计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(7):57−62 [doi: 10.15888/j.cnki.csa.006445]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于流计算的电力调度网络流量监测平台①

吴　奔[1,2], 李喜旺[1], 周心圆[3]

[1](中国科学院 沈阳计算技术研究所, 沈阳 110168)
[2](中国科学院大学, 北京 100049)
[3](吉林大学, 长春 130012)
通讯作者: 吴　奔, E-mail: wuben15@mails.ucas.ac.cn

摘　要: 由于电力调度网出现任何网络故障都可能发生极度严重的事故, 因此具有的极高可靠性及安全性的要求. 而当前传统的网络监测系统在面对大数据量时, 其实时处理能力和扩展能力都无法满足需求. 因此对实时产生的大规模各类型数据的分析处理则需要一种专门的实时数据分析平台完成. 本文结合电力调度信息网络的特点以及监测准确性及实时性的需求, 构建出一个基于流计算的数据处理分析平台, 以 Apache Spark 中的 Spark Streaming 为代表的开源流计算框架, 加入如 Kafka 分布式消息队列、Redis 内存数据库等组件, 为数据分析平台提供稳定高效的数据来源和数据服务接口, 从而实现适用于电力调度网的各类海量数据的实时分析处理完成流量异常监测场景.

关键词: 智能电网; 电力大数据; 流计算; Apache Spark; 实时监测

引用格式: 吴奔,李喜旺,周心圆.基于流计算的电力调度网络流量监测平台.计算机系统应用,2018,27(7):57–62. http://www.c-s-a.org.cn/1003-3254/6445.html

Power Dispatch Network Flow Monitoring Platform Based on Flow Calculation

WU Ben[1,2], LI Xi-Wang[1], ZHOU Xin-Yuan[3]

[1](Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)
[2](University of Chinese Academy of Sciences, Beijing 100049, China)
[3](Jilin University, Changchun 130012, China)

Abstract: Due to any network failure of the power dispatching network may lead to serious accidents, it is required to have high reliability and safety. Faced the large amount of data, the traditional network monitoring system cannot meet the demand at present in terms of the actual processing capacity and expansion capacity. Therefore, a special real-time data analysis platform is needed for the real-time analysis and processing of large amount of data. This study constructed a platform based on flow calculation. Spark Streaming in Apache Spark of the open source stream computing framework, adding Kafka message queue and Redis memory database components, provides stable data sources and efficient interface for data analysis and data service platform, so as to realize the real-time analysis and processing of all kinds of massive data, thus to complete the flow anomaly monitoring of power dispatching network.

Key words: smart grid; power big data; flow calculation; Apache Spark; real-time monitoring

1　引言

电力调度网是电网调度自动化、信息化的基础, 是确保电网安全、稳定、经济运行的重要手段, 是电力系统的重要基础设施, 传统的电力调度网安全监测, 主要是依靠工程师对网络设备进行排查或依靠网管对管理信息库及参数的分析进行定位.

① 基金项目: 国家科技重大专项 (2017ZX01030-201)
Foundation item: Major Project of the Ministry of Science and Technology of China (2017ZX01030-201)
收稿时间: 2017-11-08; 修改时间: 2017-12-15; 采用时间: 2017-12-27; csa 在线出版时间: 2018-06-27

随着电力系统信息化进程的加快, 持续推动了实时监测系统、现场移动检修系统、测控一体化系统、智能变电站和电力信息管理系统的扩建和应用, 使电力行业正逐渐步入到由复杂及异构数据源广泛存在和驱动的电力大数据时代. 电力学术领域开始利用云计算技术解决智能电网海量数据, 但还是无法达到很好的实时处理能力. 想要真正实现海量实时监测, 需要研究其他大数据处理技术, 例如利用内存计算, 大数据流计算等技术, 如目前主流的大数据流计算框架 Hadoop、Storm、Spark 等[1−3], 采用流计算对产生的数据流进行实时处理, 并将数据在内存数据库中缓存, 通过内存计算的方式加速数据的处理速度[4], 提高分析处理的性能.

2 流计算处理技术及系统设计

2.1 流计算技术介绍及计算框架的选择

相较于传统的数据处理方式, 流计算的技术特点主要体现在流入系统的数据流是实时的, 流计算能够对流入的数据进行实时处理, 并将数据在内存数据库中缓存, 通过内存计算的方式加速数据的处理速度, 提高分析处理的性能. 流数据处理的一般过程如图 1 所示.

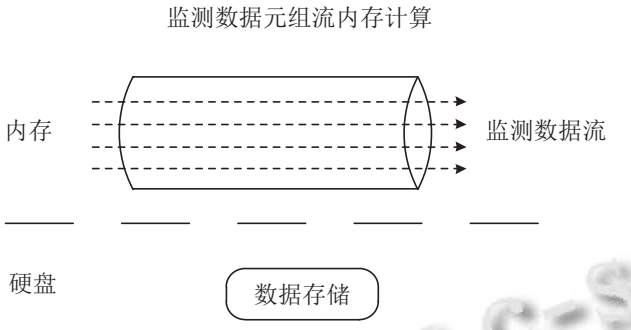

图 1 流计算处理一般过程

目前著名的开源数据流计算框架有 Hadoop 平台的 MapReduce 计算框架, Apache Storm 计算框架和 Apache Spark 计算框架, 他们是目前最常见的处理海量数据的开源框架.

Hadoop 是磁盘级计算, 而 Storm 和 Spark 是内存级计算, 磁盘访问延迟约为内存访问的 75 000 倍, 因此 Storm 和 Spark 更快. 对于 Storm 和 Spark 这两个高性能并行计算引擎的最大区别在于实时性: Spark 是准实时, 先收集一段时间再处理, 实时计算延迟是秒级; 而 Storm 是纯实时, 实时计算延迟是毫秒级. 但 Spark 拥有更高的吞吐量, Spark 还有一个特别的地方是, Spark 的软件栈允许将一些 library (Spark SQL, MLlib, GrapnX) 与数据流相结合[5], 提供便捷的一体化编程模型. Spark 的各个组件如图 2 所示.

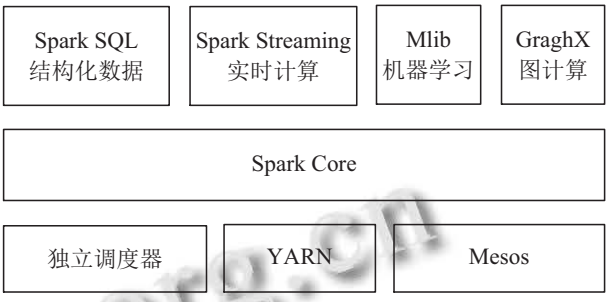

图 2 Spark 软件栈

Spark 计算框架解决了大数据处理遇到的批处理, 实时流处理和交互式查询等难题, 结合 Spark 高度抽象的 RDD (Resilient Distribute Dataset, 弹性分布式数据集) 概念[6], 针对多种不同的数据处理场合, 基于 Spark 的编程模式将被同一成相同的处理方式, Spark 统一了技术栈, 降低了研发成本. 另外 Spark 拥有更清晰, 等级更高的 API.

2.2 流计算网络监测模型介绍

为满足对电力调度数据网实时监测分析的实时性和高吞吐量的要求, 基于流计算的大数据实时处理分析基础平台以电力调度网络的大量实时监测数据为处理对象, 主要包括: 数据接入模块, 训练模块, 实时计算模块, 分布式存储及可视化模块. 分布式存储使用内存数据库和分布式文件数据库, 完成对实时推送数据进行存储, 实现实时分析结果存储, 以及离线处理功能. 流计算网络监测模型图, 如图 3 所示.

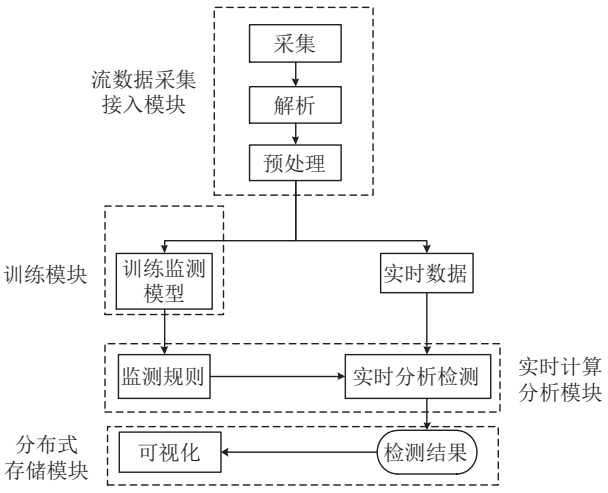

图 3 网络监测模型图

2.3 系统的整体架构及工作流程

基于流数据的实时处理分析基础平台以电力调度网的大量实时监测数据为处理对象, 主要包括数据源接入, 实时流计算, 以及分布式存储展示三个基本过程. 其中考虑到调度数据网中产生的实时监测数据的源头很多, 而且数据源只有接入实时处理系统后, 才可以进行流计算处理, 这里数据源是通过自适应采集获取的特定类型的数据. 结合数据流处理流向, 实时流计算系统框架图如图 4 所示.

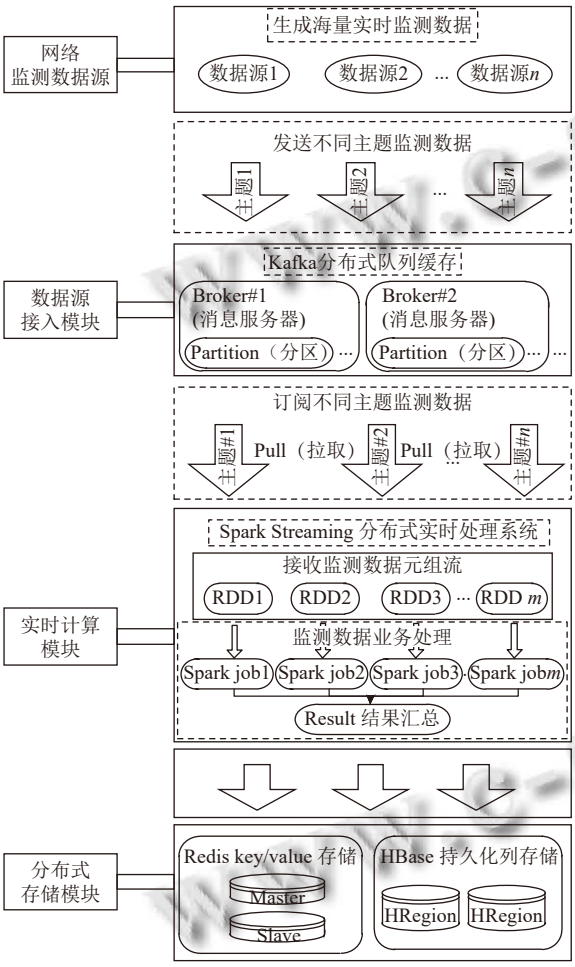

图 4 流计算实时处理系统整体架构图

2.3.1 数据接入模块

使用分布式消息队列系统 Kafka 作为系统的数据接入模块[7], 发挥其发布订阅消息传递机制及海量消息缓存特性, 为实时监测数据的连续流计算提供数据保障. 由于数据流的生成方式采用的是 Kafka 分布式消息队列, 因此数据在进行整合发送时, 还需要根据发送数据的类型, 将数据添加话题字段, 同一 Topic 内部的消息按照一定的 key 和算法被分区到不同的服务器上. 本系统可以包含多种数据源, 如调度网设备运行状态信息, 调度网网络流量特征等, 发布信息时流数据产生系统作为 Kafka 消息数据的生产者将数据流分发给 Kafka 消息主题, 流计算系统 Spark Streaming 实时消费并计算数据. Kafka 分布式集群架构如图 5 所示.

2.3.2 实时流计算模块

系统平台的实时流计算模块主要是基于 Spark Streaming 的分布式流计算框架构成, 它将流式计算分解成一系列短小的批处理作业[8], 将 Kafka 中每一个话题的连续数据源定义为一个数据流 DStream, 而 DStream 为每个时间段所对应的 RDD 的集合, 每一段数据都转化成 Spark 中的 RDD 弹性分布式数据集. Dstream 数据流的定义如图 6 所示.

然后将 Spark Streaming 中对 DStream 的 Transformation 操作变为针对 Spark 中对 RDD 的 Transformation 操作[9], 将 RDD 经过操作变成中间结果保存在内存中. 整个流式计算根据业务的需求可以对中间的结果进行叠加, 或者存储到外部设备. Spark Streaming 的运行流程如图 7 所示.

2.3.3 分布式存储模块

为提高数据分析处理和数据监测预警的实时性, 对于数据的存储模块则选用内存数据库实现, 这里使用分布式内存数据库 Redis 将实时处理分析的结果进行数据 key/value 存储. 由于内存数据库存储容量限制, 对于访问频率较低, 数据量较大, 用以进行定期离线分析的数据,则需要借助分布式文件数据库 HBase 对其进行存储, 确保数据存储的可靠性, 高并发, 及扩展能力.

3 电力调度网络实时监测应用实现

3.1 网络流量异常监测

网络流量异常监测是网络安全防护至关重要的方法, 由于网络攻击具有突发性, 要求我们能够及时发现可疑网络流量, 从而采取网络防护措施. 网络流量异常监测主要实现方法[10], 首先获取正常通讯下的网络数据和攻击下的异常网络数据, 将采集到的网络数据作为带标签的训练样本[11], 可以结合 Spark 软件栈中的 MLib 机器学习函数库应用于流数据分析中[12], 通过聚类算法对训练样本进行聚类, 建立网络流量分类模型. 结合流处理框架 Spark Streaming, 程序加载分类模型对新增的流量数据数据进行分类, 对大规模网络流量准实时监测[13,14].

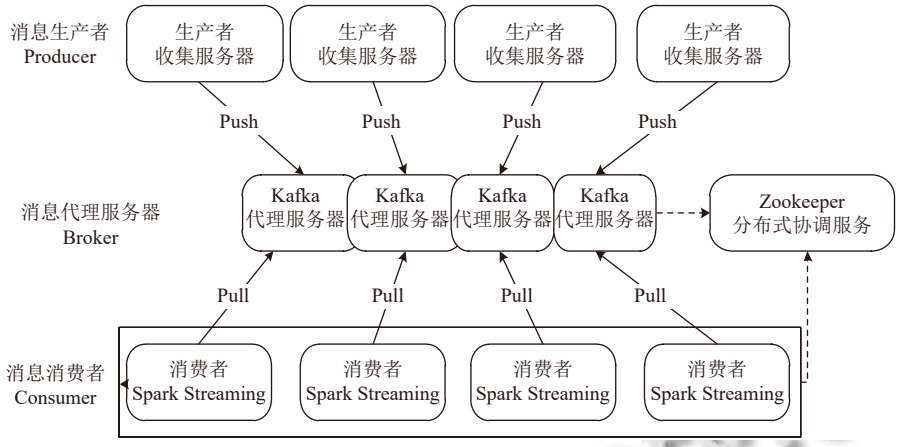

图 5 Kafka 分布式集群架构图

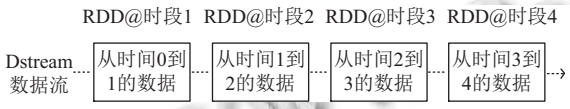

图 6 Dstream 的定义

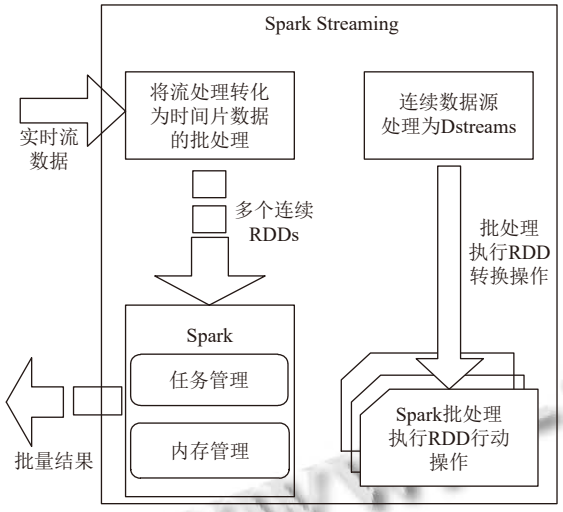

图 7 Spark Streaming 运行流程图

3.2 异常监测的实现

对于网络流量的特征向量, 采用基于机器学习的流量异常监测方法最常用的是聚类算法对数据集样本进行训练[15]. 对于如 K-means 等传统的划分聚类方法仅能发现球状簇, 它们很难发现任意形状的簇, 无法避免地将噪声或离群点包含进簇中. 为了发现任意形状的簇, 可以把簇看做数据空间中被稀疏区域分开的稠密区域, 即基于密度实现聚类. 对于对象 o 的密度则可以用靠近 o 的对象数度量. DBSCAN (Density Based Spatial Clustering of Application with Noise, 具有噪声应用的基于密度的空间聚类) 则是基于密度聚类算法的典型代表[7]. 该算法指定参数 ε 来表示每个对象的邻域半径, 对象 o 的 ε 邻域则是以 o 为中心、以 ε 为半径的空间. 邻域的大小由参数 ε 确定, 因此邻域的密度可以简单地用邻域内的对象数度量. DBSCAN 通过另一参数 MinPts, 即指定稠密区域的密度阈值, 来衡量邻域是否稠密. DBSCAN 算法在发现簇的过程如下文.

(1) 首先将给定数据集 D 中的所有对象都标记为“unvisited”.

(2) DBSCAN 随机地选择一个未访问的对象 p, 标记 p 为“visited”, 并检查 p 的 ε-邻域是否至少包含 MinPts 个对象. 如果不是, 则 p 被标记为噪声点.

(3) 否则为 p 创建一个新的簇 C, 并且把 p 的 ε-邻域中的所有对象都放到候选集合 N 中. DBSCAN 迭代地把 N 中不属于其他簇的对象添加到 C 中.

在此过程中, 对于 N 中标记为“unvisited”的对象, DBSCAN 把它标记为“visited”, 并且检查它的 ε-邻域. 如果的 ε-邻域至少有 MinPts 个对象, 则的 ε-邻域中的对象都被添加到 N 中. DBSCAN 继续添加对象到 C, 直到 C 不能再扩展, 即直到 N 为空. 此时, 簇 C 完全生成, 于是被输出. 为了寻找下一个簇, DBSCAN 从剩下的对象中随机地选择一个未访问的对象. 聚类过程继续, 直到所有对象都被访问.

4 实验与结果分析

4.1 实验环境

系统实验环境所使用的 Spark 集群搭建在基于

Hadoop 的基于分布式的安装, 集群有 3 个节点, 其中将一个节点配置为 Master, 其他 2 个配置为 Slave, 每个节点的配置都是内存 8 GB, 并搭载 Centos 操作系统, 相关软件版本如表 1 所示.

表 1 集群的软件配置

软件	版本
Hadoop	2.6.0
Spark	1.6.2
Kafka	0.9.0.1
Redis	3.2.4
Scala	2.11.6

4.2 实验结果与分析

本论文使用的数据为从电力调度数据网通过自适应采集及预处理过的网络流量数据, 每个网络连接的统计信息, 数据集的大小约为 708 M, 包含 490 万个连接. 数据集中每个连接信息包括发送的字节数, 登录次数, TCP 错误数等. 数据集包含 38 个特征, 下面是其中的一个连接的样例:

2, tcp, http, SF, 1684, 363, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 104, 66, 0.63, 0.03, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, normal.

数据集中每个连接的信息包括发送的字节数, 登陆次数, TCP 错误数等. 以上代表一个 TCP 连接, 他访问 http 服务, 发送了 1684 字节的数据, 收到数据 363 字节, 用户登录成功等. 许多特征值取值为 0 或 1, 比如第 15 列的 su_attemted,它们代表某种行为出现与否. 最后的字段表示类别标号, 大多数为 normal.

在建立监测模型时, 由于每个特征的属性值, 和阈值不同, 我们需要将数据集进行数据归一化 (数据标准化) 处理, 数据归一化的标准采用的是 z-score 归一化方法, z-score 方法是基于数据集的均值$\bar{A}$和标准差 σ 计算归一化后的结果, 计算方式如公式 (1) 所示:

$$v_i' = \frac{v_i - \bar{A}}{\sigma} \tag{1}$$

其中, v_i为数据集中每条数据的原始值, v_i'为数据集中规范化后的值.

实验首先经过 Kafka 客户端读取数据集特征数据通过创建话题的方式生产主题, 发送给 Spark Streaming 消费, 这里使用 Direct 方式读取并计算分析, 将特征数据以 DBSCAN 聚类学习算法进行聚类, 使用 Spark-Mlib 中的 DbscanModel 的变体 StreamingDbsan[16]. StreamingDbscan 模型可以根据增量对簇进行更新. 我们分别就网络流量异常监测的准确性和平台计算的实时性进行测试. 准确率通过合并各个 SparkStreaming 输出数据来计算. 计算每个类簇所含的主要攻击种类个数与数据总数的比值.

某个类簇的准确率 p 的计算公式如公式 (2) 所示:

$$p = \frac{w}{m} \tag{2}$$

其中, m 为类簇中数量占第一位的数据总数, 即主要攻击的类型个数, w 为类簇的数据总数.

数据的总准确率 P 的计算公式如公式 (3) 所示:

$$P = \frac{M}{W} \tag{3}$$

其中, M 为所有类簇中数量占第一位的数量总数, W 为所有类簇的所有数据的总和.

表 2 是经过 SparkStreaming 结合 Dbscan 数据聚类分析的得出的结果.

表 2 流量数据聚类检测结果

簇类编号	0	1	2	3	4	5	6	...
攻击类型	Normal	Neptune	Smurf	Teardrop	Normal	Neptune	IP sweep	...
本攻击记录数	5340	2252	548	970	8019	10 411	1135	...
其他记录数	423	4	0	0	354	3	327	...
记录总数	5763	2256	548	970	8463	10 414	1462	...
监测准确率 (%)	92.66	99.82	100	100	95.82	99.92	77.63	...

从表 2 可以看出, 经过聚类分析将数据分为 19 类, 通过公式 (3) 可以得出总的准确率 P 为 97.48%. 准确率较高.

实验分别在云计算和流计算处理平台, 分别以每 100 万条数据, 5 个测试等级对应时间出来开销, 分别测试并对最终获得结果, 从图 8 所示的实验结果可知, 与云计算方式的系统架构对比, 使用流计算的系统框架具备了分布式流处理的高吞吐的性能, 能够满足海

量数据实时处理分析的性能需求.

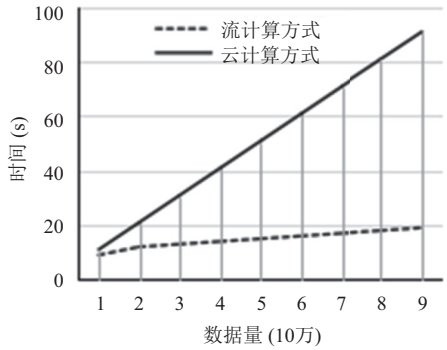

图 8 云计算方式与流计算方式吞吐量对比

5 结语

本文提出了基于流计算的处理方式, 针对电网调度数据网海量数据监测分析, 构建实时监测分析平台, 兼具高吞吐量高实时性及容错性和可扩展性的优势, 该系统基于电网调度数据网流量数据实现了流量异常的监测, 结合流计算技术实现了海量实时数据的计算分析处理及存储的需求, 同时为电力调度网的自动化运维等其他需求提供有效可靠的借鉴思路. 但本文只是对已知的网络攻击进行分析, 还需加强未知类型攻击的算法模型创建, 系统仍然需要更加深入的改进.

参考文献

1 乔媛媛. 基于 Hadoop 的网络流量分析系统的研究与应用[博士学位论文]. 北京: 北京邮电大学, 2014.
2 孙朝华. 基于 Storm 的数据分析系统设计与实现[硕士学位论文]. 北京: 北京邮电大学, 2014.
3 胡俊, 胡贤德, 程家兴. 基于 Spark 的大数据混合计算模型. 计算机系统应用, 2015, 24(4): 214–218.
4 Guller M. Machine learning with spark. Guller M. Big Data Analytics with Spark. Berkeley, CA, USA: Apress, 2015. 153–205.
5 罗乐, 刘轶, 钱德沛. 内存计算技术研究综述. 软件学报, 2016, 27(8): 2147–2167. [doi: 10.13328/j.cnki.jos.005103]
6 Zaharia M, Chowdhury M, Franklin MJ, *et al*. Spark: Cluster computing with working sets. Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing. Boston, MA, USA. 2010. 10.
7 王震, 陈亮. 基于 Kafka 消息队列的电网设备准实时数据接入方法研究. 山东电力技术, 2015, 42(6): 41–43.
8 夏俊鸾, 邵赛赛. Spark Streaming: 大规模流式数据处理的新贵. 程序员, 2014, (2): 44–47.
9 耿嘉安. 深入理解 Spark. 北京: 机械工业出版社, 2016.
10 王海凤. 工业控制网络的异常检测与防御资源分配研究[硕士学位论文]. 杭州: 浙江大学, 2014.
11 柏骏, 夏靖波, 吴吉祥, 等. 实时网络流量分类研究综述. 计算机科学, 2013, 40(9): 8–15.
12 Pentreath N. Machine Learning with Spark. Bermingham: Packt Publishing, 2014.
13 杨晨光, 马永征. 基于 Spark 的大规模网络流量准实时分类方法. 科研信息化技术与应用, 2016, 7(2): 25–34.
14 黄琼. 面向流量特征分析的数据流突发事件检测技术的研究与实现[硕士学位论文]. 长沙: 国防科学技术大学, 2008.
15 于晓聪. 基于网络流量分析的僵尸网络在线检测技术的研究[博士学位论文]. 沈阳: 东北大学, 2011.
16 Bell J. Apache spark. Bell J. Machine Learning: Hands-on for Developers and Technical Professionals. New York, USA: John Wiley & Sons, Inc, 2015. 275–314.