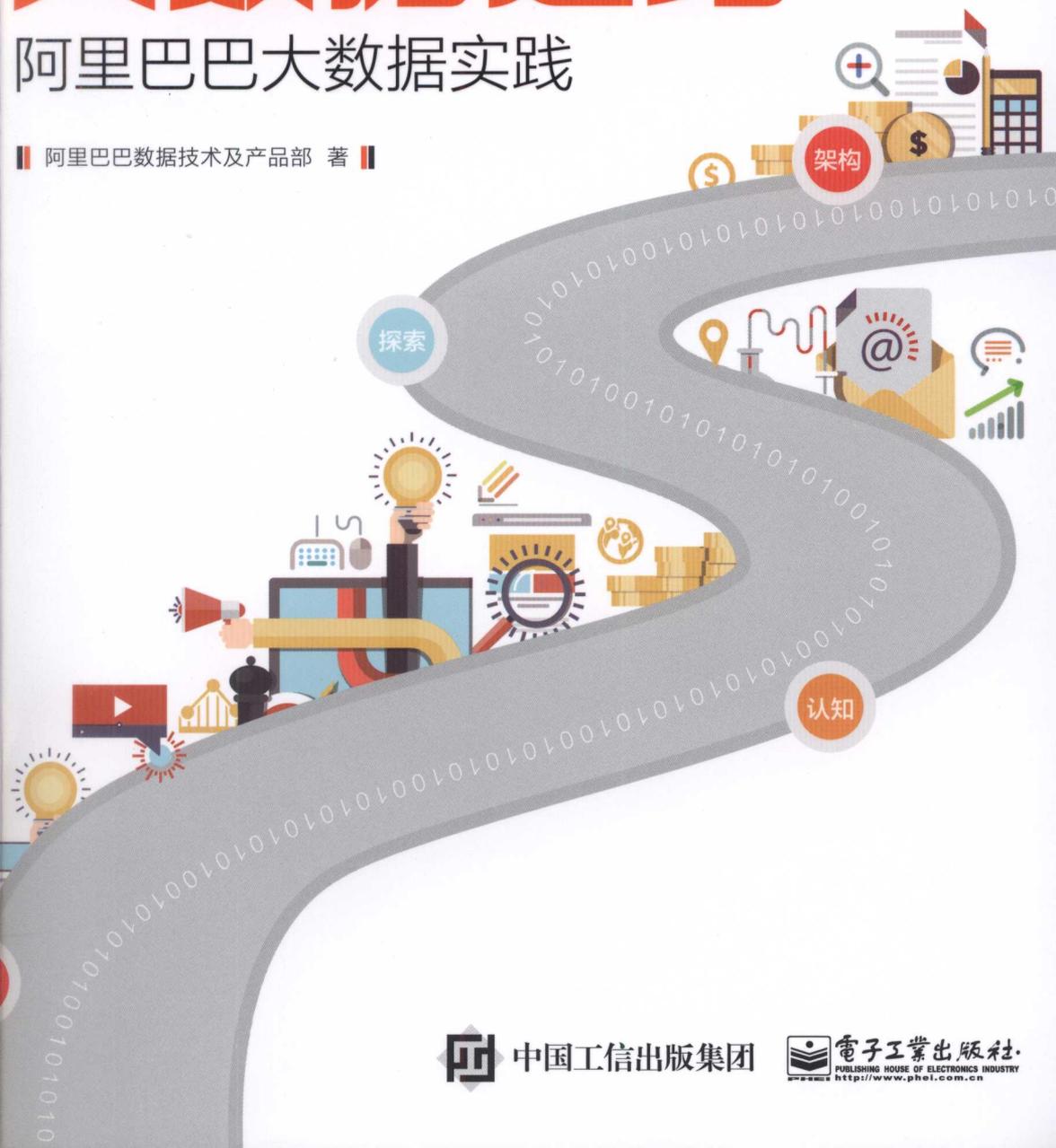


大数据之路

阿里巴巴大数据实践

|| 阿里巴巴数据技术及产品部 著 ||



阿里巴巴数据技术及产品部

定位于阿里集团数据中台，为阿里生态内外的业务、用户、中小企业提供全链路、全渠道的数据服务。作为阿里大数据战略的核心践行者，致力于“让大数据赋能商业，创造价值”。

经过多年的实践，数据技术及产品部已经构建了从底层的数据采集、数据处理，到挖掘算法、数据应用服务以及数据产品的全链路、标准化的大数据体系。通过这个体系，超过EB级别的海量数据能够高效融合，并以秒级的响应速度，服务并驱动阿里巴巴自身的业务和外部千万用户的发展。

现在，阿里巴巴数据技术及产品部正通过技术和产品上的创新，探索全域数据的价值，将阿里在大数据上沉淀的能力对外分享，为各行各业的发展带来更多可能性。



阿里数据官网



阿里数据
微信公众号

大数据之路

阿里巴巴大数据实践

|| 阿里巴巴数据技术及产品部 著 ||

電子工業出版社·

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

在阿里巴巴集团内，数据人员面临的现实情况是：集团数据存储已经达到 EB 级别，部分单张表每天的数据记录数高达几千亿条；在 2016 年“双 11 购物狂欢节”的 24 小时中，支付金额达到了 1207 亿元人民币，支付峰值高达 12 万笔/秒，下单峰值达 17.5 万笔/秒，媒体直播大屏处理的总数据量高达百亿级别且所有数据都需要做到实时、准确地对外披露……巨大的信息量给数据采集、存储和计算都带来了极大的挑战。

《大数据之路——阿里巴巴大数据实践》就是在此背景下完成的。本书中讲到的阿里巴巴大数据系统架构，就是为了满足不断变化的业务需求，同时实现系统的高度扩展性、灵活性以及数据展现的高性能而设计的。

本书由阿里巴巴数据技术及产品部组织并完成写作，是阿里巴巴分享对大数据的认知，与生态伙伴共创数据智能的重要基石。相信本书中的实践和思考对同行会有很大的启发和借鉴意义。

本书著作权归淘宝（中国）软件有限公司所有，未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

大数据之路：阿里巴巴大数据实践 / 阿里巴巴数据技术及产品部著. —北京：电子工业出版社，2017.7

ISBN 978-7-121-31438-4

I. ①大… II. ①阿… III. ①企业管理—数据管理 IV. ①F272.7

中国版本图书馆 CIP 数据核字（2017）第 094934 号

策划编辑：张彦红

责任编辑：葛娜

印 刷：三河市双峰印刷装订有限公司

装 订：三河市双峰印刷装订有限公司

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：720×1000 1/16 印张：21 字数：338 千字

版 次：2017 年 7 月第 1 版

印 次：2017 年 7 月第 1 次印刷

印 数：4000 册 定价：79.00 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 zltz@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。

本书编委会

编委会主任：朋新宇

主 编：王赛、王永伟

编委会成员（排名不分先后）：

罗金鹏、王静、王赛、王永伟、张磊、赵唯行

本书内容撰写人员（排名不分先后）：

总体架构：王俊华、王永伟

日志采集：李涛、殷霞

数据同步：陈永俊、王俊华

离线数据开发：陈永俊、王永伟

实时技术：黄晓锋

数据服务：方建江、郭才祥、江岚、徐锐

数据挖掘：何覃、王鹏、王中要、徐萧萧、应倩岚、
郑苏杭

大数据领域建模综述：王赛

阿里巴巴数据整合及管理體系：张子良、王永伟

维度设计：王永伟

事实表设计：陈永俊、方彬、王永伟、张子良

元数据：魏海、王永伟

计算管理：陈中强、贾元乔、孔令娟、袁杰

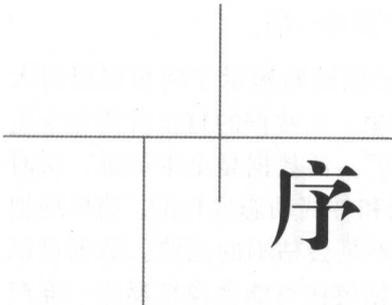
存储和成本管理：潘旻、王伟

数据质量：方彬

数据应用：肖美丽、郑育杰

特别感谢（排名不分先后）：

陈同杰、邓中华、丁燕、何露莎、黄荣、金高平、刘凡、李炉阳、刘健男、林鸣晖、李启平、邱坤东、梅婧婷、孙伟光、苏艳、王政、王子凌、向师富、姚滨晖、杨红霞、张启、张伟、阮城锋、谢丹丹、商渭清、裴逸钧、罗鸣、王鹏、冯敏敏、曾文秋等对本书出版工作的支持！



序

大数据是什么？在过去的5年里，恐怕没有另外一个词比大数据更高频；也没有另外一个概念如大数据一样，被纷繁解读，著书立说。有趣的是，作为距离大数据最近的公司之一——尽管我们的初心或许和大数据没有直接关系——在关于大数据的理论和概念的争论中，阿里巴巴却鲜有高谈阔论。

因为自知而敬畏，因为敬畏而谦逊。甚至在大数据这个概念出现很久之前，阿里巴巴就不得不直面、认知、探索，并架构和大数据有关的一切。数据作为一个生态级的平台企业最直接的沉淀，亦是最基本的再生产资料。如果没有基于大数据的人工智能的应用，淘宝根本不可能面对每天亿级的用户访问数量。因此，仅仅因为本能，阿里巴巴一开始就自然生长在这样一个数据的黑洞中，并且被越来越多、越来越密集的数据风暴裹挟。阿里巴巴在大数据方面所做的各种艰苦努力，其实就是力图对抗这种无序和复杂的熵增，从中梳理结构，提炼价值。

这是一个历经磨炼、也卓有成效的长期过程。如书中所提到的，阿里巴巴不仅数据量超宇宙级，而且更是因为业务场景的复杂和多元化，其面对着甚至超过 Google 和 Facebook 的更复杂的难题。大部分时候，阿里巴巴都是在无人区艰难跋涉。每一组功能和逻辑，每一套架构与系统，都与业务和场景息息相关。这个黑洞膨胀之快，以至于大部分时候都是在出现痛点从而刺激了架构升级。换言之，大数据系统——如果我们非要用一个系统去描述的话——其复杂度之高，是几乎不可能在一开始就完整和完美地进行自上而下定义和设计的。从需求→设计→迭代→

升华为理论，在无数次的迭代进化中，我们对大数据的理解才逐渐成形，慢慢能够在将数据黑洞为我所用的抗争中扳回一局。

这个系统生长和进化的过程实际上已经暗暗揭示了阿里巴巴对大数据真髓的理解。大、快、多样性只是表象，大数据的真正价值在于生命性和生态性。阿里巴巴称之为“活数据”。活数据是全本记录、实时驱动决策和迭代，其价值是随着使用场景和方式动态变化的。简单地把数据定义为正/负资产都太简单。数据也不是会枯竭的能源。数据可以被重复使用，并在使用中升值；数据与数据链接可能会像核反应一样产生价值的聚变。数据使用和数据聚变又产生新的数据。活数据的基础设施就需要来承载、管理和促进这个生态体的最大价值实现（以及相应的成本最小化）。丰富的数据形式、多样化的参与角色和动机，以及迥异的计算场景都使得这个系统的复杂度无限升级。阿里巴巴的大数据之路就是在深刻理解这种复杂性的基础上，摸索到了一些重要的秩序和原理，并通过技术架构来验证和夯实。

如果说互联网实现了人人互联和通信，并没有深度地协同计算，那么这样的一个大数据平台和架构就是一张升级的、智能的互联网。这是人类自己设计出来的复杂的信息处理系统，同时也将是真正意义上人类智力大联合的基础设施。这是一个伟大的蓝图，我们敬畏其复杂度和潜能。《大数据之路——阿里巴巴大数据实践》便是阿里巴巴分享对大数据的认知、与世界共创数据智能的重要基石。数据技术及产品部作为阿里巴巴集团的数据中台，一直致力为阿里巴巴集团内、外提供大数据方面的系统服务，承载了阿里巴巴集团大数据梦想至关重要的数据平台建设。相信他们的实践和思考对同行会有很大的启发和借鉴意义。

曾鸣教授

阿里巴巴集团学术委员会主席

湖畔大学教务长

2017年5月

目 录

第 1 章 总述	1
第 1 篇 数据技术篇	
第 2 章 日志采集	8
2.1 浏览器的页面日志采集	8
2.1.1 页面浏览日志采集流程	9
2.1.2 页面交互日志采集	14
2.1.3 页面日志的服务器端清洗和预处理	15
2.2 无线客户端的日志采集	16
2.2.1 页面事件	17
2.2.2 控件点击及其他事件	18
2.2.3 特殊场景	19
2.2.4 H5 & Native 日志统一	20
2.2.5 设备标识	22
2.2.6 日志传输	23
2.3 日志采集的挑战	24
2.3.1 典型场景	24
2.3.2 大促保障	26
第 3 章 数据同步	29
3.1 数据同步基础	29
3.1.1 直连同步	30

3.1.2	数据文件同步	30
3.1.3	数据库日志解析同步	31
3.2	阿里数据仓库的同步方式	35
3.2.1	批量数据同步	35
3.2.2	实时数据同步	37
3.3	数据同步遇到的问题与解决方案	39
3.3.1	分库分表的处理	39
3.3.2	高效同步和批量同步	41
3.3.3	增量与全量同步的合并	42
3.3.4	同步性能的处理	43
3.3.5	数据漂移的处理	45
第4章	离线数据开发	48
4.1	数据开发平台	48
4.1.1	统一计算平台	49
4.1.2	统一开发平台	53
4.2	任务调度系统	58
4.2.1	背景	58
4.2.2	介绍	59
4.2.3	特点及应用	65
第5章	实时技术	68
5.1	简介	69
5.2	流式技术架构	71
5.2.1	数据采集	72
5.2.2	数据处理	74
5.2.3	数据存储	78
5.2.4	数据服务	80
5.3	流式数据模型	80
5.3.1	数据分层	80
5.3.2	多流关联	83
5.3.3	维表使用	84
5.4	大促挑战&保障	86

5.4.1	大促特征	86
5.4.2	大促保障	88
第 6 章	数据服务	91
6.1	服务架构演进	91
6.1.1	DWSOA	92
6.1.2	OpenAPI	93
6.1.3	SmartDQ	94
6.1.4	统一的数据服务层	96
6.2	技术架构	97
6.2.1	SmartDQ	97
6.2.2	iPush	100
6.2.3	Lego	101
6.2.4	uTiming	102
6.3	最佳实践	103
6.3.1	性能	103
6.3.2	稳定性	111
第 7 章	数据挖掘	116
7.1	数据挖掘概述	116
7.2	数据挖掘算法平台	117
7.3	数据挖掘中台体系	119
7.3.1	挖掘数据中台	120
7.3.2	挖掘算法中台	122
7.4	数据挖掘案例	123
7.4.1	用户画像	123
7.4.2	互联网反作弊	125
第 2 篇 数据模型篇		
第 8 章	大数据领域建模综述	130
8.1	为什么需要数据建模	130
8.2	关系数据库系统和数据仓库	131

8.3	从 OLTP 和 OLAP 系统的区别看模型方法论的选择	132
8.4	典型的数据仓库建模方法论	132
8.4.1	ER 模型	132
8.4.2	维度模型	133
8.4.3	Data Vault 模型	134
8.4.4	Anchor 模型	135
8.5	阿里巴巴数据模型实践综述	136
第 9 章	阿里巴巴数据整合及管理体系	138
9.1	概述	138
9.1.1	定位及价值	139
9.1.2	体系架构	139
9.2	规范定义	140
9.2.1	名词术语	141
9.2.2	指标体系	141
9.3	模型设计	148
9.3.1	指导理论	148
9.3.2	模型层次	148
9.3.3	基本原则	150
9.4	模型实施	152
9.4.1	业界常用的模型实施过程	152
9.4.2	OneData 实施过程	154
第 10 章	维度设计	159
10.1	维度设计基础	159
10.1.1	维度的基本概念	159
10.1.2	维度的基本设计方法	160
10.1.3	维度的层次结构	162
10.1.4	规范化和反规范化	163
10.1.5	一致性维度和交叉探查	165
10.2	维度设计高级主题	166
10.2.1	维度整合	166
10.2.2	水平拆分	169

10.2.3	垂直拆分	170
10.2.4	历史归档	171
10.3	维度变化	172
10.3.1	缓慢变化维	172
10.3.2	快照维表	174
10.3.3	极限存储	175
10.3.4	微型维度	178
10.4	特殊维度	180
10.4.1	递归层次	180
10.4.2	行为维度	184
10.4.3	多值维度	185
10.4.4	多值属性	187
10.4.5	杂项维度	188
第 11 章	事实表设计	190
11.1	事实表基础	190
11.1.1	事实表特性	190
11.1.2	事实表设计原则	191
11.1.3	事实表设计方法	193
11.2	事务事实表	196
11.2.1	设计过程	196
11.2.2	单事务事实表	200
11.2.3	多事务事实表	202
11.2.4	两种事实表对比	206
11.2.5	父子事实的处理方式	208
11.2.6	事实的设计准则	209
11.3	周期快照事实表	210
11.3.1	特性	211
11.3.2	实例	212
11.3.3	注意事项	217
11.4	累积快照事实表	218
11.4.1	设计过程	218
11.4.2	特点	221

11.4.3	特殊处理	223
11.4.4	物理实现	225
11.5	三种事实表的比较	227
11.6	无事实的事实表	228
11.7	聚集型事实表	228
11.7.1	聚集的基本原则	229
11.7.2	聚集的基本步骤	229
11.7.3	阿里公共汇总层	230
11.7.4	聚集补充说明	234
第 3 篇 数据管理篇		
第 12 章	元数据	236
12.1	元数据概述	236
12.1.1	元数据定义	236
12.1.2	元数据价值	237
12.1.3	统一元数据体系建设	238
12.2	元数据应用	239
12.2.1	Data Profile	239
12.2.2	元数据门户	241
12.2.3	应用链路分析	241
12.2.4	数据建模	242
12.2.5	驱动 ETL 开发	243
第 13 章	计算管理	245
13.1	系统优化	245
13.1.1	HBO	246
13.1.2	CBO	249
13.2	任务优化	256
13.2.1	Map 倾斜	257
13.2.2	Join 倾斜	261
13.2.3	Reduce 倾斜	269

第 14 章 存储和成本管理	275
14.1 数据压缩	275
14.2 数据重分布	276
14.3 存储治理项优化	277
14.4 生命周期管理	278
14.4.1 生命周期管理策略	278
14.4.2 通用的生命周期管理矩阵	280
14.5 数据成本计量	283
14.6 数据使用计费	284
第 15 章 数据质量	285
15.1 数据质量保障原则	285
15.2 数据质量方法概述	287
15.2.1 消费场景知晓	289
15.2.2 数据加工过程卡点校验	292
15.2.3 风险点监控	295
15.2.4 质量衡量	299
第 4 篇 数据应用篇	
第 16 章 数据应用	304
16.1 生意参谋	305
16.1.1 背景概述	305
16.1.2 功能架构与技术能力	307
16.1.3 商家应用实践	310
16.2 对内数据产品平台	313
16.2.1 定位	313
16.2.2 产品建设历程	314
16.2.3 整体架构介绍	317
附录 A 本书插图索引	320

轻松注册成为博文视点社区用户 (www.broadview.com.cn), 扫码直达本书页面。

- **提交勘误:** 您对书中内容的修改意见可在 [提交勘误](#) 处提交, 若被采纳, 将获赠博文视点社区积分 (在您购买电子书时, 积分可用来抵扣相应金额)。
- **交流互动:** 在页面下方 [读者评论](#) 处留下您的疑问或观点, 与我们和其他读者一同学习交流。

页面入口: <http://www.broadview.com.cn/31438>



第 1 章

总述

2014 年，马云提出，“人类正从 IT 时代走向 DT 时代”。如果说在 IT 时代是以自我控制、自我管理为主，那么到了 DT (Data Technology) 时代，则是以服务大众、激发生产力为主。以互联网（或者物联网）、云计算、大数据和人工智能为代表的新技术革命正在渗透至各行各业，悄悄地改变着我们的生活。

在 DT 时代，人们比以往任何时候更能收集到更丰富的数据。IDC 的报告显示：预计到 2020 年，全球数据总量将超过 40ZB（相当于 40 万亿 GB），这一数据量是 2011 年的 22 倍！正在呈“爆炸式”增长的数据，其潜在的巨大价值有待发掘。数据作为一种新的能源，正在发生聚变，变革着我们的生产和生活，催生了当下大数据行业发展热火朝天的盛景。

但是如果不能对这些数据进行有序、有结构地分类组织和存储，如果不能有效利用并发掘它，继而产生价值，那么它同时也成为一场“灾难”。无序、无结构的数据犹如堆积如山的垃圾，给企业带来的是令人咋舌的高额成本。

在阿里巴巴集团内，我们面临的现实情况是：集团数据存储达到

EB 级别，部分单张表每天的数据记录数高达几千亿条；在 2016 年“双 11 购物狂欢节”的 24 小时中，支付金额达到了 1207 亿元人民币，支付峰值高达 12 万笔/秒，下单峰值达 17.5 万笔/秒，媒体直播大屏处理的总数据量高达百亿且所有数据都需要做到实时、准确地对外披露……这些给数据采集、存储和计算都带来了极大的挑战。

在阿里内部，数据工程师每天要面对百万级规模的离线数据处理工作。阿里大数据井喷式的爆发，加大了数据模型、数据研发、数据质量和运维保障工作的难度。

同时，日益丰富的业态，也带来了各种各样、纷繁复杂的数据需求。如何有效地满足来自员工、商家、合作伙伴等多样化的需求，提高他们对数据使用的满意度，是数据服务和数据产品需要面对的挑战。

如何建设高效的数据模型和体系，使数据易用，避免重复建设和数据不一致性，保证数据的规范性；如何提供高效易用的数据开发工具；如何做好数据质量保障；如何有效管理和控制日益增长的存储和计算消耗；如何保证数据服务的稳定，保证其性能；如何设计有效的数据产品高效赋能于外部客户和内部员工……这些都给大数据系统的建设提出了更多复杂的要求。

本书介绍的阿里巴巴大数据系统架构，就是为了满足不断变化的业务需求，同时实现系统的高度扩展性、灵活性以及数据展现的高性能而设计的。

如图 1.1 所示是阿里巴巴大数据系统体系架构图，从图中可以清晰地看到数据体系主要分为数据采集、数据计算、数据服务和数据应用四大层次。

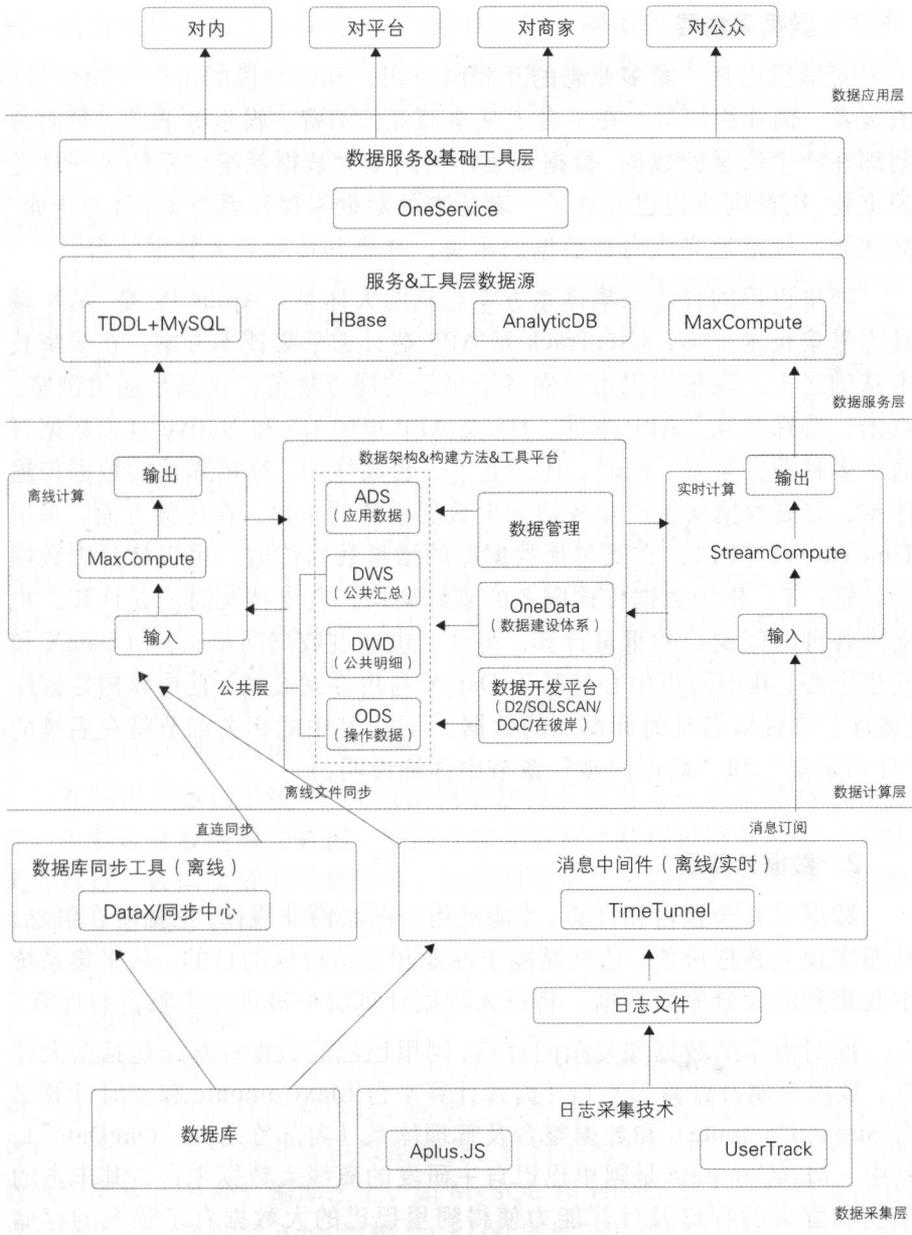


图 1.1 阿里巴巴大数据系统体系架构图

1. 数据采集层

阿里巴巴是一家多业态的互联网公司，几亿规模的用户（如商家、消费者、商业组织等）在平台上从事商业、消费、娱乐等活动，每时每刻都在产生海量的数据，数据采集作为阿里大数据系统体系的第一环尤为重要。因此阿里巴巴建立了一套标准的数据采集体系方案，致力全面、高性能、规范地完成海量数据的采集，并将其传输到大数据平台。

阿里巴巴的日志采集体系方案包括两大体系：Aplus.JS 是 Web 端日志采集技术方案；UserTrack 是 APP 端日志采集技术方案。在采集技术基础之上，阿里巴巴用面向各个场景的埋点规范，来满足通用浏览、点击、特殊交互、APP 事件、H5 及 APP 里的 H5 和 Native 日志数据打通等多种业务场景。同时，还建立了一套高性能、高可靠性的数据传输体系，完成数据从生产业务端到大数据系统的传输。在传输方面，采用 TimeTunnel (TT)，它既包括数据库的增量数据传输，也包括日志数据的传输；TT 作为数据传输服务的基础架构，既支持实时流式计算，也支持各种时间窗口的批量计算。另外，也通过数据同步工具（DataX 和同步中心，其中同步中心是基于 DataX 易用性封装的）直连异构数据库（备库）来抽取各种时间窗口的数据。（注：其中的相关细节将在后续的“日志采集”和“数据同步”章节中详细说明。）

2. 数据计算层

数据只有被整合和计算，才能被用于洞察商业规律，挖掘潜在信息，从而实现大数据价值，达到赋能于商业和创造价值的目的。从采集系统中收集到的大量原始数据，将进入数据计算层中被进一步整合与计算。

面对海量的数据和复杂的计算，阿里巴巴的数据计算层包括两大体系：数据存储及计算云平台（离线计算平台 MaxCompute 和实时计算平台 StreamCompute）和数据整合及管理體系（内部称之为“OneData”）。其中，MaxCompute 是阿里巴巴自主研发的离线大数据平台，其丰富的功能和强大的存储及计算能力使得阿里巴巴的大数据有了强大的存储和计算引擎；StreamCompute 是阿里巴巴自主研发的流式大数据平台，在内部较好地支持了阿里巴巴流式计算需求；OneData 是数据整合及管理的方法体系和工具（注：为方便内部工作及沟通，在阿里内部将这一

统一的方法体系和工具简称为“OneData”), 阿里巴巴的大数据工程师在这一体系下, 构建统一、规范、可共享的全域数据体系, 避免数据的冗余和重复建设, 规避数据烟囱和不一致性, 充分发挥阿里巴巴在大数据海量、多样性方面的独特优势。借助这一统一化数据整合及管理的方法体系, 我们构建了阿里巴巴的数据公共层, 并可以帮助相似大数据项目快速落地实现。

从数据计算频率角度来看, 阿里数据仓库可以分为离线数据仓库和实时数据仓库。离线数据仓库主要是指传统的数据仓库概念, 数据计算频率主要以天(包含小时、周和月)为单位; 如 T-1, 则每天凌晨处理上一天的数据。但是随着业务的发展特别是交易过程的缩短, 用户对数据产出的实时性要求逐渐提高, 所以阿里的实时数据仓库应运而生。“双11”实时数据直播大屏, 就是实时数据仓库的一种典型应用。

阿里数据仓库的数据加工链路也是遵循业界的分层理念, 包括操作数据层(Operational Data Store, ODS)、明细数据层(Data Warehouse Detail, DWD)、汇总数据层(Data Warehouse Summary, DWS)和应用数据层(Application Data Store, ADS)。通过数据仓库不同层次之间的加工过程实现从数据资产向信息资产的转化, 并且对整个过程进行有效的元数据管理及数据质量处理。

在阿里大数据系统中, 元数据模型整合及应用是一个重要的组成部分, 主要包含数据源元数据、数据仓库元数据、数据链路元数据、工具类元数据、数据质量类元数据等。元数据应用主要面向数据发现、数据管理等, 如用于存储、计算和成本管理等。

3. 数据服务层

当数据已被整合和计算好之后, 需要提供给产品和应用进行数据消费。为了有更好的性能和体验, 阿里巴巴构建了自己的数据服务层, 通过接口服务化方式对外提供数据服务。针对不同的需求, 数据服务层的数据源架构在多种数据库之上, 如 MySQL 和 HBase 等。后续将逐渐迁移至阿里云数据库 ApsaraDB for RDS(简称“RDS”)和表格存储(Table Store)等。

数据服务可以使应用对底层数据存储透明, 将海量数据方便高效地

开放给集团内部各应用使用。现在，数据服务每天拥有几十亿的数据调用量，如何在性能、稳定性、扩展性等方面更好地服务于用户；如何满足应用各种复杂的数据服务需求；如何保证“双 11”媒体大屏数据服务接口的高可用……随着业务的发展，需求越来越复杂，因此数据服务也在不断地前进。

数据服务层对外提供数据服务主要是通过统一的数据服务平台（为方便阅读，简称为“OneService”）。OneService 以数据仓库整合计算好的数据作为数据源，对外通过接口的方式提供数据服务，主要提供简单数据查询服务、复杂数据查询服务（承接集团用户识别、用户画像等复杂数据查询服务）和实时数据推送服务三大特色数据服务。

4. 数据应用层

数据已经准备好，需要通过合适的应用提供给用户，让数据最大化地发挥价值。阿里对数据的应用表现在各个方面，如搜索、推荐、广告、金融、信用、保险、文娱、物流等。商家，阿里内部的搜索、推荐、广告、金融等平台，阿里内部的运营和管理人员等，都是数据应用方；ISV、研究机构和社会组织等也可以利用阿里开放的数据能力和技术。

阿里巴巴基于数据的应用产品有很多，本书选择了服务于阿里内部员工的阿里数据平台和服务于商家的对外数据产品——生意参谋进行基础性介绍。其他数据应用不再赘述。对内，阿里数据平台产品主要有实时数据监控、自助式的数据网站或产品构建的数据小站、宏观决策分析支撑平台、对象分析工具、行业数据分析门户、流量分析平台等。

我们相信，数据作为新能源，为产业注入的变革是显而易见的。我们对数据新能源的探索也不仅仅停留在狭义的技术、服务和应用上。我们正在挖掘大数据更深层次的价值，为社会经济和民生基础建设等提供创新方法。

注：本书中出现的专有名词、专业术语、产品名称、软件项目名称、工具名称等，是淘宝（中国）软件有限公司内部项目的惯用词语，如与第三方名称雷同，实属巧合。

第1篇

数据技术篇

- 第2章 日志采集
- 第3章 数据同步
- 第4章 离线数据开发
- 第5章 实时技术
- 第6章 数据服务
- 第7章 数据挖掘

第2章

日志采集

数据采集作为阿里大数据系统体系的第一环尤为重要。因此阿里巴巴建立了一套标准的数据采集体系方案，致力全面、高性能、规范地完成海量数据的采集，并将其传输到大数据平台。本章主要介绍数据采集中的日志采集部分。

阿里巴巴的日志采集体系方案包括两大体系：Aplus.JS 是 Web 端（基于浏览器）日志采集技术方案；UserTrack 是 APP 端（无线客户端）日志采集技术方案。

本章从浏览器的页面日志采集、无线客户端的日志采集以及我们遇到的日志采集挑战三块内容来阐述阿里巴巴的日志采集经验。

2.1 浏览器的页面日志采集

浏览器的页面型产品/服务的日志采集可分为如下两大类。

(1) 页面浏览（展现）日志采集。顾名思义，页面浏览日志是指当一个页面被浏览器加载呈现时采集的日志。此类日志是最基础的互联网

日志，也是目前所有互联网产品的两大基本指标：页面浏览量（Page View, PV）和访客数（Unique Visitors, UV）的统计基础。页面浏览日志是目前成熟度和完备度最高，同时也是最具挑战性的日志采集任务，我们将重点讲述此类日志的采集。

(2) 页面交互日志采集。当页面加载和渲染完成之后，用户可以在页面上执行各类操作。随着互联网前端技术的不断发展，用户可在浏览器内与网页进行的互动已经丰富到只有想不到没有做不到的程度，互动设计都要求采集用户的互动行为数据，以便通过量化获知用户的兴趣点或者体验优化点。交互日志采集就是为此类业务场景而生的。

除此之外，还有一些专门针对某些特定统计场合的日志采集需求，如专门采集特定媒体在页面被曝光状态的曝光日志、用户在线状态的实时监测等，但在基本原理上都脱胎于上述两大类。限于篇幅，此内容在本书中就不予展开介绍了。

2.1.1 页面浏览日志采集流程

网站页面是互联网服务的基本载体，即使在如今传统互联网形态逐渐让位于移动互联网的背景下，HTML 页面依旧是最普遍的业务形态，对于以网页为基本展现形式的互联网产品和服务，衡量其业务水平的基本指标是网页浏览量（PV）和访客数（UV）。为此，我们需要采集页面被浏览器加载展现的记录，这是最原始的互联网日志采集需求，也是一切互联网数据分析得以展开的基础和前提。

目前典型的网页访问过程是以浏览器请求、服务器响应并返回所请求的内容（大多以 HTML 文档的形式）这种模式进行的，浏览器和服务器之间的通信普遍遵守 HTTP 协议（超文本传输协议，目前以 HTTP 1.1 为主，逐渐向最新的 HTTP 2.0 过渡）。浏览器发起的请求被称为 HTTP 请求（HTTP Request），服务器的返回则被称为 HTTP 响应（HTTP Response）。

我们以用户访问淘宝首页（www.taobao.com）为例，一次典型的页面访问过程描述如图 2.1 所示。

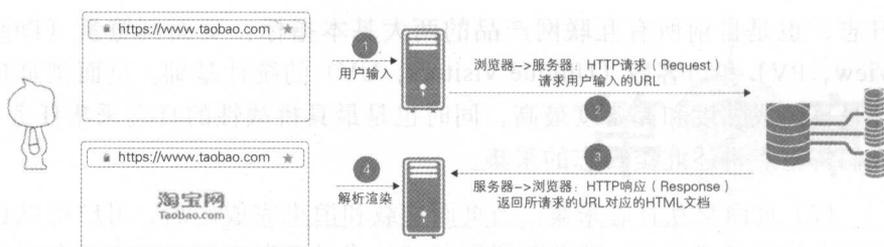


图 2.1 一次典型的互联网页面请求-响应过程

(1) 用户在浏览器内点击淘宝首页链接（或在地址栏中输入 `www.taobao.com` 并回车）。

(2) 浏览器向淘宝服务器发起 HTTP 请求。在本例子中，用户可以看见的内容只是显示于浏览器地址栏内的 `http://www.taobao.com`，而浏览器在执行时，会解析用户请求并按照 HTTP 协议中约定的格式将其转化为一个 HTTP 请求发送出去。

按照 HTTP 协议，一个标准的 HTTP 请求由如下三部分构成。

- 请求行 (HTTP Request Line)。请求行内有三个要素，分别是请求方法、所请求资源的 URL 以及 HTTP 协议版本号。在本例子中，这三个要素分别是 GET、`http://www.taobao.com/` 以及 HTTP 1.1，对于我们所讨论的话题，记住请求行内最重要的信息是这个 URL 就可以了。
- 请求报头 (HTTP Message Header)。请求报头是浏览器在请求时向服务器提交的附加信息，请求报头一般会附加很多内容项（每项内容被称为一个头域 (Header Field)，在不引起混淆的情况下，往往将 Header Field 简称为 Header)。需要注意的是，如果用户在本次页面访问之前已经到访过网站或者已经登录，则一般都会请求头中附加一个或多个被称为 Cookie 的数据项，其中记录了用户上一次访问时的状态或者身份信息，我们只需理解浏览器在发起请求时会带上一个标明用户身份的 Cookie 即可。
- 请求正文 (HTTP Message Body)。这一部分是可选的，一般而言，HTTP 请求的正文都是空的，可以忽略。

(3) 服务器接收并解析请求。服务器端的业务处理模块按业务逻辑处理本次请求并按照 HTTP 协议规定的格式，将处理结果以 HTTP 响应形式发回浏览器。

与 HTTP 请求相对应，一个标准的 HTTP 响应也由三部分构成。

- 状态行。状态行标识了服务器对于此次 HTTP 请求的处理结果。状态行内的主要内容是一个由三位数字构成的状态码，我们最熟知的两个状态码分别是代表成功响应的 200 (OK) 和代表所请求的资源在服务器端没有被找到的 404 (Not Found)。
- 响应报头。服务器在执行响应时，同样可以附加一些数据项，这些数据项将在浏览器端被读取和使用。事实上，在大多数页面和应用中，响应报头内的内容在确保页面正确显示和业务正常进行方面都发挥着至关重要的作用。其中最重要的一类 Header 即上面所提到的 Cookie，浏览器所记录的 Cookie，其实是由服务器在响应报头内指令浏览器记录的。举个例子，如果用户在页面登录，则服务器会在登录请求的响应报头内指示浏览器新增一个名为 userid 的 Cookie 项，其中记录了登录用户的 id。如此一来，当用户随后再次访问该网站时，浏览器将自动在请求报头内附加这个 Cookie，服务器由此即可得知本次请求对应的用户到底是谁；如果服务器发现浏览器在请求时传递过来的 Cookie 有缺失、错误或者需要更新，则会在响应报头内指令浏览器增加或更新对应的 Cookie。
- 响应正文。和请求正文一样，这一部分在协议中也被定义为可选部分，但对于大多数 HTTP 响应而言，这一部分都是非空的，浏览器请求的文档、图片、脚本等，其实就是被包装在正文内返回浏览器的。在本例子中，服务器会将淘宝首页对应的 HTML 文档封装在正文内。

(4) 浏览器接收到服务器的响应内容，并将其按照文档规范展现给用户，从而完成一次请求。在本例子中，浏览器请求淘宝首页，服务器返回对应的 HTML 文档，浏览器即按照 HTML 文档规范解析文档并将整个页面渲染在屏幕上。

上面描述了一次典型的网页浏览过程，如果需要记录这次浏览行为，则采集日志的动作必然是附加在上述四个步骤中的某一环节内完成的。在第一步和第二步，用户的请求尚未抵达服务器；而直到第三步完成，我们也只能认为服务器处理了请求，不能保证浏览器能够正确地解析和渲染页面，尚不能确保用户已确实打开页面，因此在前三步是无法采集用户的浏览日志的。那么采集日志的动作，需要在第四步，也就是浏览器开始解析文档时才能进行。根据前文所述，可以很自然地得出在这一模式下最直接的日志采集思路：在 HTML 文档内的适当位置增加一个日志采集节点，当浏览器解析到这个节点时，将自动触发一个特定的 HTTP 请求到日志采集服务器。如此一来，当日志采集服务器接收到这个请求时，就可以确定浏览器已经成功地接收和打开了页面。这就是目前几乎所有互联网网站页面浏览日志采集的基本原理，而业界的各类网页日志采集的解决方案只是在实施的细节、自动采集内容的广度以及部署的便利性上有所不同。

目前阿里巴巴采用的页面浏览日志采集方案的流程框架如图 2.2 所示。在图 2.2 所示的页面浏览日志采集过程中，所涉及的日志相关的几个主要过程简单介绍如下：

(1) 客户端日志采集。日志采集工作一般由一小段被植入页面 HTML 文档内的 JavaScript 脚本来执行。采集脚本被浏览器加载解析后执行，在执行时采集当前页面参数、浏览行为的上下文信息（如读取用户访问当前页面时的上一步页面）以及一些运行环境信息（如当前的浏览器和分辨率等）。在 HTML 文档内植入日志采集脚本的动作可以由业务服务器在响应业务请求时动态执行，也可以在开发页面时由开发人员手动植入。在阿里巴巴，这两种方式均有采用，其中前一种方式的占比较高，这一点与业界的普遍状况有所不同。图 2.2 中的第三、四步描述了阿里业务服务器端植入日志采集脚本的过程。

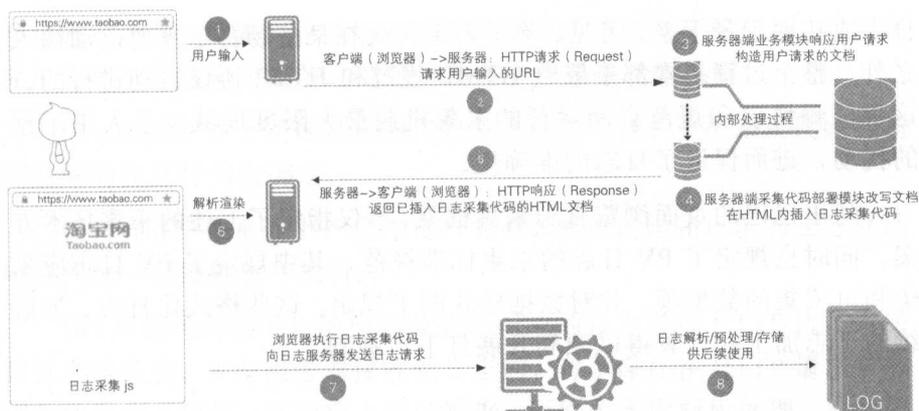


图 2.2 阿里巴巴页面浏览日志采集方案流程框架

(2) 客户端日志发送。采集脚本执行时，会向日志服务器发起一个日志请求，以将采集到的数据发送到日志服务器。在大多数情况下，采集完成之后会立即执行发送；但在个别场景下，日志采集之后可能会经过一段时间的延迟才被发出。日志采集和发送模块一般会集成在同一个 JavaScript 脚本文件内，且通过互联网浏览器必然支持的 HTTP 协议与日志服务器通信，采集到的日志信息一般以 URL 参数形式放在 HTTP 日志请求的请求行内。

(3) 服务器端日志收集。日志服务器接收到客户端发来的日志请求后，一般会立即向浏览器发回一个请求成功的响应，以免对页面的正常加载造成影响；同时，日志服务器的日志收集模块会将日志请求内容写入一个日志缓冲区内，完成此条浏览日志的收集。

(4) 服务器端日志解析存档。服务器接收到的浏览日志进入缓冲区后，会被一段专门的日志处理程序顺序读出并按照约定的日志处理逻辑解析。由日志采集脚本记录在日志请求行内的参数，将在这个环节被解析（有时候伴随着转义和解码）出来，转存入标准的日志文件中并注入实时消息通道内供其他后端程序读取和进一步加工处理。

经过采集—发送—收集—解析存档四个步骤，我们将一次页面浏览

日志成功地记录下来。可见，除了采集代码在某些场合下需要手动植入之外，整个过程基本都是依照 HTML 规范和 HTTP 协议自动进行的，这种依赖协议和规范自动运行的采集机制最大限度地减少了人工干预的扰动，进而保证了日志的准确性。

阿里巴巴的页面浏览日志采集框架，不仅指定了上述的采集技术方案，同时也规定了 PV 日志的采集标准规范，其中规定了 PV 日志应采集和可采集的数据项，并对数据格式做了规定。这些格式化日志，为后续的日志加工和计算得以顺利开展打下了基础。

2.1.2 页面交互日志采集

PV 日志的采集解决了页面流量和流量来源统计的问题，但随着互联网业务的发展，仅了解用户到访过的页面和访问路径，已经远远不能满足用户细分研究的需求。在很多场合下，需要了解用户在访问某个页面时具体的互动行为特征，比如鼠标或输入焦点的移动变化（代表用户关注内容的变化）、对某些页面交互的反应（可借此判断用户是否对某些页面元素发生认知困难）等。因为这些行为往往并不触发浏览器加载新页面，所以无法通过常规的 PV 日志采集方法来收集。在阿里巴巴，通过一套名为“黄金令箭”的采集方案来解决交互日志的采集问题。

因为终端类型、页面内容、交互方式和用户实际行为的千变万化不可预估，交互日志的采集和 PV 日志的采集不同，无法规定统一的采集内容（例如，活动页面的游戏交互和淘宝购物车页面的功能交互两者相比，所需记录的行为类型、行为数据以及数据的结构化程度都截然不同），呈现出高度自定义的业务特征。与之相适应，在阿里巴巴的日志采集实践中，交互日志的采集（即“黄金令箭”）是以技术服务的形式呈现的。

具体而言，“黄金令箭”是一个开放的基于 HTTP 协议的日志服务，需要采集交互日志的业务（下文简称“业务方”），经过如下步骤即可将自助采集的交互日志发送到日志服务器。

(1) 业务方在“黄金令箭”的元数据管理界面依次注册需要采集交

互日志的业务、具体的业务场景以及场景下的具体交互采集点，在注册完成之后，系统将生成与之对应的交互日志采集代码模板。

(2) 业务方将交互日志采集代码植入目标页面，并将采集代码与需要监测的交互行为做绑定。

(3) 当用户在页面上产生指定行为时，采集代码和正常的业务互动响应代码一起被触发和执行。

(4) 采集代码在采集动作完成后将对应的日志通过 HTTP 协议发送到日志服务器，日志服务器接收到日志后，对于保存在 HTTP 请求参数部分的自定义数据，即用户上传的数据，原则上不做解析处理，只做简单的转储。

经过上述步骤采集到日志服务器的业务随后可被业务方按需自行解析处理，并可与正常的 PV 日志做关联运算。

2.1.3 页面日志的服务器端清洗和预处理

上面介绍了阿里巴巴的两类浏览器页面日志的采集方案，并粗略介绍了日志到达日志服务器之后的解析处理。但在大部分场合下，经过上述解析处理之后的日志并不直接提供给下游使用。基于如下几个原因，在对时效要求较宽松的应用场合下，一般还需要进行相应的离线预处理。

(1) 识别流量攻击、网络爬虫和流量作弊（虚假流量）。页面日志是互联网分析和大数据应用的基础源数据，在实际应用中，往往存在占一定比例的虚假或者恶意流量日志，导致日志相关指标的统计发生偏差或明显谬误。为此，需要对所采集的日志进行合法性校验，依托算法识别非正常的流量并归纳出对应的过滤规则集加以滤除。这是一个长期而艰苦的对抗过程。

(2) 数据缺项补正。为了便利后续的日志应用和保证基本的数据统计口径一致，在大多数情况下，需要对日志中的一些公用且重要的数据项做取值归一、标准化处理或反向补正。反向补正，即根据新日志对稍早收集的日志中的个别数据项做回补或修订（例如，在用户登录后，对