大数据丛书

“十三五”国家重点出版物出版规划项目

Broadview
www.broadview.com.cn

深入理解Flink

实时大数据处理实践

余海峰 著

中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

大数据丛书

“十三五”国家重点出版物出版规划项目

深入理解Flink

实时大数据处理实践

余海峰 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书介绍了实时数据处理引擎 Flink，讲解了流处理 API、批处理 API、机器学习引擎 FlinkML、关系型 API、复杂事件处理，以及指标度量与部署模式，分析了流式数据处理理论中时间、窗口、水印、触发器、迟到生存期之间的关联和关系，深入分析了多项式曲线拟合、分类算法、推荐算法的理论和 FlinkML 实现。

本书适合希望快速上手 Flink 以开展实时大数据处理与在线机器学习应用的从业者阅读。

图书在版编目（CIP）数据

深入理解 Flink：实时大数据处理实践 / 余海峰著. —北京：电子工业出版社，2019.4
（大数据丛书）
ISBN 978-7-121-36045-9

Ⅰ. ①深… Ⅱ. ①余… Ⅲ. ①数据处理软件 Ⅳ.①TP274

中国版本图书馆 CIP 数据核字（2019）第 033826 号

策划编辑：郑柳洁
责任编辑：牛　勇
印　　刷：三河市良远印务有限公司
装　　订：三河市良远印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×980　1/16　印张：19　字数：308 千字
版　　次：2019 年 4 月第 1 版
印　　次：2019 年 4 月第 1 次印刷
定　　价：89.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

前言

实时大数据是与时俱进的变革

从互联网时代的数据爆炸，到即将大规模铺开的 5G 通信支撑下的物联网时代的大数据浩海，作为赋能工具的大规模数据处理，技术架构起到了决定性的作用，反过来也推动了技术架构与时俱进。

在谷歌公司发表的三篇划时代论文（分别介绍 MapReduce、GFS 和 BigTable）的推动下，开源项目 Hadoop 横空出世，并于 2008 年 1 月正式成为 Apache 的顶级项目；此后，Hadoop 迅速建立起大数据生态体系，并由此衍生出一系列大数据处理的理论和与之对应的大数据处理框架：从批处理到流处理，从 Hadoop 到 Storm / Spark，再到 Flink。本书将阐述大数据实时处理理论的变迁，并着重介绍流处理框架 Flink。

数据处理任务往往需要对全量数据进行计算，而全量数据很难使用传统关系型数据库进行批量计算，原因如下：

（1）磁盘寻址时间的提升速度远远落后于磁盘带宽的提升速度。如果数据访问包含大量的磁盘寻址，则大数据处理势必带来较大的延迟，因此基于传输带宽设计大数据处理系统更符合现状。

（2）相比全量数据计算，关系型数据库适用于在线事务处理（OLTP，On-Line Transaction Processing）场景，查询和更新是其设计的要点，索引是主要的设计方案。但是在大数据集的场景下，索引的效率往往不如全量扫描。因此，Hadoop 应运而生，借助 MapReduce 计算引擎成功解决了大数据所面临的可计算（可参考谷

歌的论文 *MapReduce:Simplified Data Processing on Large Clusters*）、伸缩、容错等困难，成为大数据系统的标配组件。

数据爆炸式增长，以及数据处理的实时性要求越来越高，大数据处理系统越来越复杂。在这种情况下，传统的 OLTP+OLAP（On-Line Analysis Processing，在线分析处理）系统架构不堪重负：复杂的数据需求实现流程、过载的数据仓库抽取任务队列、不同的技术栈带来的需求理解偏差等将导致数据从 IT 部门到 DT 部门的周期过长；微服务方法的大规模应用，导致在分布式系统中维护全局状态的一致性异常困难，而以数据流作为中心数据源的流处理方法能有效规避这种困难。有的学者甚至提出通过合理的架构设计，打破 CAP 定理。因此，低延迟、强一致性、适用于乱序的流处理框架 Flink 正席卷而来，即将成为大数据领域流处理的标配组件。

本书特色

本书将从多个角度讲解同一个技术概念：

（1）分析引入 Flink 这个技术概念的原因，使读者能够快速理解相关技术的应用场景，如为什么需要实时数据处理、为什么需要机器学习架构、为什么需要关系型 API、为什么需要复杂事件处理。

（2）剖析 Flink 技术的理论创新过程，使读者能够深入理解 Flink 的理论基础，使 Flink 应用开发架构师或工程师能够游刃有余地解决线上系统遇到的实际问题，如 Flink 一致性保证的异步检查点屏障的理论创新过程、机器学习中分类和推荐算法的分布式实现的理论创新过程、复杂事件处理的自动机理论创新。

（3）解析 Flink 编程 API 的架构。使读者可以从理论框架与 Flink 架构实现两个角度体会这个技术概念的内涵。

（4）总结应用 API 编程解决实际问题的方法。使读者能够在理解理论和编程 API 的基础上编程解决实际问题。

目录

第 1 章
流式数据架构理论

在移动互联网领域，个性化服务、极致的用户体验要求业务系统具备实时数据处理能力，传统的批处理数据架构已经不堪重负。经过一系列理论创新与实践探索，流式数据架构 Flink 在实时数据处理领域取得了巨大成功，正成为大数据处理的标配框架。

为了让读者厘清大数据处理架构变革的源与流，1.1 节先概述大数据处理架构的演进历程，如 Storm、Spark、Lambda、Flink；为了让读者更容易理解流式数据架构思想，1.2 节将以韩国 SK 电信的 Driving score 技术架构演变和流式数据架构在机器学习领域的应用为例，介绍流式数据架构的产生背景及应用场景；1.3 节将介绍流、时间、窗口、水印、触发器等，并在这些概念的基础上剖析数据处理的各种模式；1.4 节将梳理流式数据架构中主要概念间的关联和关系，并以实例分析根据事件时间开滚动窗口的内在机制；1.5 节将论述流式数据架构中一致性理论的基础及实现方式，如有状态计算、检查点、保存点等概念。

1.1 大数据处理架构演进历程

谷歌发表的三篇划时代论文（分别介绍 MapReduce、GFS 和 BigTable），特别

是介绍 MapReduce 的那篇论文，开启了大规模数据处理波澜壮阔的发展历程。一篇篇论文和那些大数据从业者耳熟能详的大数据处理架构，是这个历程中的重要里程碑，图 1-1 所示为主流大数据处理架构的发展历程。

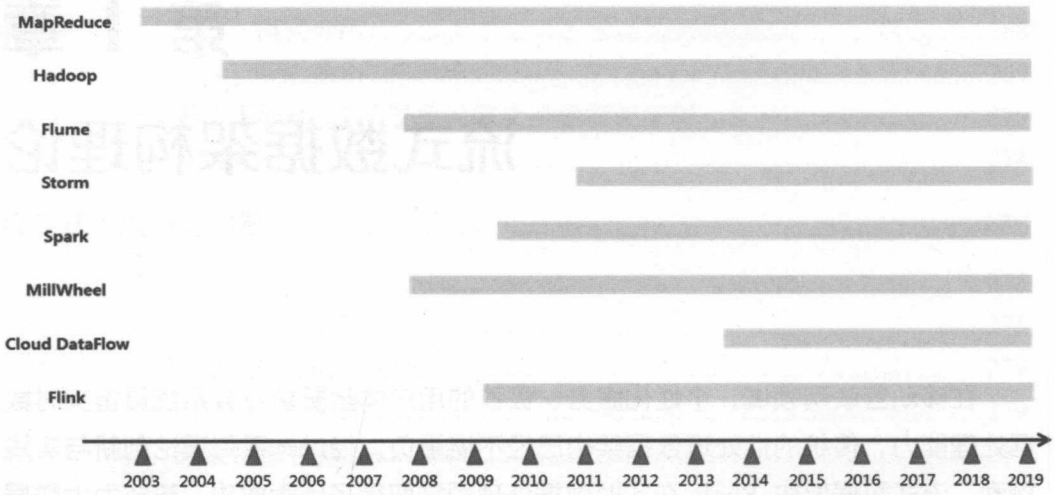

图 1-1　主流大数据处理架构的发展历程

2003 年，谷歌的工程师便开始构建各种定制化数据处理系统，以解决大规模数据处理的几大难题：**大规模数据处理特别困难**（Data Processing is hard），这里的难有多个方面，仅仅是在大规模数据上构建一个处理程序也不是一件容易的事情；保证**可伸缩性很难**（Scalability is hard），让处理程序在不同规模的集群上运行，或者更进一步，让程序根据计算资源状况自动调度执行，也不是一件容易的事情；**容错很难**（Fault-tolerance is hard），让处理程序在由廉价机器组成的集群上可靠地运行，更不是一件容易的事情。这些困难促使 MapReduce（不是 Hadoop 中的 MapReduce）诞生。MapReduce 将处理抽象成 Map+Shuffle+Reduce 的过程，这种抽象对大数据处理理论变革有着深远的影响。

以计算词频为例，MapReduce 将输入（Input）文本以行为单位分片（Split），每个 Map 任务将分片中的每个词映射为键值对的形式（Dear, 1），Shuffle 将相同键的记录组合在一起，最后由 Reduce 任务计算词频并输出（Output）结果，图 1-2 描述了一个有 3 个 Map 和 3 个 Reduce 的词频计算过程。

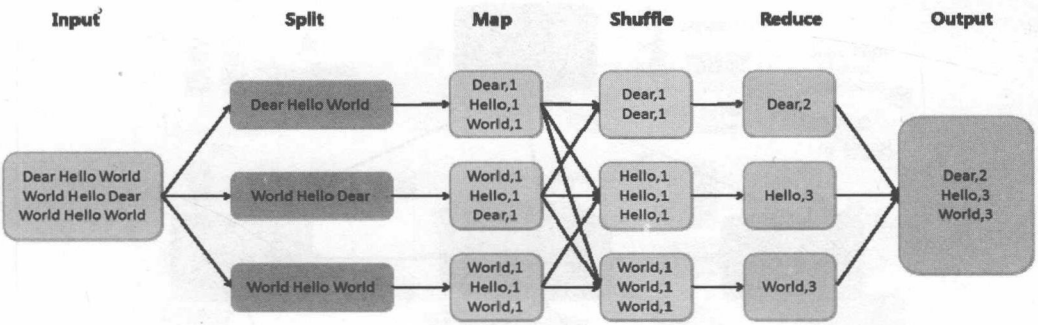

图 1-2　基于 MapReduce 计算词频的过程

笔者有一段相似的架构经历，能够帮助读者更好地理解是什么驱动谷歌的工程师开发 MapReduce 这个通用框架。驱动笔者开发一个定制化数据处理程序的想法主要来自业务需求，也有 MapReduce 思想的启发。当时，笔者就职的公司有 TB 级的短文本数据，笔者需要将这些文本的一些相邻行合并成一条记录，再对这些记录进行聚合操作，并在这之上构建一个用于语义分析的应用。出于保密要求，这些数据被**分批**归集到公司内网的一台 x86 服务器上，语义分析程序也运行在这台内网机器上。笔者有两套方案，其中一套方案使用 Hadoop，但是由于只有两台物理机器，而且用 Hadoop 有点“大炮打蚊子”的感觉，加之因着迷于 Linux 内核之美而“继承”下的“一言不合便动手造轮子”的理念，笔者决定采用另一套方案：使用 Java 语言自己动手构建一个简易的、定制化的多线程数据处理框架（类 MapReduce 数据处理框架），如图 1-3 所示。

其中，Reader 用于并行读取数据；Dealer 用于实现可级联的数据处理逻辑，如先计算记录总数，再过滤非目标记录，最后分词并计算语义标签；Writer 将 Dealer 处理的最终结果以配置的格式写入输出文件。

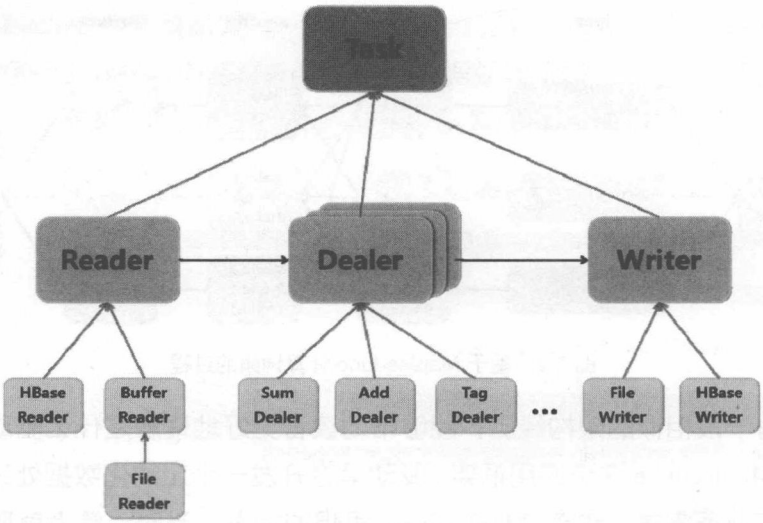

图 1-3　类 MapReduce 数据处理框架

多线程并行处理将程序运行速度提高了好几个量级。尽管如此，这段经历也令笔者回味深长：

（1）语义分析应用程序和底层组件间耦合得太紧，以至于这套软件只能由笔者维护。因为承担这个任务的部门的其他同事都是做数据分析的，没有软件开发工作经验。

（2）语义分析训练通常是相当耗时的，没有功能更强大的框架支持，手工操作的时间成本比较高。

这段经历让笔者深刻领悟到 MapReduce 框架的深思熟虑。

2004 年，Doug Cutting 和 Mike Cafarella 在构建 Nutch 时受到谷歌公司发表的 MapReduce 论文的启发，实现了开源版本的 MapReduce，即 Hadoop。此后，Pig、Hive、HBase 等工具不断涌现，Hadoop **批处理**生态系统蓬勃发展，也让人们再次领教了开源的力量，图 1-4 展示了 Hadoop 生态系统。

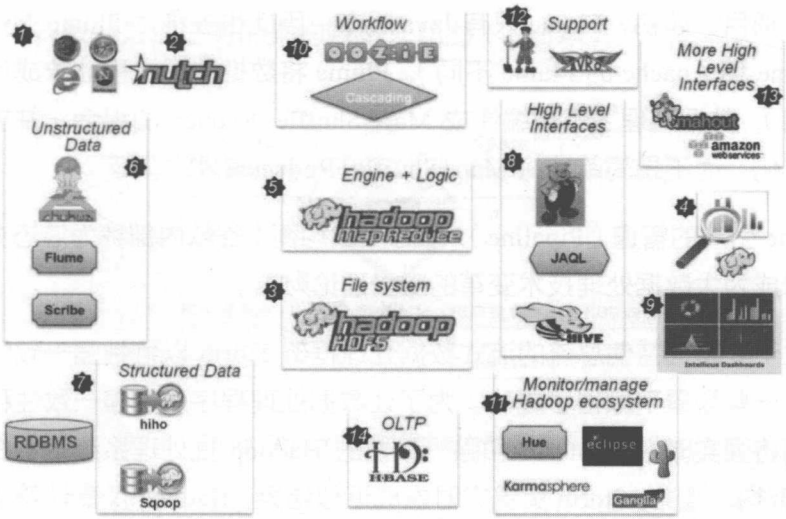

图 1-4　Hadoop 生态系统

批处理（batch）的概念由来已久。在操作系统理论中，批处理是指用户将一批作业提交给操作系统后就不再干预，由操作系统控制它们自动运行。这种操作系统被称为批处理操作系统，它是为了提高 CPU 的利用率而提出的一种操作系统。例如，在 DOS 和 Windows 系统中，我们可以在扩展名为.bat 的脚本文件中顺序定义一系列操作命令，让操作系统自动运行这些命令。

在数据处理理论中有对应的批处理系统。批处理系统的核心功能是在大容量静态数据集上运行预定义功能的、可预期完成时间的计算任务。这里的静态是指数据集是有界的，是数据集的时间属性。

流处理（streaming）系统则是构建在无界数据集之上的、可提供实时计算的另一类数据处理系统。

经过一段时间的应用实践，MapReduce 的缺陷也逐渐暴露，最让人诟病的是 Map+Shuffle+Reduce 编程模型导致计算作业效率低下。为此，2007 年，谷歌发起

了 Flume 项目。起初，Flume 只有 Java 版本，因此也被称为 Flume Java（这里所说的 Flume 和 Apache 的 Flume 不同）。Flume 将数据处理过程抽象成计算图（有向无环图），数据处理逻辑被编译成 Map+Shuffle+Reduce 的组合，并加入物理执行计划优化，而不是简单地将 Map+Shuffle+Reduce 串联。

Flume 引入的管道（Pipeline）、动态负载均衡（谷歌内部称为液态分片）和流语义思想成为大数据处理技术变革的宝贵理论财富。

产生于处理推特信息流的流式数据处理框架 Storm 以牺牲强一致性换取实时性，并在一些场景下取得了成功。为了让数据处理程序兼备强一致性和实时性，工程师们将强实时性的 Storm 和强一致性的 Hadoop 批处理系统融合在一起，即 Lambda 架构。其中，Storm 负责实时生成近似结果，Hadoop 负责计算最终精准结果。Lambda 架构需要部署两套队列集群，数据要持久化存放两份，这会导致数据冗余，增加系统维护成本。Lambda 架构示意图，如图 1-5 所示。

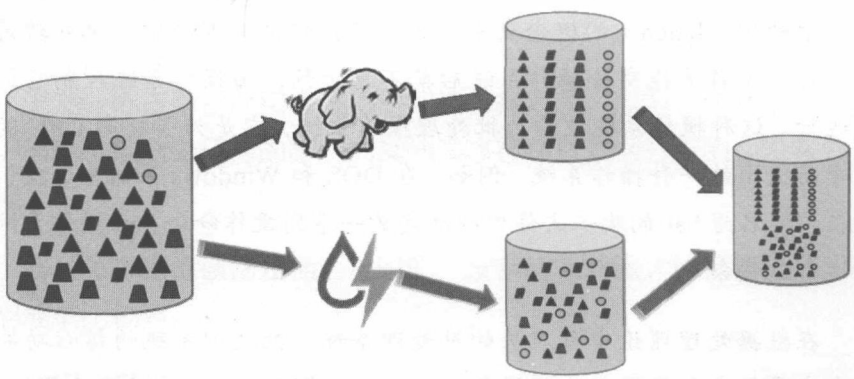

图 1-5　Lambda 架构示意图

MapReduce 模型严重依赖分布式文件系统，如 Map 将计算结果临时写入文件系统，而 Shuffle 从文件系统中读入该结果，这往往会产生较大的计算性能损耗，因此基于内存的计算是另一个选择，这就是 Spark 成功的秘诀。此外，Spark 还支持流式数据处理，即 Spark Streaming，其原理是将多个微批处理任务串接起来构建流式数据处理任务。但是这种采用微批重复运行的机制牺牲了低延迟和高吞吐的优势，引发了 Spark Streaming 是不是真正流式数据处理引擎的争议。Spark