume：分布式日志采集系统……66
4.3.1 Flume agent……67
4.3.2 配置 agent……68
4.3.3 Flume 数据模型……68
4.3.4 channel 选择器……69
4.3.5 Flume source……71
4.3.6 Flume sink……78
4.3.7 sink processor……80
4.3.8 Flume channel……80
4.3.9 Flume Interceptor……81
4.3.10 集成定制 Flume 组件……83
4.3.11 运行 Flume agent……83
4.4 总结……83

第 5 章 流数据的处理……85

5.1 分布式流数据处理……85
5.1.1 协调……86
5.1.2 分区和融合……86
5.1.3 事务……86
5.2 用 Storm 处理数据……86
5.2.1 Storm 集群的组件……87
5.2.2 配置 Storm 集群……88
5.2.3 分布式集群……89
5.2.4 本地集群……92
5.2.5 Storm 拓扑……92
5.2.6 实现 bolt……95
5.2.7 实现并使用 spout……99
5.2.8 分布式远程过程调用……104
5.2.9 Trident：Storm 的 DSL……105
5.3 用 Samza 处理数据……111
5.3.1 Apache YARN……111
5.3.2 从 YARN 和 Samza 开始……112
5.3.3 将 Samza 集成进数据流程……115
5.3.4 Samza 作业……116
5.4 总结……122

第 6 章 流数据的存储……123

6.1 一致性哈希……123
6.2 “NoSQL”存储系统……124
6.2.1 Redis……125
6.2.2 MongoDB……132
6.2.3 Cassandra……150
6.3 其他存储技术……159
6.3.1 关系数据库……160
6.3.2 分布式内存数据网格……160
6.4 存储技术的选择……160
6.4.1 键 - 值存储……160
6.4.2 文档存储……160
6.4.3 分布式哈希表存储……161
6.4.4 内存网格……161
6.4.5 关系数据库……161
6.5 数据仓库……161
6.5.1 将 Hadoop 作为 ETL 和数据仓库……162
6.5.2 Lambda 架构……166
6.6 总结……166

第二部分　流分析与可视化

第 7 章　流度量的交付 …… 168

7.1　流 Web 应用 …… 168

7.1.1　使用 Node …… 169

7.1.2　用 NPM 管理 Node 项目 …… 171

7.1.3　基于 Node 开发 Web 应用 …… 174

7.1.4　基本的流仪表板 …… 176

7.1.5　向 Web 应用加入流 …… 180

7.2　数据可视化 …… 190

7.2.1　HTML5 Canvas 和内联 SVG …… 190

7.2.2　数据驱动文档：D3.js …… 196

7.2.3　高层工具 …… 204

7.3　移动流应用 …… 208

7.4　总结 …… 209

第 8 章　精确的聚集计算和交付 …… 211

8.1　定时计数与求和 …… 214

8.1.1　基于 Bolt 的计数 …… 214

8.1.2　基于 Trident 的计数 …… 216

8.1.3　基于 Samza 的计数 …… 217

8.2　多分辨率时间序列的聚集计算 …… 218

8.3　随机优化 …… 222

8.4　时间序列数据的交付 …… 223

8.4.1　用 D3.js 绘制带状图 …… 224

8.4.2　高速 Canvas 图 …… 225

8.4.3　地平线图 …… 226

8.5　总结 …… 227

第 9 章　流数据的统计近似 …… 229

9.1　数值计算库 …… 229

9.2　概率和分布 …… 230

9.2.1　期望和方差 …… 231

9.2.2　统计分布 …… 232

9.2.3　离散分布 …… 232

9.2.4　连续分布 …… 233

9.2.5　联合分布 …… 235

9.3　参数估计 …… 236

9.3.1　参数推断 …… 236

9.3.2　Delta 方法 …… 237

9.3.3　分布不等式 …… 238

9.4　随机数产生器 …… 238

9.5　抽样过程 …… 242

9.5.1　从固定数据集中抽样 …… 242

9.5.2　从流数据中抽样 …… 243

9.5.3　有偏流抽样 …… 244

9.6　总结 …… 245

第 10 章　使用略图近似流数据 …… 246

10.1　寄存器和哈希函数 …… 246

10.1.1　寄存器 …… 247

10.1.2　哈希函数 …… 247

10.2　集合 …… 249

10.3　Bloom Filter …… 251

10.3.1　算法 …… 251

10.3.2　Bloom Filter 大小的选择 …… 253

10.3.3　并集和交集 …… 253

10.3.4　基数估计 …… 254

10.3.5　有趣的变体 …… 255

10.4　Distinct Value 略图 …… 258

10.4.1　Min-Count 算法 …… 258

10.4.2　HyperLogLog 算法 …… 260

10.5　Count-Min 略图 …… 264

10.5.1　点查询 …… 265

10.5.2　Count-Min 略图的实现 …… 265

10.5.3　Top-K 和“Heavy Hitters” …… 266

10.5.4　范围查询和分位数查询 …… 268

10.6　其他应用 …… 270

10.7 总结 …… 271

第 11 章 流数据的应用 …… 272

11.1 实时数据模型 …… 273
11.1.1 简单时间序列模型 …… 273
11.1.2 线性模型 …… 276
11.1.3 逻辑回归 …… 280
11.1.4 神经网络模型 …… 281
11.2 用模型预测 …… 289
11.2.1 指数平滑法 …… 289
11.2.2 回归法 …… 291
11.2.3 神经网络法 …… 293
11.3 监控 …… 294
11.3.1 离群点检测 …… 294
11.3.2 变化检测 …… 296
11.4 实时优化 …… 297
11.5 总结 …… 298

第 1 章
流数据简介

当今世界每天都在以更快的速度发展。人和地区之间的联系越来越紧密，人员和企业的反应速度也越来越快。这越来越接近于人类反应能力的极限，因此人们创建了一些工具处理决策者面临的海量数据，分析、展现这些数据，并在事件发生的时候及时做出反应。

这些数据的采集和处理有许多应用领域，其中某些领域将在下一节介绍。我们将在本章后面讨论这些应用。这些应用需要一个专门针对流数据的基础架构和分析方法。幸运的是，与以前的批处理类似，流基础架构的最新技术关注于使用商用硬件和软件来构建系统，而不是像互联网时代之前那样使用专用的实时分析系统。这样的系统与基于云的灵活环境结合在一起，使得实时系统的实现对几乎所有企业都变得触手可及。利用这些商用系统，企业能够实时分析数据，并且能随着时间的推移对基础架构加以扩展，以满足企业未来发展和变革的需求。

本书面向广泛的大众读者以及企业的技术实现人员，目的是使他们能够轻松应对全栈应用。当实时项目进入某个阶段时，它们应该成为易于修改的、敏捷且自适应的系统，这需要用户除了了解所关注的领域，还要对全栈应用有清晰的整体理解。“实时”不仅针对数据本身，对于新的分析任务的开发也同样适用。许多原来规划良好的项目之所以失败了，是因为项目的实现耗时太久，导致项目发起人转而从事其他项目，或者干脆忘记了当初为什么需要这些数据。通过让项目变得敏捷和可递增，就可以尽可能地避免这种情况。

本章划分为三小节，分别对应三部分内容。第一节介绍一些常见的流数据来源和应用。这部分内容基本上按照时间先后排序，并对流数据基础架构的起源给出了一些背景介绍。这个介绍不仅具有历史意义，更重要的是，其中介绍了当初开发的许多工具和框架，用来解决这些领域中的问题，其设计反映了它们当时所处领域的一些独特挑战。例如，第 4 章中介绍的数据移动工具 Kafka 就被开发作为 Web 应用工具，而第 5 章中涉及的处理框架 Storm，起初就是由 Twitter 开发，用于处理社交媒体数据的。

第二节涵盖流数据三个重要特性：持续数据交付、松散结构数据和高基数（high-cardinality）数据集。第一个特性当然是将系统定义为实时流数据环境的首要特性。其他两个特性尽管并不是流数据所特有的，却为流数据应用的设计者带来了独特的挑战。这三者一起构成了本质意义上的流数据环境。

第三节简要介绍对流数据使用基础架构和算法的重要性。

1.1 流数据的来源

流数据的来源多种多样，这一节只介绍一些主要的数据类别。尽管出现了越来越多的数据来源，也会有许多专用的数据来源，本节还是只讨论令流数据受到关注的应用领域中的数据类别。我们主要按照时间先后对这些应用领域排序。对书中讨论的软件追根溯源，它们中很多都是在解决这些特定应用领域中的问题时产生的。

书中给出的数据移动系统，刚开始是为 LinkedIn、Yahoo! 和 Facebook 的网站分析与在线广告处理数据。设计这样的处理系统是为了应对 Twitter 和 LinkedIn 这样的社交网络所带来的社交媒体数据处理的挑战。

Google 公司的商业帝国与在线广告息息相关，它们大量使用的高级算法与第 11 章中的算法异曲同工。Google 对一项名为深度学习的技术特别感兴趣，该技术利用超大规模神经网络来学习复杂模式。

通过使物联网以及其他高度分布式的数据采集手段经济可行，这些系统甚至正在开辟数据采集和分析的全新领域。我们希望，通过对以往应用领域的勾勒，能够对这些技术尚未预见的应用有所启示。

1.1.1 运行监控

实体系统的运行监控是流数据最初的应用。刚开始是用专用硬件和软件来实现的（在计算机时代之前，甚至是用模拟和机械系统）。当今最常见的运行监控的用例就是对为互联网提供动力的实体系统进行性能监视。

这些数据中心有几千个甚至数万个独立的计算机系统。这些系统持续不断地记录自己的物理状态数据，包括处理器温度、散热风扇的转速，以及消耗的电量。它们还记录磁盘驱动器状态信息和基础的运维度量，如处理器负载、网络活动，以及存储访问时间。

为了对所有这些实体系统进行监控，并及时发现问题，数据通过各种机制实时采集和聚集。刚开始的系统往往是专门的特殊机制，但是当这类技术开始应用于其他领域时，这些系统就开始使用与其他数据采集机制相同的采集系统了。

1.1.2 Web 分析

通过电子商务和在线广告引入的商用化 Web，引发了对网站行为的监控需求。与报纸的发行量类似，一天之中某个网站的不同访问者数量是重要的信息。对电子商务网站而言，与网站访问的人数相比，更重要的数据是人们浏览的各种商品以及这些商品之间的相互关系。

为了分析这些数据，许多专门的日志处理工具被研发出来，并推向市场。随着大数据以及 Hadoop 等工具的兴起，许多 Web 分析的基础架构转变为这类大规模批处理系统，用来实现推荐系统以及其他分析任务。显而易见，还可以通过对网站的结构进行实验，来看它们如何影响我们感兴趣的各种度量。这称为 A/B 测试，因为它采用与验光师确定最好搭配方案时相同的方法，即让两个选择相互对比来确定哪一个最好。这些测试多数是顺序进行的，但这存在许多问题，其中最重要的问题是进行对比研究所需要的时间量。

随着越来越多的企业挖掘其网站数据，对缩减反馈环节的时间，以及以持续不断的方式

采集数据的需求愈发重要。使用系统监控领域的工具，使得实时采集数据和执行 A/B 测试这样的任务得以并行，而不再需要顺序执行。随着度量维数以及对适当审计的需求（缘于用于计费的度量）的增加，数据分析领域开发了本书提到的许多流基础架构，将数据从分散在世界各地的网站服务器安全地移动到处理和计费系统中。

尽管这类数据通常位于企业内部而不公开，它们仍然是广泛的海量信息来源。其应用范围从用于计费的简单聚集，到用于根据当前浏览历史（在 Netflix 公司则是影片观看历史）对产品推荐的实时优化。

1.1.3　在线广告

实时数据的主要用户和主要来源之一就是在线广告。在线广告的最初形式与对应的印刷广告类似，都是提前几个月“购买”。随着广告交换和实时竞价基础架构的兴起，广告市场中很大一部分流量更具流动性，并且这部分流量始终在不断增长。

对于这些应用来说，在不同环境和不同网站上所花费的资金是以按分钟计费的方式管理的，这与股票市场几无二致。另外，这些广告购买需求经常以某种度量进行管理，例如购买数量（称为转化（conversion））甚至是某个广告的点击率这样更简单的度量。当访问者通过现代的广告交换进入网站，就会有许多竞价代理（每次可能有 30 或 40 个）被调用，它们实时地对页面点击量进行竞价。竞拍过程结束后，竞价获胜一方的广告就会被显示出来。这通常在加载页面其他部分的时候发生，所花费的时间一般不超过 100 毫秒。如果页面包含多个广告（这很常见），经常会对所有这些广告进行竞拍，有时会涉及多个不同的广告交换。

这个过程中所有的参与方，包括广告交换、竞价代理、广告客户和广告发行商，都会为了各自不同的目的实时采集数据。对于广告交换来说，数据是竞价过程的关键部分，同时对于实时反馈机制也很重要，这种机制用途广泛。例如对欺诈流量交易的监控，以及限制对各参与方广告显示的访问等其他风险管理活动。

广告交换双方的广告客户、广告发行商和竞价代理也都实时采集数据。他们的目标是管理和优化当前运行的竞价活动。从竞价的选择（如果是广告客户）或者“保留”价格（广告发行商），到决定对某个特定类型的流量哪个广告交换的价格最好，数据都是以分秒必争的方式管理的。

一个大型的广告活动或大规模网站可以轻松拥有几千万甚至几亿的页面点击量。如果将点击和转化等其他事件包括在内，事件的数量可以轻易翻番。一个竞价代理通常一次性代表许多不同的广告客户或广告发行商，因此每天采集到几亿到几十亿量级事件的情况很常见。即使是中等规模的广告交换，中间方每天也可以有几十亿的事件。所有这些数据一旦生成，都会被采集、移动、分析和存储。

1.1.4　社交媒体

另一种新近的海量数据来源是社交媒体，特别是像 Twitter 这样的开放数据来源。截至 2013 年年中，Twitter 声称每天能采集到 50 亿条推文，每秒 15 万条左右。这个数字当然还会不断上升。

这些数据被实时采集和传播，成为新闻媒体和全世界其他用户的重要信息来源。在 2011 年，纽约的 Twitter 用户收到华盛顿特区传出的发生地震的消息，大约 30 秒后地震才波及