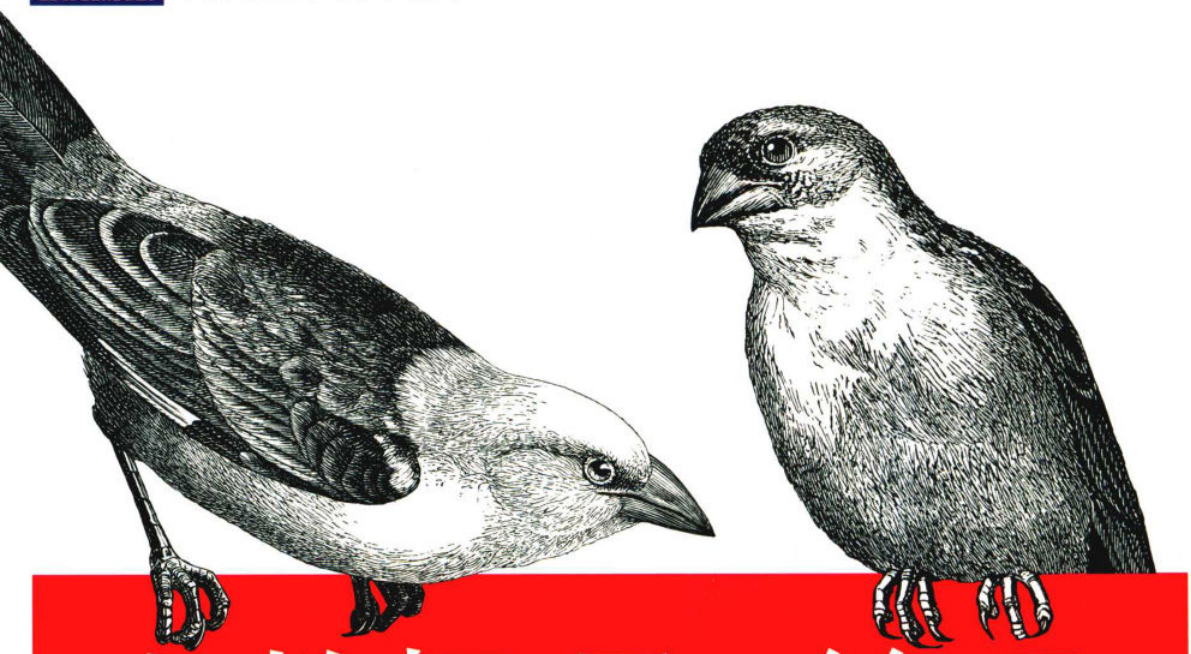
O'REILLY

TURING 图灵程序设计丛书



大数据项目管理

从规划到实现

Foundations for Architecting Data Solutions

大数据项目的“孙子兵法”，助你拥有软件开发大局观

[美] 特德·马拉斯卡　乔纳森·塞德曼 著

薛命灯 译

中国工信出版集团　人民邮电出版社 POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

大数据项目管理：从规划到实现

Foundations for Architecting Data Solutions: Managing Successful Data Projects

[美] 特德 • 马拉斯卡 [美] 乔纳森 • 塞德曼 著
薛命灯 译

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

人民邮电出版社
北　京

图书在版编目（CIP）数据

大数据项目管理 ：从规划到实现 /（美）特德·马拉斯卡（Ted Malaska），（美）乔纳森·塞德曼（Jonathan Seidman）著 ；薛命灯译. -- 北京 ：人民邮电出版社，2020.1
（图灵程序设计丛书）
ISBN 978-7-115-45736-3

Ⅰ. ①大… Ⅱ. ①特… ②乔… ③薛… Ⅲ. ①数据处理－研究 Ⅳ. ①TP274

中国版本图书馆CIP数据核字(2019)第270722号

内 容 提 要

本书提供了一个框架，从整体上介绍与大数据项目开发相关的基本概念，帮助读者评估大数据项目，理解成功的现代数据项目的基本要素。全书共 8 章，内容包括现代数据项目的主要类型、生命周期、风险管理、接口设计、分布式存储系统、元数据管理、数据处理等。本书旨在让读者厘清思路，顺利地从数据项目的规划阶段走到执行阶段，实现健壮、可维护的架构和解决方案。

本书适合首席信息官、首席运营官、技术主管、系统架构师及相关的开发人员阅读。

◆ 著 [美] 特德·马拉斯卡 [美] 乔纳森·塞德曼
译 薛命灯
责任编辑 谢婷婷
责任印制 周昇亮

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：9.75
字数：231千字 2020年1月第1版
印数：1-3 000册 2020年1月北京第1次印刷
著作权合同登记号 图字：01-2019-6607号

定价：59.00元

读者服务热线：(010)51095183转600 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

前言

既然你开始阅读本书，那么就应该知道，近几年来，数据管理领域发生了巨大的变化。我们已经看到了从第三方专有解决方案到新的开源分布式数据系统的转变。通常使用“大数据”来指代这些新的解决方案（我们发现这个词的指代作用越来越弱），但其实早期的很多专有系统也采用了可以存储和处理大量数据的分布式架构。尽管这些专有解决方案和新的开源解决方案都可以用来解决很多相同的问题，但它们之间存在一些明显的差异，这些差异促成了新系统的发展。这些差异不仅体现在开源的经济性方面，也与技术的发展有关。技术的发展促进了新系统的实现，而如果使用以前的解决方案来实现这些系统颇具挑战性。

随着这些系统的发展，出现了很多相关的书、文章、培训、会议等。这些资源可以帮助你以及这个领域的其他从业者更好地使用这些系统。那么，为什么还要再写一本与“大数据”相关的书呢？我们想说的是：不要因为一棵树而错过整片森林。这些资源大都侧重于底层的细节，例如使用 MapReduce 或 Spark 之类的分布式处理引擎来实现应用程序，或者应用高级算法来分析数据。除此之外，也有一些资源关注更高层次的架构，例如由本书作者和另外两位作者合著的《Hadoop 应用架构》[1]。

这些资源缺乏的是一个更广阔的视野，换句话说，需要采取哪些步骤来确保数据项目能够从规划阶段成功地走到执行阶段？要成功地实施数据项目，获取与架构和组件系统相关的专业知识固然重要，但其他的一些考虑因素也同样重要，而这些因素往往在探索新技术的过程中被忽视。

这些考虑因素包括：

- 理解问题；
- 选择适合用例的软件解决方案；

注 1：《Hadoop 应用架构》由人民邮电出版社出版，详见 http://iturning.cn/book/1710。——编者注

- 应对项目风险；
- 组建团队，以便成功交付项目；
- 在项目进行过程中，实现健壮、可维护的架构和解决方案。

如果你是经验丰富的软件开发人员，可能已经很熟悉这些因素了。成功管理现代数据项目的大部分流程与管理其他软件开发项目是一样的，只是在开发新的软件系统和架构时，需要一些新的知识，还需要考虑到一些额外的事项。例如，评估开源软件与选择专有解决方案有很大的不同。我们的目的不是提供又一本有关软件项目管理的书，而是指导你将行之有效的项目管理和开发实践应用到现代数据解决方案中。

读者对象

本书主要面向数据项目的决策者和实施者，例如以下角色：

- 负责高层决策的首席信息官或首席技术官；
- 负责交付数据项目的项目经理和产品经理；
- 负责开发数据项目的首席架构师、技术主管和开发人员。

再次强调，我们不打算介绍如何使用特定组件来实现应用程序；相反，我们会提供一个框架，帮助你理解成功的现代数据项目都有哪些基本要素。我们希望你能够掌握这些知识，从而成功地掌控数据项目，并做出正确的项目决策，让项目为用户带来真正的价值。

阅读方式

本书的每一章都会涉及一个与数据项目管理相关的主题。你不必从头到尾阅读整本书，因为大多数章节的内容相对独立。不过，在启动数据项目之前，先阅读第 1~3 章将大有裨益。

以下是各章的主要内容。

第 1 章，*数据项目的主要类型及考虑因素*，概述 3 种主要的数据项目用例，并针对每个用例列举需要注意的一系列考虑因素。在启动新的数据项目之前，最好先阅读这一章。

第 2 章，*评估和选择数据管理解决方案*，为在分布式开源世界中选择技术解决方案提供指导。如果你正尝试启动数据项目，或者刚刚进入这个领域，这一章对于你来说也会非常有用。

第 3 章，*数据项目的风险管理*，讨论项目风险以及如何管理它们。风险管理是软件项目的一项重要活动，大型数据项目存在一些独特的风险，要成功实现这些项目，需要管理好它们。

第 4 章，*接口设计*，讨论系统接口的设计和实现。对于创建可维护和可扩展的系统来说，定义有效的抽象和合约至关重要。因此，我们在这一章会根据自己实现大型数据项目的经验提供一些指导。

第 5 章，*分布式存储系统*，讨论分布式存储系统。数据存储是所有数据系统的核心组件，这一章将列举一些常用的分布式存储系统。更重要的是，它还会提供一个用于评估存储系统的框架。

第 6 章，*企业元数据*，讨论元数据管理。这是在构建数据系统时的另一个至关重要但经常被忽视的方面。

第 7 章，*确保数据完整性*，讨论数据的完整性问题。这是在构建数据系统时的另一个需要注意的事项，需要在项目开始时进行规划。在构建支持多种存储格式的数据系统时，确保数据的完整性和传承关系变得更具挑战性。

第 8 章，*数据处理*，讨论可用于处理分布式数据的框架。在构建有价值的数据系统时，处理和分析数据的能力是另一个重要方面。与第 5 章类似，这一章也会提供一个框架，用于了解可用的数据处理系统以及评估哪些系统适合你的应用场景。

排版约定

本书使用下列排版约定。

- **黑体字**
  表示新术语或重点强调的内容。
- 等宽字体（`constant width`）
  表示程序片段，以及正文中出现的变量、函数名、数据库、数据类型、环境变量、语句和关键字等。

该图标表示一般注记。

使用代码示例

本书是要帮你完成工作的。一般来说，如果本书提供了示例代码，你可以把它用在你的程序或文档中。除非你使用了很大一部分代码，否则无须联系我们获得许可。比如，用本书的几个代码片段写一个程序就无须获得许可，销售或分发 O'Reilly 图书的示例光盘则需要获得许可；引用本书中的示例代码回答问题无须获得许可，将书中大量的代码放到你的产品文档中则需要获得许可。

我们很希望但并不强制要求你在引用本书内容时加上引用说明。引用说明一般包括书名、作者、出版社和 ISBN，例如“*Foundations for Architecting Data Solutions* by Ted Malaska

and Jonathan Seidman (O'Reilly). Copyright 2018 Ted Malaska and Jonathan Seidman, 978-1-492-03874-0"。

如果你觉得自己对示例代码的用法超出了上述许可的范围，欢迎你通过 permissions@oreilly.com 与我们联系。

O'Reilly Safari

Safari（之前称作 Safari Books Online）是一个针对企业、政府、教育者和个人的会员制培训和参考平台。

会员可以访问来自 250 多家出版商的上千种图书、培训视频、学习路径、互动式教程和精选播放列表，这些出版商包括 O'Reilly Media、Harvard Business Review、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Adobe、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 等。

要了解更多信息，可以访问 http://www.oreilly.com/safari。

联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表[2]、示例代码以及其他信息。本书的网站地址是：http://shop.oreilly.com/product/0636920161417.do。

对于本书的评论和技术性问题，请发送电子邮件到 bookquestions@oreilly.com。

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问网站：http://www.oreilly.com。

注 2：本书中文版勘误，请到 http://ituring.cn/book/2641 查看和提交。——编者注

我们在 Facebook 的地址如下：http://facebook.com/oreilly。

请关注我们的 Twitter 动态：http://twitter.com/oreillymedia。

我们的 YouTube 视频地址如下：http://www.youtube.com/oreillymedia。

致谢

在撰写本书的过程中，很多人为我们提供了宝贵的反馈和支持，尤其是 Mark Grover、Kevin O'Dell 和 Steven Totman，他们为审阅本书内容付出了宝贵的时间。这些审阅者帮助我们提高了本书的质量，如果书中仍存在错误，都应该由我们自己负责。

我们要感谢 O'Reilly 的编辑 Nicole Tache 和 Michele Cronin。正是在她们的指导下，我们才顺利完成了本书。我们还要感谢 O'Reilly Media 的其他人提供的帮助和支持。

如果致谢清单遗漏了哪位，我们深表歉意。

电子书

扫描如下二维码，即可购买本书电子版。

目录

第1章

数据项目的主要类型及考虑因素

了解自己要构建什么，以及在设计可靠的解决方案时需要考虑哪些主要因素，这些是任何一个数据项目取得成功的基本条件。我们根据经验将数据项目分为3种类型，它们代表了大多数的数据项目。这种分类方式有助于在开始实现解决方案之前探究需要考虑的主要因素。并非每个项目都恰好属于其中一个类别，有些项目甚至可能同属多个类别。但我们认为，这些项目类型提供了一个有用的框架，帮你更好地了解数据用例。

本章首先描述主要的项目类型，然后介绍实现解决方案时需要考虑的主要事项，最后深入探讨每种项目类型的考虑因素。

1.1 数据项目的主要类型

我们先来看看数据项目的3种主要类型。

- *数据管道和数据暂存*

  可以将这类项目视为提取－转换－加载型项目，换句话说，这类项目涉及对数据集的收集、暂存、存储、建模，等等。实际上，这类项目为执行后续的数据处理和分析奠定了基础。

- *数据的处理和分析*

  这类项目最终会提供某种可用价值，可能是生成报告、创建和执行机器学习模型，等等。

- *应用程序开发*

  这类项目提供能够实时支持业务需求的数据框架，例如Web应用程序或移动应用程序的数据后端。

接下来，本章将着重关注每个项目类型的以下方面。

- 主要考虑因素

  尽管这 3 种项目类型有很多共同点，但也有一些会影响架构决策和优先级的区别，而架构决策将反过来推动项目的其余部分。在深入探讨这 3 种项目类型时，我们将首先详细介绍每个项目类型的主要考虑因素。

- 风险管理

  任何数据项目都伴随着一定的风险。我们将讨论与特定项目类型相关的潜在风险及处理方法。在很多情况下，特定场景的风险会有多种风险管理方法，因此我们需要从不同的维度进行探讨。

第 3 章将详细介绍风险管理。

- 团队组成

  为交付不同类型的项目组建团队时，需要考虑到一系列因素。不同类型的项目所需要的技能、经验和兴趣是不一样的，因此我们就每一种项目类型提供一些用于组建团队的建议。

- 安全

  安全问题可能是所有项目都会涉及的一个重要的考虑因素。安全是一个非常重要和宽泛的主题，所涉及的内容可以单独写成一本书。事实上，针对你所使用的系统，能够找到一些有用的参考资料。因为这是一个非常重要的主题，所以本书不会详细介绍，但会列出在项目过程中需要牢记的一些安全事项。

对于某些开源数据管理系统而言，安全措施更像是马后炮。这是因为早期用户更关心与存储和处理大量数据的能力相关的技术问题。此外，这些系统通常部署在内部网络中，对它们的访问是可控的。随着越来越多的企业部署这些解决方案，他们也越来越关注存储在这些系统中的数据的安全性和私密性。于是，这些项目和供应商努力做出变更和改善，以便帮助企业更好地使用这些系统。

在为项目安全做规划时，应该考虑以下维度。

- 身份验证

  确保访问系统的用户是合法的。任何成熟的系统都应该支持强身份验证，这通常可以通过 Kerberos 或轻量目录访问协议等方式来实现。

❑ 授权

在确保访问用户的合法性之后，还需要决定他们可以访问哪些数据。成熟的系统需要提供不同粒度的访问控制。例如，不仅可以提供数据库表级别的访问控制，还可以提供列级别的访问控制。在为敏感数据构建数据架构时，具备控制哪些用户和用户组可以访问哪些特定数据的能力是非常重要的。

❑ 加密

除了控制对数据的访问，出于安全方面的考虑，保护这些数据免受恶意用户和恶意入侵的影响也至关重要。数据加密是最常用的保护方法。我们需要从两个角度来考虑这个问题。

- **静止的数据**是指已经进入系统并保存在磁盘上的数据。很多数据管理供应商为此提供了解决方案，并将它们作为管理平台的一部分。一些第三方供应商也为此提供了解决方案。
- **传输中的数据**是指在系统中移动的数据。通常，供应商或项目会为此提供标准的加密机制，例如传输层安全协议。

❑ 审计

安全问题的最后一个考量维度是能够捕获与数据相关的活动，比如数据的传承关系、谁在访问数据，以及如何使用数据，等等。这个问题仍然需要通过供应商或项目提供的工具来解决。

如果安全对项目非常重要，最好的办法是找到可以解决上述 4 个问题的方案或供应商。这样一来，就可以减少花在数据安全性管理方面的时间，而将更多的时间用于解决其他问题。

1.2 数据管道和数据暂存

我们从 3 个数据项目类型中范围最广的开始讨论，因为它涉及从外部数据源到目标数据源的整个路径，并为构建数据解决方案的其余部分奠定了基础。

对于这个项目类型，在设计解决方案时需要考虑以下因素：

- 针对目标数据将执行哪些类型的查询和处理；
- 客户的数据要求；
- 已收集数据的类型。

考虑到这些数据在后续处理和分析中的重要性，我们在建模和存储这些数据时要十分谨慎，为后续的数据访问提供便利。

1.2.1 主要考虑因素和风险管理

对于数据管道和数据暂存项目，有以下主要考虑因素：

- 源数据消费；
- 数据传递保证；
- 数据的管理和治理；
- 延迟和传递确认；
- 目标数据的访问模式。

接下来，我们将逐个介绍这些考虑因素以及每个因素的属性会如何影响项目的优先级。

1. 源数据消费

当我们说到数据源时，基本上是指那些生成数据的系统，它们为我们构建的数据解决方案提供必要的数据。数据源可以是手机、传感器、应用程序、机器日志、操作型数据库和事务型数据库，等等。数据源大都位于数据管道和数据暂存系统之外。实际上，你可以根据花费在与数据源团队合作上的时间来评估系统的成功程度。数据工程团队在数据源集成上花费的时间通常与数据源集成设计的优劣成反比。

可以使用一些标准的方法收集源数据。

❑ 嵌入式代码

你可以为源系统提供代码，将它们嵌入到源系统中，这些代码知道如何将必要的数据发送到你的数据管道中。

❑ 代理

这是一个非常靠近数据源的独立系统，大多数情况下与数据源位于同一设备上。与嵌入式代码不同，代理是作为单独的进程运行的，而且没有依赖项。

❑ 接口

这是最轻量级的方式，例如 REST 和用于接收源数据的 WebSocket 端点。

当然，除了这些，还有其他一些常用的数据收集方式：

- 第三方数据集成工具，可以是开源的，也可以是商用的；
- 批量数据摄取工具，例如 Apache Sqoop 和特定项目提供的工具（如 Hadoop 分布式文件系统提供的 put 命令）。

你可以根据实际的用例选择工具，它们可以帮你更好地构建数据管道。因为其他参考资料以及供应商和项目的文档已经详细介绍了它们，所以本书不再赘述。

哪种方法最好？答案通常取决于数据来源。但在某些情况下，可能几种方法都适用，关键

是要确保正确地使用这些方法。因此，我们将讨论与不同数据收集类型相关的一些注意事项。先从嵌入式代码开始讲。

嵌入式代码

在使用嵌入式代码收集源数据时，需要考虑以下准则。

- 限制编程语言的使用

  不要试图支持多种编程语言，而应该先使用一种语言实现，然后为其他语言提供绑定。例如，使用 C、C++ 或 Java 实现，然后为需要支持的其他语言创建绑定。以 Kafka 为例，Kafka 的核心项目提供了 Java 版本的生产者和消费者，而其他语言的库或客户端需要绑定到 Kafka 提供的库，这些库是 Kafka 发行包的一部分。

- 限制依赖项的使用

  任何嵌入式代码都存在潜在的库冲突。限制依赖项的使用有助于缓解这个问题。

- 提供可见性

  人们可能会关注嵌入式代码中究竟包含了哪些内容，所以需要通过开源或将代码放在公开代码库中来提供嵌入式代码的可见性，这是一种简单而安全的方式。用户可以看到所有的代码，进而减轻对某些潜在问题（如内存使用、网络使用等）的担忧。

- 运维问题

  还有一个考虑因素是嵌入式代码可能会在生产环境中造成哪些问题。确保你已经考虑到了内存泄漏或性能问题，并定义了用于解决这些问题的支持模型。日志和代码插桩有助于在发生故障时找出问题。

- 版本管理

  在使用嵌入式代码时，你可能无法控制代码的更新。这个时候，确保向后兼容并定义良好的版本就显得非常重要。

代理

在架构中使用代理时，请注意以下事项。

- 部署

  与架构中的其他组件一样，请确保代理的部署是经过测试的，并且是可重复的。这可能需要使用某种自动化工具或容器。

- 资源使用情况

  确保源系统拥有足够的资源来支持代理进程的运行，包括内存、CPU 等。

- 隔离

  虽然代理在应用程序外部运行，但仍然需要防止代理对数据收集带来负面影响。