

一种移动端 APM 系统框架设计与实现

金安

(上海爱奇艺新媒体科技有限公司,上海 200050)

摘要:随着移动应用数据爆发式增长,微服务架构越来越庞大,排查系统故障变得越来越复杂,建立一种能快速定位解决问题的系统势在必行。针对互联网行业特点及存在的相关问题,设计并实现了一种符合企业业务特点的移动端 APM 系统。对原始数据分析算法优化、数据存储聚合策略和符号解析系统等关键技术进行了阐述。APM 系统的运行数据表明,业务应用崩溃率下降了 50% 以上。

关键词:应用性能管理;拨测;卡顿;可变网格;密度偏差抽样

DOI: 10.11907/rjdk.211130

开放科学(资源服务)标识码(OSID):



中图分类号:TP319

文献标识码:A

文章编号:1672-7800(2022)001-0205-05

Design and Implementation of A Kind of Mobile APM System Framework

JIN An

(Shanghai iQIYI New Media Technology Co., Ltd., Shanghai 200050, China)

Abstract: With the continuous growth of mobile application and the popularity of micro-service architecture, it becomes complicated to troubleshoot system faults. Therefore, it is imperative to establish a system that can quickly locate and solve problems. According to the characteristics of the Internet industry and current situation of the enterprise business, design and implement a mobile APM system. In this paper, key technical points such as the optimization of original data analysis algorithm, data storage aggregation strategy and symbol parsing system are explained. The data of half a year after the system was put into operation shows that the failure rate of business application has been reduced by more than 50%.

Key Words: APM; dail testing; choppy; variable grid; density biased sampling

0 引言

随着云计算、大数据、虚拟化、容器化等新技术出现和移动互联网的普及,移动端应用系统架构越来越庞大,业务形态越来越多,给传统的监控分析带来全新挑战^[1],如何在应用出现问题时快速定位系统故障迫在眉睫。一些互联网企业的移动端应用系统呈现形式为超级 APP,通常服务于海量用户,不同地域、时段、请求的用户体验完全不同,单纯依靠人工发现和排查问题显然行不通。

如何寻找移动业务的性能瓶颈,定位应用系统故障,优化和提升用户体验是急需解决的问题,应用性能管理(Application Performance Management,简称 APM)系统应运而生^[1]。文献[2]基于 J2EE 架构从 Application 探针、管理服务和管理控制台 3 个部件设计并实现了 APM 系统;对于提高 APM 海量数据存储与访问需求的可靠性与读写效

率,文献[3]、文献[4]详细介绍了分布式协调服务 ZooKeeper、查询引擎 Elasticsearch、Storm 与 Kafka 等数据处理技术框架;文献[5]分析了中小型企业异构业务系统复杂现状,从设计原则、功能架构、技术架构等角色详细阐述面向中小型企业业务系统的 APM 框架。然而对于大型互联网系统,必须考虑海量数据处理算法与大数据存储聚合存储策略的优化;文献[6]提出一种基于 Canopy&K-means 聚类算法的分析算法,用于挖掘应用的典型性能瓶颈参数;文献[7]对于海量数据分析处理提出了基于可变网格划分的密度偏差抽样算法。以上算法都为本文数据处理算法优化提供了较好的参考。

本文以某互联网公司为例,针对该企业遇到的应用性能管理难题,解决快速定位故障,驱动产品迭代,缩短产品开发周期,设计并实现一套基于大数据平台的移动端 APM 系统。

收稿日期:2021-01-25

作者简介:金安(1980-),男,硕士,上海爱奇艺新媒体科技有限公司高级工程师,研究方向为互联网流式大数据采集存储、实时计算分析、大型互联网系统运维和云游戏研究。

1 应用性能管理

应用性能管理(APM)指对企业数字化的关键业务应用进行监测、优化,提高应用的可靠性和质量,保证用户得到良好服务,降低IT总拥有成本^[8]。

在互联网应用的交付链中,越靠近机房或商业云的区域,比如SLB、Web服务、应用服务器、DB、消息队列、分布式存储等,由于有专业人员的QoS保证,相对来说比较可控;靠近终端的用户区域,涉及到国内各大运营商的网络链路和第三方服务(CDN、消息推送、互联支付、跨国访问等),且用户移动设备多种多样,特别是国内安卓端设备各大厂家定制化严重,很容易发生因为设备的不兼容出现性能和用户体验问题。

移动互联网应用场景中经常会碰到如网络基站拥塞、各地运营商域名劫持、CDN节点故障、骨干网线路维护、云端服务的宕机和浏览器或移动设备不兼容等问题,这些问题都是APM系统应该关注和解决的。

APM技术分类可根据数据采集位置区分,基本上分为针对最终用户侧和应用服务侧两大类技术。通常说的端到端APM,实质上是这两端APM技术的融合。企业在选择APM产品时,除了满足功能及需求的匹配度,还需考虑APM所采用的数据采集方式。表1为APM数据采集方式优缺点对比。

Table 1 Comparison of advantages and disadvantages of APM's data collection methods

表1 APM数据采集方式优缺点对比

采集方式	优点	缺点
SDK嵌入	诊断定位问题深入	有一定开发量、维护成本较高
旁路监听	只需镜像交换机,无开发成本	较高的硬件资源,监控颗粒粗
日志采集	监控指标灵活,深入业务	日志格式需规范化,复用度差
拨测	24小时监控,灵活配置探测	模拟数据,精度程度不高
SNMP协议	通用性高,部署简单	监测指标固定,不可靠UDP协议

越来越复杂的应用架构和业务场景需要结合使用不同的APM技术,只有更加关注于运维数据采集,对端到端的性能数据进行相关性分析,才能快速准确地定位业务故障,支持精细化运营。

2 系统框架设计与实现

2.1 某企业移动端应用系统问题

某互联网企业随着APP系统的不断壮大,原来使用的类APM系统越来越不能满足业务需求。由于投递数据的缺失,业务多次在性能下降和服务异常时未能快速预测和定位问题点,导致流失了一定的用户量。

2.2 移动端应用数据监测与采集

APM中两个最重要的维度是最终的用户体验和业务指标,需要采集海量数据并根据算法优化进行聚合分类,

快速发现、定位和修复问题^[9]。

本文APM系统采集以下数据:

(1)崩溃数据。该数据晦涩难懂但十分重要,需要特殊翻译,包含堆栈、日志、用户行为路径、异常、寄存器、设备信息等数据。

(2)卡顿监控数据。包含ANR、UI卡顿栈、日志等数据。

(3)页面性能数据。包含APP启动时间、页面打开时间、埋点Trace时间分析等数据。

(4)网络监控数据。该数据最复杂也是问题最多的,包含Http请求性能、DNS、WebView请求、连接超时、CDN等数据。

(5)热更数据。包含热更耗时、成功率、到达率等数据。

(6)业务异常数据。包含异常栈、业务模块、接口耗时、附加数据等。

2.3 系统框架设计

APM系统设计根据覆盖范围分为数字化体验监控、终端程序发现跟踪和深度应用分析诊断3个层次分阶段实现。数字化体验监控主要体现在数据采集、监控和可视化;终端程序发现跟踪偏向程序的跟踪检查和问题定位等;深度应用分析诊断模式依托大数据分析和机器学习、AI技术,可以更加智能地预测、发现问题,在减少人为干预的情况下自动修复故障。

根据业务需求设计一个移动端APM实时监控系统,在收集到APP端海量数据后进行归类聚合,实时分析并基于相关阈值异常发出警告,给出离线分析报告。系统框架设计如图1所示。

(1)各种终端设备嵌入APM SDK,通过接口将业务所需要的信息投递到Nginx和Tracker系统,其中Nginx接收短数据信息,Tracker系统保存长数据信息。

(2)通过Flume将数据汇聚整理,投递数据根据业务需求进入实时和非实时Kafka消息队列中。

(3)实时消息队列信息可通过FileBeat组件将数据清洗后入库到ES集群中,提供给Kibana或者Grafana展示。

(4)清洗设备Spark Streaming把解析后的信息包存入HBase,并使用时间戳+序号作为HBase的key,Spark定时周期性触发Hadoop MR任务。任务被触发后,通过时间段信息获知key范围,然后对HBase中该范围的数据做scan还原,对数据做统计计算,比如某时间段的崩溃率、崩溃机型分布、卡顿类型等。

(5)Hadoop MR把统计结算结果写入MySQL等数据库,计算结果包括各APP版本的启动次数、崩溃次数、崩溃机型分布、崩溃OS版本分布等,该结果提供给Portal页面展示。

(6)符号解析系统启动对Hbase中的“Crash信息”做二次处理,将处理结果写入HBase、ES和MySQL中。具体步骤为:iOS崩溃的符号还原,类型聚合;安卓/iOS崩溃的模块和类定位,通过Git/配置文件定位崩溃并将问题自动投递

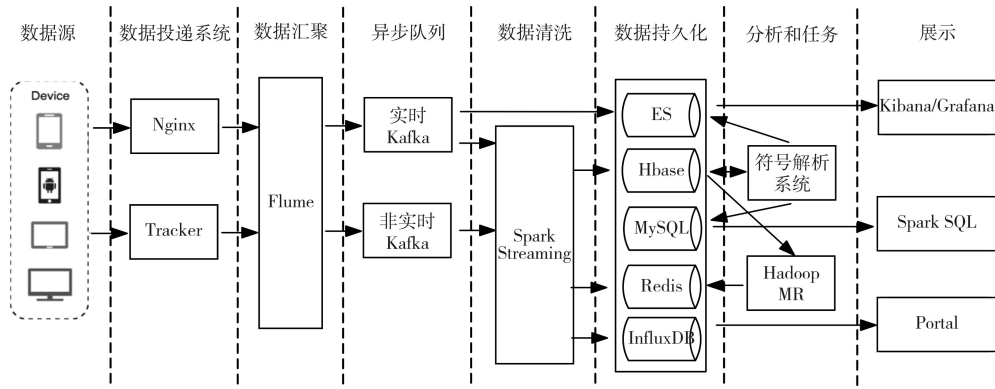


Fig. 1 APM system architecture
图 1 APM 系统框架

到 Jira 系统邮件,或者短信告知相关研发 Owner。

(7)用户可以通过浏览器查看数据和各类图表,展示端从各种数据源获取数据返回给查询端。

3 APM 系统实现关键技术

3.1 APM 系统数据存储聚合策略

案例企业用户多,采集的数据庞大,为达到更好的时效和经济性,从数据存储和数据计算两方面进行优化。

Hbase 主要用来保存完整用户崩溃日志、崩溃前的用户操作细节、ANR/卡顿等详情。由于 APP 版本较多,每个崩溃和 ANR/卡顿类型最多每天保存 1 000 条,详细数据按照国家要求保存 6 个月。当 Hadoop MR 任务根据原始数据计算统计写入 MySQL 和 Redis 等数据库后,为节约存储空间,根据业务特点清除 HBase 中不再需要的对应的明细数据。

ES 集群采集实时的 Kafka 消息,通过 Kibana 实时展现各种趋势图,包含 APP 名称、APP 版本号、APP Patch 版本号、时间范围、异常类型等。MySQL 数据库保存了 HBase 中详细信息 key 的索引关系,另外符号解析系统和 Hadoop MR 任务也会更新 MySQL 中的崩溃统计数据信息。

3.2 APM 系统大数据聚类算法优化说明

考虑到 PB 级数据采集和未来系统的数据扩展,为加快数据分析,有必要对 APM 系统中大数据聚类算法进行优化。

相对于固定网格划分的密度偏差抽样(Density Biased Sampling, DBS)算法,可变网格划分指首先分析原始数据集的属性特征及分布特征,然后划分与数据集特征保持一致的网格空间。对原始数据集的每一维动态地划分区间,使每一维上的区间大小不等,不同维的区间段个数也不一样,最终得到大小不等的区间构成网格空间^[10-11]。该算法在保持原始数据分布特征的基础上能有效减少网格数,比固定网格划分技术更为灵活。

3.2.1 可变网格划分方法

构建可变网格时对原始的每一维数据集(N 个数据点、d 维数据集 A)用快速排序法按小到大排序后再进行等深划

分,然后对密度相似的相邻区间段进行合并操作,数据集每一维重复此操作最终实现可变网格的划分^[12],公式如下:

排序后 i 维数据集:

$$D_{ai} = \{D_{i1}, D_{i2}, \dots, D_{in}\}, i = \{1, 2, \dots, d\} \quad (1)$$

计算各相邻区段间的密度相似性:

$$\varepsilon_i = \begin{cases} I_{ik} / I_{i(k+1)}, & I_{ik} \leq I_{i(k+1)} \\ I_{i(k+1)} / I_{ik}, & I_{ik} > I_{i(k+1)} \end{cases} \quad (2)$$

计算每维数据阈值:

$$V_i = \alpha \left(\frac{1}{k-1} \sum_{i=1}^{k-1} \varepsilon_i \right) \quad (3)$$

对 D_{ai} 等深划分为 k 个区间段,每个区间段数据个数为 N/k ,区间密度为该区间上下确界的差, I_{ik} 代表第 i 维上第 k 区段的密度, α 可取经验值 $1/2$ 。

若密度相似性 ε_i 大于阈值 V_i ,表示这两个相邻区间段密度相似,全部比较完成后合并密度相似的相邻区间段,然后重复式(2)和式(3),完成可变网格划分^[13]。

3.2.2 改进型可变网格划分 DBS 核心算法

传统可变网格划分需要对每维数据先进行排序,这必然增加网格划分的时间。为节约时间,对该网格的划分方法进行标准方差维度改进如下:

$$\bar{D} = \sum_{i=1}^N \frac{\bar{D}_{ij}}{N} \quad (4)$$

\bar{D} 为某一维度所有数据的平均值。

公式(5)中, σ_i 为某一维度所有数据的标准方差:

$$\sigma_i = \sqrt{\frac{\sum_{i=1}^N (\bar{D}_{ij} - \bar{D})^2}{N-1}} \quad (5)$$

划分网格时不必通过对数据集每一维进行排序,只需对每一维度标准方差信息动态确认该维划分粒度即可,理论上缩短了网格矩阵构建时间。将矩阵看成空间,计算各维相邻区间密度相似性与整维密度阈值,得出划分后的网格单元数和各网格的数据数。

根据 DBS 算法定义和条件,推定在每个聚合里每个数据点被抽取的概率函数如下:

$$f(n_i) = p/n_i^e \quad (0 \leq e \leq 1) \quad (6)$$

式中, n_i 为第 i 个聚合大小, 实际上 e 可取经验值为 $1/2$ 。

样本数量 n 等于 D_i 抽取样本数量的总和, 得出以下公式:

$$\begin{cases} n = \sum_{i=1}^D n_i f(n_i) = \sum_{i=1}^D n_i p / n_i^e \\ p = n / \sum_{i=1}^D n_i^{1-e} \end{cases} \quad (7)$$

结合式(6)和式(7)得出各个聚合数据点的抽样概率函数为:

$$f(n_i) = n / (n_i^e \sum_{i=1}^D n_i^{1-e}) \quad (8)$$

由式(8)得出每个聚合抽样的个体函数为:

$$w_i = f(n_i) \times n_i \quad (9)$$

3.3 APM 符号解析系统

用户体验监控是 APM 系统必须要解决的首要问题, 其中 APP 崩溃数据解析又是重中之重。本文中的符号解析系统基本上由以下需求或功能构成: 收集包括 Objective-C、Swift、C/C++ 在内的 crash 和 dump crash 的调用堆栈信息, 分析发生在 crash 之前的用户操作细节以更快定位故障原因。本文设计的 APM 符号解析系统如图 2 所示。

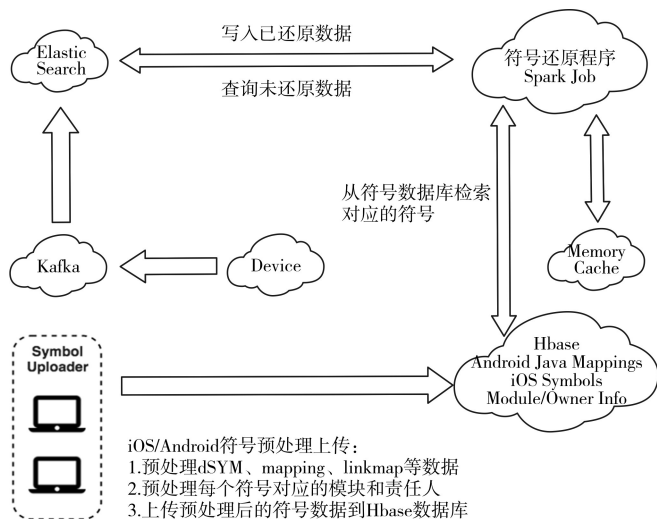


Fig. 2 APM symbolic parsing system

图 2 APM 符号解析系统

APM 符号解析步骤如下: ①通过 APM 后台上传需要的 dSYM、mapping、linkmap 和 SDK 类名等数据文件, 经过预处理后存入映射表数据库加速符号化; ②通过 APM 系统定时调用符号解析系统对崩溃文件进行解析并进行地址符号转换; ③拿到转化后可读的崩溃信息后, 按照定义的格式写入 ES 和其他数据持久化集群提供 Portal 查询; ④根据后台上传的 SDK 类名列表, 解析出崩溃来自哪一类 App SDK; ⑤把解析的结果输出到相关数据库中, 等待被生产系统获取。

通过嵌套的聚合查询, 可预先聚合统计指定多个维度的数据, 使离线数据聚合优化。由于 ES 集群数据量巨大,

要对每个查询请求启动 cache 缓存优化策略, 将 ES 的索引刷新率降到 5min 以提高 cache 命中率, 减少 cache 频繁刷新。

4 系统上线数据结果分析

本文的 APM 系统于 2020 年上线, 与市面上常见的 skywalking、zipkin 等 APM 组件采用 Jmeter 工具进行并发百万用户的压力测试比较, 发现自研的 APM 系统投递对系统吞吐量影响最小, 仅为无投递时的 10% 左右, 而 skywalking、zipkin 达到 25% 以上。内部服务 CPU 和内存的影响差不多在 10% 以内, 前两者在 15% 左右^[14]。

为了加快符号化过程, 将上传预处理过的符号数据传到 hbase 数据库, 通过地址直接 scan 还原, 一次还原时间小于 5ms; 通过 cache 加速技术, 使命中率达到 96% 以上。大部分情况下还原一条信息耗时小于 1ms, 说明在进行抽样过程中, 基于标准方差的网格划分方法和 DBS 算法并结合缓存技术, 在适当降低样本准确率的情况下能显著提高大数据的抽样速度, 减少构建网格空间的时间, 从而提高算法性能, 加快数据分析过程。极限情况下, 1ms 内能处理并匹配 29 652 条崩溃消息, 并以实时邮件或其他触达方式反馈到责任人。

该系统上线后, 极大缩短了人员定位与排障时间, 平均定位 bug 时间从 30min 降低到 5min 左右, 应用的崩溃率从 0.345% 下降到 0.134% 左右, 接近行业顶尖公司千分之一的崩溃率。

5 结语

本研究在生产环境中基本实现了 APM 系统设计的数字化体验监控和终端程序发现跟踪功能。系统能进行指定崩溃问题分析、实时崩溃问题趋势分析、iOS/Android 性能检测、问题协同报告和崩溃率自动告警等, 较好地解决了先前无法解决的痛点, 提高了定位线上故障的效率。

后续应重点在优化算法、引入机器人学习和 AI 技术方面深入研究。

参考文献:

[1] LIANG W, YANG M C, FENG M, et al. Research and development of application performance management [J]. Telecommunications Technology, 2017(6):42-45.
梁伟, 杨明川, 冯明, 等. 应用性能管理技术的研发与应用[J]. 电信技术, 2017(6):42-45.

[2] WANG Q. Design and implementation of application performance management system [J]. Computer & Network, 2018, 44(24):63-65.
王强. 应用性能管理系统的设计与实现[J]. 计算机与网络, 2018, 44(24):63-65.

[3] GAN Z M. Design and implementation of application performance management anomaly monitoring system [D]. Hangzhou: Zhejiang University, 2018.

- 甘志敏. 应用性能异常监控系统设计与实现[D]. 杭州: 浙江大学, 2018.
- [4] HUANG S. Design and implementation of data processing subsystem in application performance management system [D]. Beijing: Beijing University of Posts and Telecommunications, 2017.
黄岁. 应用性能管理系统中数据处理子系统的设计与实现[D]. 北京: 北京邮电大学, 2017.
- [5] YAN X X. A study on APM of SME's business system [D]. Ji'nan: Shandong University, 2018.
延啸雪. 中小型企业业务系统的应用性能管理研究[D]. 济南: 山东大学, 2018.
- [6] LIU Y. Research and implementation of application performance bottleneck analysis system in APM [D]. Beijing: Beijing University of Posts and Telecommunications, 2017.
刘杨. 应用性能管理系统中的应用性能瓶颈分析系统的研究与实现[D]. 北京: 北京邮电大学, 2017.
- [7] SHENG K Y, QIAN X Z, WU Q. Density biased sampling algorithm based on variable grid division [J]. Journal of Computer Application, 2013, 33(9): 2419-2422.
盛开元, 钱学忠, 吴秦. 基于可变网格划分的密度偏差抽样算法[J]. 计算机应用, 2013, 33(9): 2419-2422.
- [8] TIAN B, WANG W, SU Q, et al. Research on application performance management platform based on micro services architecture [J]. Information Technology and Informatization, 2018(1): 125-128.
田兵, 王伟, 苏琦, 等. 基于微服务架构的应用性能监控平台研究[J]. 信息技术与信息化, 2018(1): 125-128.
- [9] ZHAO F, CHEN F, LI Z, et al. Research on full link monitoring method of application system [J]. Journal of Frontiers of Computer Science and Technology, 2017, 11(1): 626-628.
赵峰, 陈峰, 李志, 等. 应用系统全链路监测方法研究[J]. 计算机科学与探索, 2017, 11(1): 626-628.
- [10] CHEN X Q, JIN X L, WANG Y Z, et al. Survey on big data system and analytic technology [J]. Journal of Software, 2014, 25(9): 1889-1908.
程学旗, 靳小龙, 王元卓, 等. 大数据系统和分析技术综述[J]. 软件学报, 2014, 25(9): 1889-1908.
- [11] LYU D, LONG H, GAO J, et al. An improved algorithm for density deviation sampling based on uneven data [J]. Software Guide, 2018, 17(2): 78-80.
吕丹, 龙华, 高杰, 等. 基于不均匀数据的密度偏差抽样改进算法[J]. 软件导刊, 2018, 17(2): 78-80.
- [12] PAN C Y, WU Y F, LI F. An application of density biased sampling technique of variable meshing for clustering [J]. Journal of Kaili University, 2017, 35(3): 15-19.
潘春燕, 吴有福, 李方. 一种基于可变网格划分的密度偏差抽样技术及其在聚类中的应用研究[J]. 凯里学院学报, 2017, 35(3): 15-19.
- [13] Architecture House. APM principle and frame selection [EB/OL]. <https://www.cnblogs.com/xiaoqi/p/apm.html>, 2018.
架构之家. APM 原理与框架选型 [EB/OL]. <https://www.cnblogs.com/xiaoqi/p/apm.html>, 2018.
- [14] HUAWAI CLOUD. Application performance management [EB/OL]. https://en.wikipedia.org/wiki/Application_performance_management, 2021.
- [15] CHEN B. Research and implementation of application performance management based on Android [D]. Guangzhou: South China University of Technology, 2018.
成斌. 基于 Android 应用的性能管理方案研究与实现[D]. 广州: 华南理工大学, 2018.
- [16] SHI F T, MENG X D, HUO C. Improving the precision of smart campus services based on application performance management concept [J]. Journal of Shenzhen University (Science and Engineering), 2018, 37(1): 160-164.
史福田, 孟曦东, 霍超. 基于 APM 理念提升智慧校园服务精细度[J]. 深圳大学学报(理工版), 2018, 37(1): 160-164.
- [17] LI H, YANG J Y, MO Q. Research on application scheme of APM in IT operation platform of China unicom [J]. Designing Techniques of Posts and Telecommunications, 2019, 33(11): 84-88.
李衡, 杨洁艳, 莫穹. APM 在中国联通 IT 运营平台应用方案研究[J]. 邮电设计技术, 2019, 33(11): 84-88.
- [18] LIAO C, WU F Q, SHU Y, et al. Application of application performance management on core business system [J]. Information Technology & Standardization, 2021, 36(1-2): 69-73.
廖臣, 吴方权, 舒彧, 等. 应用性能管理在电网核心业务系统中的应用[J]. 信息技术与标准化, 2021, 36(1-2): 69-73.
- [19] RAO Y. A centralized performance management platform based on enterprise big data applications [J]. Information & Communications, 2019, 35(2): 279-280.
饶毅. 一种基于企业大数据应用的集中性能管理平台[J]. 信息通讯, 2019, 35(2): 279-280.

(责任编辑: 杜能钢)