·技术 / TECHNOLOGY·

基于 Spark 的大规模网络流量准实时分类方法

杨晨光[1,2]，马永征[2]

1. 中国科学院大学，北京 100049
2. 中国科学院计算机网络信息中心，北京 100190

摘 要： 大数据时代催生了互联网流量的指数级增长，为了有效地管控网络资源，提高网络安全性，需要对网络流量进行快速、准确的分类，这就对流量分类技术的实时性提出了更高的要求。目前，国内外的网络流量分类研究大多是在单机环境下进行的，计算资源有限，难以应对高速网络中的 (准) 实时流量分类任务。本文在充分借鉴已有研究成果的基础上，吸收当前最新的思想和技术，基于 Spark 平台，有机结合其流处理框架 Spark Streaming 与机器学习算法库 MLlib，提出一种大规模网络流量准实时分类方法。实验结果表明，该方法在保证高分类准确率的同时，也具有很好的实时分类能力，可以满足实际网络中流量分类任务的实时性需求。

关键词： Spark；流量分类；大规模；准实时；机器学习

doi: 10.11871/j.issn.1674-9480.2016.02.004

Quasi-realtime Classification Method for Large-Scale Network Traffic Based on Spark

Yang Chenguang[1, 2], Ma Yongzheng[2]

1. University of Chinese Academy of Sciences, Beijing 100049, China
2. Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China

Abstract: In big data era, the internet traffic presents an exponential growth. In order to effectively control network resources and improve network security, internet traffic should be classified quickly and accurately, which leads to a higher requirement for real time performance of the traffic classification technology. At present, the classification of network traffic were carried out in the stand-alone environment for most of researches, so the computing resources were too limited to

respond to real-time or quasi-realtime classification of internet traffic in the high-speed network. In this paper, with reference to the existing research results and the latest theories and technologies, based on the Spark platform, combining the flow processing framework Spark Streaming with machine learning algorithm library MLlib, a quasi-realtime classification method of large scale network traffic was proposed. The experimental result showed that the proposed method guarantees high classification accuracy, and it has a good capacity of real-time classification, which meets the real-time requirements of the traffic classification in real network.

Keywords: Spark; traffic classification; large-scale; quasi-realtime; machine learning

引言

随着互联网的高速发展，不断增加的用户数量和层出不穷的新兴业务，使得互联网流量数据激增，大规模流量分类已经成为紧迫的需求。对于网络服务提供商，为了提供更好的网络服务质量 (Quadlity of Service，QoS)，减少网络拥塞状况的发生，需要及时获知当前占据主要带宽的流量类型并快速地做出反应，针对不同应用提供不同级别的服务质量 [1]。此外，网络攻击等危害网络安全的行为具有突发性，需要及时发现可疑网络流量并采取网络防护措施，以提高网络安全性。这就对流量分类技术提出了更高的要求，即在大规模网络流量分类时，不仅要保证较高的识别准确率，而且要减少分类需要的代价，快速进行分类判定。

传统的网络流量分类方法包括端口识别技术、特征字段分析等。然而，由于越来越多的网络应用采用动态端口、多重封装和加密等技术，使得这些方法不再可靠 [2]。为了克服这些不足，许多研究者开始着眼于机器学习方法，即基于网络流的某些统计特征，使用机器学习算法对未知类型的流量进行分类，这也逐渐成为网络流量分类领域一个重要的研究方向。机器学习算法可以分为无监督学习、有监督学习和半监督学习，其中无监督学习的预测准确性一般，而半监督学习的预测时间又相对较长，因此，在网络流量分类时，通常使用分类算法 (有监督学习)，以保证快速、准确的预测 [3]。目前，基于机器学习的流量分类研究大多是在单机环境下进行的，难以满足实际网络中大规模流量分类任务的实时性需求，为了提高分类系统的数据处理能力，采用分布式计算是一个有效的途径。

Spark 是继 Hadoop 之后的新一代大数据分布式计算框架，由 UC Berkeley AMP Lab 开发，它拥有 Hadoop 所具有的优点，不同之处在于 Job 的中间结果可以保存在内存中，不再需要读写 HDFS，因此 Spark 能更好地适用于机器学习等需要迭代的 MapReduce 算法 [4]。MLlib 是 Spark 对常用机器学习算法的实现库，涵盖分类、聚类、回归、降维等，分类算法中适用于多类分类问题的有 logistic 回归、决策树、随机森林和朴素贝叶斯，其中 logistic 回归必须由 L-BFGS 最优化算法求解，SGD (Stochastic Gradient Descent) 只适用于二分类 logistic 回归。此外，Spark 还提供了大规模流处理框架 Spark Streaming，它拥有完善的容错机制，具备准实时性和高吞吐量的特点，其内部处理机制是将实时接收到的流数据按照 Batch Size (最小可选取在 0.5 至 2 秒之间) 分解成一系列短小的批处理作业 (Discretized Stream or DStream)，每个 DStream 内部都由一组连续的 RDD (Resilient Distributed Dataset) 来表示，然后将 Spark Streaming 中对 DStream 的 Transformation 操作转化为 Spark 中对 RDD 的 Transformation 操作 [5]，因而，它可以与 MLlib (机器学习) 以及 GraphX (图计算) 完美结合。

本文基于 Spark 分布式计算平台，有机结合其子框架 Spark Streaming 与 MLlib，选取 Moore_Set 作为实验数据并采用 FCBF (Fast Correlation-Based Filter) 算法获取优质特征子集，实现对大规模网络流量的准实时分类，最后从准确性和实时性等方面对系统进行评估。

1 相关研究现状

近年来，基于机器学习的网络流量 (准) 实时分类研究已经取得了一定的成果。Meifeng Sun 等人 [6] 自定义了从数据包头部获取的流特征 ACK-len ab 和 ACK-len ba (只需要保存网络流的前几个数据包，极大地节省了储存空间)，使用 C4.5 决策树对 Moore 数据集和自采流量数据集 (包含WWW、MAIL、FTP、P2P) 进行分类，实验结果表明，该方法的分类准确率达到 97% 以上，而且受流量包到达顺序的影响较小，能有效应对高速网络中由于网络拥塞引起的流量包失序问题，对实时网络流量分类研究极具借鉴意义。Jun Li 等人 [7] 使用决策树算法 C4.5、REPTree 对 P2P 流量和非 P2P 流量进行二类分类，其中C4.5的建模时间不超过 4s，REPTree 不超过 1s，实验证明，这两种算法每秒都可以处理大约 10 万个网络流，且分类准确率都达到 90% 以上，此外，该方法具有较强的灵活性和健壮性，能根据传输特性、行为特征等信息识别未知的 P2P 流量。Zhu Li 等人 [8] 使用优化的 SVM 算法对包含 7 种应用类型的网络流量进行分类，通过实验发现，建模时间、预测时间与所选流特征的数量成正比关系，精简特征数量并不会使分类准确性大幅下降，该方法每分钟能处理 6 万至 8 万个网络流 (取决于特征数量)，保守估计可以满足网速为 100Mbps 网络中的实时流量分类任务。Wei Li 等人 [9] 提出了基于 Naïve Bayes 的准实时流量分类框架 ANTc，包含数据包捕获、流构造、特征提取 (从网络流的前几个数据包中获取 10 个具有较强分类能力的特征，有效降低了时间开销) 和分类 4 个模块，实验证明，ANTc 进行在线流量分类的准确性和离线环境相差无几，能够有效实现对网络流量的准实时分类。Tavallaee 等人 [10] 针对在线网络流量分类提出一种混合分类机制，先使用基于载荷的方法识别网络流量，然后对剩余的未知流量使用机器学习算法进行预测，实验结果表明，基于 J48 决策树的混合流量分类方法效果最好 (权衡分类准确率、模型训练速度以及预测速度)，拥有较强的实时分类能力。王涛等人 [11] 提出一种 BRSVM 算法并用于大规模网络流量分类，与传统的 SVM 相比，BRSVM 通过对训练样本集进行比特压缩与聚合，有效缩减了训练样本规模，可大幅提高SVM的训练速度与预测速度，而且其分类准确率也保持在较高水平，实验证明，该方法对大规模网络流量具有一定的实时分类能力。

然而，目前的网络流量分类研究绝大多数是在单机环境下进行的，很难满足高速网络环境中的大规模流量分类任务，因此，许多学者开始把视线转向分布式计算平台。乔媛媛 [12] 设计了一个基于 Hadoop 平台的离线流量分析系统 FLAS，以解决大规模流量数据的存储和分析难题，实验证明，该系统的数据处理能力极强，此外，也具有很好的可靠性和可扩展性。刘勇等人 [13] 针对海量网络流量数据，提出一种基于 MapReduce 编程模型的网络流特征计算方法，实验结果表明，该方法与传统的单机计算方式相比性能优势明显，可以适用于大规模网络流量分类任务。

2 基于 Spark 的大规模网络流量准实时分类方法

Spark 作为通用分布式计算框架的新贵，相比于 Hadoop 提供了更加强大的并行运算能力，而且非常适合需要多次迭代的算法 (迭代计算速度比 Hadoop 快 100 倍以上)，因此在机器学习方面有着无与伦比的优势，同时 Spark 拥有出色的容错和调度机制，能够确保系统的稳定运行。本文基于 Spark 平台，采用 FCBF (Fast Correlation-Based Filter) 算法从 Moore_Set 中选取优质特征子集，有效降低特征空间复杂度，接着使用 Spark MLlib 训练 logistic 回归分类模型 (由 L-BFGS 最优化算法求解得到)，最后结合流处理框架 Spark Streaming，实现对大规模网络流量的准实时分类，整体流程如图 1 所示。

2.1 FCBF (Fast Correlation-Based Filter) 特征选择算法

特征选择是一个重要的机器学习预处理阶段，它根据特定的评估标准，消除类别相关性弱和冗余的

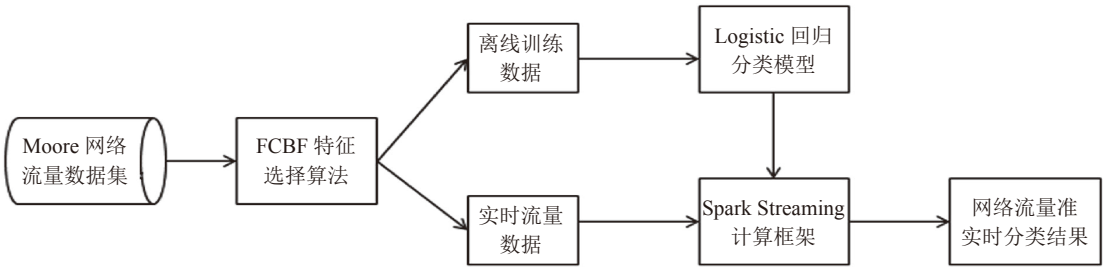

图1　基于 Spark 的大规模网络流量准实时分类流程

Fig. 1　The mechanics of quasi-realtime classification for large-scale network traffic based on Spark

特征，有效降低特征空间的维度，从而提升机器学习算法的性能。特征选择算法可以大致分为两类：过滤器模式 (filter) 和封装器模式 (wrapper)，过滤器模式将数据本身的特性作为选取特征子集的评估标准，而封装器模式则是根据机器学习算法的准确率进行选择。当特征数量很大时，封装器模式的计算效率会大幅下降，因此通常选择过滤器模式，FCBF (Fast Correlation-Based Filter) 算法 [14] 是典型的基于过滤器模式的特征选择算法，它将 *SU* (Symmetrical Uncertainty) 作为优质特征的评估标准，在基于特征关联性分析的基础上选择类别相关性强且非冗余的特征子集，*SU* 的定义如下：

$$SU(X, Y) = 2\left[\frac{IG(X \mid Y)}{H(X)+H(Y)}\right] \tag{1}$$

其中 $IG(X \mid Y)$ 称为信息增益 (information gain)，表示得知变量 Y 后使得变量 X 不确定性减少的程度，它具有对称性，即 $IG(X \mid Y)=IG(Y \mid X)$，计算公式如下：

$$IG(X \mid Y)=H(X)-H(X \mid Y) \tag{2}$$

$H(X)$ 表示变量 X 的熵，即 X 的不确定性，定义为：

$$H(X)= -\sum_i P(x_i) \log_2 P(x_i) \tag{3}$$

$H(X|Y)$ 表示变量 Y 给定条件下变量 X 的条件熵，即已知变量 Y 的条件下变量 X 的不确定性，定义为：

$$H(X|Y)= -\sum_j P(y_j) \sum_i P(x_i \mid y_j) \log_2 P(x_i \mid y_j) \tag{4}$$

实现 FCBF 算法需要解决两个问题：(1) 如何判断一个特征是否与类别相关；(2) 如何判断与类别相关的特征之间是否存在冗余。

第一个问题直接通过 *SU* 的阈值 δ 来判断即可。假设数据集中的每个实例有 N 个特征，C 表示其中有一个类别，令 $SU_{i,c}$ 表示特征 f_i 与类别 C 的相关性 (C-correlation)，使用阈值 δ 在大量候选特征中进行初步选择，获取与类别高度相关的特征子集 S'，S' 满足 $\forall f_i \in S'$，$1 \le i \le N$，$SU_{i,c} \ge \delta$。

第二个问题需要在 C-correlation 基础上进行判断。令 $SU_{i,j}$ 表示特征 f_i 与 f_j 的相关性 (F-correlation)，对于 $f_i \in S'$，$f_j \in S'$ $(j \ne i)$，若 $SU_{i,j} \ge SU_{i,c}$，则称 f_j 是 f_i 的冗余特征，令 S_{P_i} 表示 f_i 的所有冗余特征构成的集合，如果 $S_{P_i} \ne \varnothing$，就将其划分为 $S_{P_i}^+$ 和 $S_{P_i}^-$ 两部分，$S_{P_i}^+=\{f_j \mid f_j \in S_{P_i}, SU_{j,c} > SU_{i,c}\}$，$S_{P_i}^-=\{f_j \mid f_j \in S_{P_i}, SU_{j,c} \le SU_{i,c}\}$。若 $S_{P_i}^+=\varnothing$，就保留 f_i 并删除 $S_{P_i}^-$ 中的特征；否则逐一处理 $S_{P_i}^+$ 中的特征，以决定是否删除 f_i，并根据 S' 中的其他特征来判断是否删除 $S_{P_i}^-$ 中的特征。通过上述算法流程，对初选后剩余的特征做进一步筛选，最终获得一个类别相关性很强且特征间相关性很弱的优质特征子集。

本文使用 FCBF 特征选择算法从 Moore_Set 中筛选出优质特征，有效降低特征空间复杂度，大幅提升 logistic 回归算法的分类速度，为大规模网络流量的准实时分类奠定了基础。

2.2　Logistic 回归算法

使用一条线对数据点进行拟合的过程称为回归，当这条线是直线时就是简单的线性回归问题，然而对于分类问题，线性回归是无法解决的。logistic 回归以线性回归为基础，根据逻辑函数做出类别预测，能够很好地解决二分类问题，其公式如下：

$$f(z) = \frac{1}{1+e^{-z}} \tag{5}$$

其中 $z=\vec{w}\vec{x}$，$\vec{w}$ 为权值向量，$\vec{x}$ 为特征向量，$f(z)$

也叫 sigmoid 函数，它能将实数范围内的值映射到 (0, 1) 区间。默认情况下，如果 $f(\vec{w}\vec{x})>0.5$，就将该数据点分入第 1 类，否则分入第 0 类，因此，也可以把 logistic 回归看成一种概率估计。

在网络流量分类中，应用类别远不止两类，因此需要将二分类 logistic 回归泛化为多分类 logistic 回归 [15]。解决多分类问题时，算法会输出一个多项式 logistic 回归模型，它包含 K-1 个与“轴心”类 (MLlib 中选取第 0 类) 进行回归的二分类 logistic 回归模型，对于每个待分类的数据点，K-1 个模型都会被执行，最终选取概率最大的类会作为预测类型。多分类 logistic 回归模型如下：

$$P(y=0 \mid \vec{w},\ \vec{x}) = \frac{1}{1+\sum_{i=1}^{K-1} exp\ (\vec{w}_i\vec{x})} \tag{6}$$

$$P(y=k \mid \vec{w},\ \vec{x}) = \frac{exp\ (\vec{w}_i\vec{x})}{1+\sum_{i=1}^{K-1} exp\ (\vec{w}_i\vec{x})},\ k=1, 2, ..., K\text{-}1 \tag{7}$$

模型中的权值矩阵 $\vec{w} = (\vec{w_1},\ \vec{w_2},\ \cdots,\ \vec{w_{K-1}})^T$，它是一个 $(K\text{-}1)\times(N+1)$ 维矩阵，其中 N 代表特征的数量。

logistic 回归的模型参数估计通常使用极大似然估计，其对数似然函数如下：

$$L\ (\vec{w},\ \vec{x})= \sum_{k=1}^{N} \log P\ (y=k \mid \vec{w},\ \vec{x_k}) \tag{8}$$

可以进一步表示为：

$$L\ (\vec{w},\ \vec{x})= \sum_{k=1}^{N} [\alpha(y_k)\ log\ P(y=0 \mid \vec{w},\ \vec{x_k})+(1-\alpha(y_k))\ log\ P\ (y=k \mid \vec{w},\ \vec{x_k}] \tag{9}$$

其中，当 $y_k=0$ 时，$\alpha(y_k)=1$；否则 $\alpha(y_k)=0$。带入 (6) 式和 (7) 式，整理得：

$$L\ (\vec{w},\ \vec{x})= \sum_{k=1}^{N}[(1-\alpha(y_k))\vec{w_{y_k}}\vec{x}-log(1+\sum_{i=1}^{K-1}exp((\vec{w_i}\vec{x}))] \tag{10}$$

对 $L\ (\vec{w},\ \vec{x})$ 求极大值，可以得到 $\vec{w}$ 的估计值，这样问题就转变为以对数似然函数为目标函数的最优化问题。Spark MLlib 为 logistic 回归提供了两种求解最优化问题的方法：随机梯度下降 (Stochastic Gradient Descent) 和 L-BFGS，其中只有 L-BFGS 支持多分类 logistic 回归。L-BFGS 是一种拟牛顿算法，其二阶导数的 Hessian 矩阵不需要直接求解，而是通过梯度来进行估计，收敛速度比随机梯度下降更快。

Spark MLlib 中提供了很多种分类算法，其中支持多分类问题的有：logistic 回归、决策树、随机森林以及朴素贝叶斯。本文采用 logistic 回归算法是在权衡分类准确率和分类速度的基础上做出的选择，以保证大规模网络流量准实时分类的整体效果。另外，从 Spark 1.3.0 版本开始，MLlib 新增加了模型的 save 和 load 方法，为使用离线训练的模型对在线网络流量进行分类提供了可能。

2.3 Spark Streaming 流处理框架

传统的分布式流处理系统大多采用连续算子架构 [16]，其工作方式如下：“源”算子从上游系统接收流数据并发送给一系列工作节点，每个工作节点运行一个或多个连续算子，每个连续算子一次处理一条 record 并通过管道传送给其他算子，最终“沉”算子将处理完的 records 输出到下游系统。连续算子是一种相对简单、自然的模型，然而，随着数据规模的不断扩大以及越来越复杂的实时分析，这个传统的架构面临了严峻的挑战。

Spark Streaming 提供了一套高吞吐量的、可容错的准实时大规模流处理框架 [17]，与每次处理一条 record 的传统流数据处理方式相比，Spark Streaming 是将流数据离散化，使之能够进行亚秒级的微型批处理，其整体架构如图 2 所示。详细来说，它可以从基础数据源 (文件系统、Socket 链接等) 或者高级数据源 (Kafka、Flume 和 Twitter 等) 接收流数据，按照 batch size 拆分成一系列批处理作业 (DStream) 并缓存至 Spark 各工作节点的内存中，DStream 是

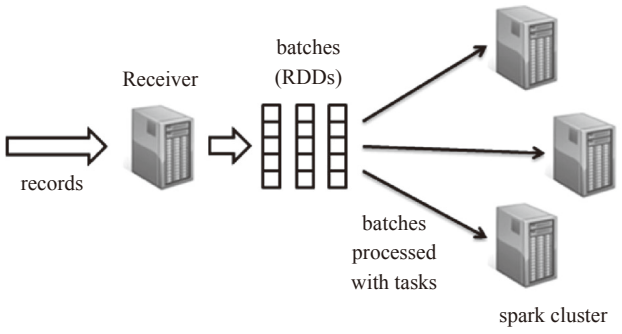

图2 Spark Streaming 架构图

Fig. 2 Architecture of Spark Streaming

Spark Streaming 为连续流数据提供的高层次抽象描述，其内部由一组 RDD (Resilient Distributed Dataset) 序列来表示，然后通过低延迟 Spark 引擎使用诸如 map、reduce、join 和 window 等高级方法进行复杂的算法处理，处理时将 Spark Streaming 中对 DStream 的 Transformation 操作转化为 Spark 中对 RDD 的 Transformation 操作，处理结束后将中间结果保存到内存中，最后根据业务需求对中间结果进行叠加或者使用 Output 操作输出到外部系统，如文件系统、数据库等，Spark Streaming 应用程序的工作流程如图 3 所示。值得注意的是，与传统的连续算子模型不同，Spark 会根据可用资源情况和数据所在位置向每个工作节点动态分配任务，从而保证了负载均衡和快速故障恢复。

Spark Streaming 中存在两种需要特别关注的故障类型，工作节点故障和驱动节点故障，它们都会导致内存中数据的丢失，除此之外，后者还会导致 SparkContext 的丢失。为了保证 24/7 不间断地工作，Spark Streaming 提供了完善的容错机制。首先，Spark Streaming 能够从可靠的文件系统 (HDFS、S3 等) 中读取数据，当出现故障时，所有的数据都可以被重新处理，不会发生丢失。然而，大多数情况下，Spark Streaming 都是从网络中接收数据，为了保证容错能力，会把数据在集群中的多个工作节点间进行拷贝 (默认拷贝 2 份)，这避免了工作节点故障导致的数据丢失，但是如果出现驱动节点故障，数据仍然会丢失。为此，从 1.2 版本开始，Spark 提供了预写日志 (write ahead logs) 功能，它会把接收到历史数据进行容错储存，改进了故障恢复机制，从而保证了零数据丢失。此外，为了能及时从逻辑无关的故障 (系统错误、JVM 崩溃等) 中恢复，Spark Streaming 还提供了 checkpoint 功能，具体分为元数据 checkpoint 和数据 checkpoint 两种，前者保存的是定义流计算的相关信息，主要用于从驱动节点故障中恢复，而后者保存的是生成的 RDD，供有状态的 Transformation 操作 (updateStateByKey、reduceByKeyAndWindow 等) 使用，数据均保存在可靠的文件系统中。需要注意的是，必须合理设置 checkpoint 间隔，间隔太短会大幅降低吞吐量，反之会使单次任务规模剧增，也会影响性能。

为了方便对 Spark Streaming 应用程序的日常监控，Spark 专门为 Streaming 提供了可视化 Web UI，默认端口为 4040。从中可以看到接收器 (Receiver) 的运行状态 (是否存在异常、接收 record 的数量等)，也能获得最近 1 000 个批处理作业 (Batch Job) 的输入速率 (Input Rate)、调度延迟 (Scheduling Delay)、处理时间 (Processing Time) 和总延迟 (Total Delay) 等信息，其中，调度延迟反映了每个批处理作业的排队时间，若其持续增加就表明系统不能及时处理接收到的流数据，需要进行一定的性能优化。首先，可以优化批作业处理时间，主要途径如下：(1) 提高数据接收和数据处理的并行度；(2) 降低数据序列化、反序列化造成的 CPU 和内存开销；(3) 降低任务分发开销。另一方面，为了让 Spark Streaming 应用程序稳定运行，必须选取合适的 batch size，通常先选择一个较大的值，再根据延迟情况逐步调整。

此外，为了满足复杂的业务需求，Spark Streaming 提供了对高级分析方式的支持，既可以结合 DataFrame 和 SQL 语句对流数据进行 SQL 查询，也可以将 MLlib (机器学习) 和 GraphX (图计算) 应用于流数据分析中。值得注意的是，MLlib 提供了一些流式机器学习算法 (Streaming Linear Regression、Streaming KMeans 等)，可以在对流数据进行学习的同时将模型应用于流数据分析，而对于其他机器学习

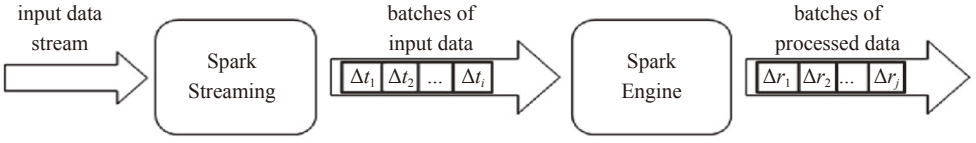

图3　Spark Streaming 应用程序工作流程图

Fig. 3　Workflow of Spark Streaming application

算法，必须使用历史数据离线训练模型，之后再对流数据进行在线分析。但遗憾的是，当前 Spark 提供的流式机器学习算法具有很强的局限性，不适用于网络流量分类任务。

本文基于 Spark 分布式计算平台，充分利用 Spark Streaming 出色的流数据处理能力并与 MLlib 相结合，实现对大规模网络流量的准实时分类。相比于其他主流的分布式框架，本文所采用的方法优势明显，Hadoop 虽然可以适用于海量数据分析，但实时性却无法得到保证，而 Storm 尽管可以保证实时性，却难以处理复杂的业务逻辑，Spark Streaming 以准实时的方式在两者之间取得一个平衡点。

3 实验与结果分析

3.1 实验环境

本文为了保证 Spark Streaming 网络流量分类程序能够长期高效、稳定的运行，进行了许多与性能优化以及容错相关的配置和参数调整，之后再打包并提交到 Spark 集群上，任务部署采用 Standalone 模式。运行时，程序会加载 logistic 回归模型 (由 MLlib 中的 LogisticRegressionWithLBFGS 算法训练生成) 并使用 textFileStream 方法监控指定目录，对目录中新增的网络流量数据进行分类。本实验所使用的 Spark 集群搭建在 Hadoop 的基础上且采用全分布式安装，集群中共有 4 个节点，将其中一个配置为 master，其他三个配置为 slave，硬件配置情况如表 1 所示，软件配置情况如表 2 所示。

3.2 实验数据及预处理

实验采用剑桥大学 Moore 教授等人 [18] 提供的网络流量数据集，记为 Moore_Set，它包括 10 个子数据集 (Moore_Set 1~10)，采自于一天中不同时间段内流经某研究机构网络出口的双向网络流量，采样时间均为 28 分钟左右，共包含 377 526 个网络流样本，分为 WWW、MAIL 等 10 种应用类型。原始的 Moore_Set 为 arff 格式，专门供数据分析软件 Weka 使用，为了应用于 MLlib 和 Spark Streaming，需要修改数据格式并填充缺省值 (数据中的问号)，此外，由于数据中 INTERACTIVE 和 GAMES 网络流样本数目较少，不具有代表性，因此本实验剔除了这两种类型

表1 Spark 集群硬件配置情况

Table 1 Hardware configuration of Spark cluster

	CPU	内存
节点1	Intel Xeon E5649	128GB
节点2	Intel Xeon E5649	128GB
节点3	Intel Xeon E5620	128GB
节点4	Intel Xeon E5620	128GB

表2 Spark 集群软件配置情况

Table 2 Software configuration of Spark cluster

操作系统	Hadoop	Spark
CentOS 5.5	Hadoop 2.6.0	Spark 1.4.1

表3 实验数据集统计信息

Table 3 Statistical information of experimental data set

流量类型	具体应用	流数目	百分比（%）
WWW	HTTP, HTTPS	328 091	86.93
MAIL	IMAP, POP2/3, SMTP	28 567	7.569
BULK	FTP	11 539	3.057
DATABASE	POSTGRES, SQLNET, Oracle, INGRES	2 648	0.702
SERVICES	X11, DNS, IDENT, LDAP, NTP	2 099	0.556
P2P	KaZaA, BitTorrent, GnuTella	2 094	0.555
ATTACK	Internet worm and virus attacks	1 793	0.475
MULTIMEDIA	Windows Media Player, Real	1 152	0.305
总计	21种	377 408	100

的流量数据，实验数据的具体情况如表 3 所示。

值得注意的是，Moore_Set 中的每个网络流样本都包含248项流特征，其中有近 100 项是通过傅里叶变换得到的，然而，实际网络环境中的流量规模巨大，提取如此多的特征将带来沉重的计算和存储开销，同时也会加重分类算法的负担。为了满足网络流量分类的实时性需求，本文采用 FCBF 特征选择算法从 Moore_Set 中获取类别相关性强且非冗余的优质特征子集来描述每个网络流，大幅提高分类效率，所选流特征如表 4 所示。完成数据预处理后，分别从 Moore_Set 1~10 中随机抽取每类网络流样本的 10% 构成训练集，共 37 783 条数据，剩余的网络流样本作为测试集，共 339 625 条数据。

3.3 实验结果及分析

实验流程如下：首先使用 MLlib 中的 LogisticRegressionWithLBFGS 算法训练 logistic 回归模型并保存到 HDFS 上，然后启动 Spark Streaming 网络流量分类程序 (batch size 设置为 1 秒)，并每 0.1 秒随机发送 1 250 条测试数据到其监控目录中，程序会定时加载 logistic 回归模型对新增的流量数据进行分类。

实验采用整体准确率 (Overall accuracy) 作为 logistic 回归算法分类效果的评价指标，它体现了算法对网络流量的整体分类能力，其定义如下：

$$\text{Overall accuracy}=\frac{\sum_{i=1}^{n}TP_i}{\sum_{i=1}^{n}(TP_i+FP_i)} \tag{11}$$

其中，TP_i(true positive) 表示被分类模型预测为 C_i 类且实际上也属于 C_i 类的网络流样本数；FP_i(false positive) 表示被分类模型预测为 C_i 类，但是实际上不属于 C_i 类的网络流样本数。

当 Spark Streaming 网络流量分类程序运行 24 小

表4 FCBF 算法筛选出的流特征

Table 4 Traffic feature filtered by FCBF algorithm

编号	特征名	描述
1	Server Port	Port Number at server
2	q1_data_ip	First quartile of total bytes in IP packet
3	min_data_control	Minimum of control bytes in packet
4	zwnd_probe_pkts	Count of all the window probe packets
5	zwnd_probe_bytes	Total bytes of data sent in the window probe packets
6	outoforder_pkts	Count of all the packets that arrive out of order
7	pushed_data_pkts	Count of all the packets with the PUSH bit set in the TCP header
8	FIN_pkts_sent	Count of all the packets with the FIN bits set in the TCP header
9	urgent_data_pkts	Total number of packets with the URG bit turned on in the TCP header
10	mss_requested	Maximum segment size requested as a TCP option in the SYN packet
11	init_window_pkts	Total number of packets sent in the initial window
12	full_sz_RTT	Total number of full-size RTT samples
13	post_loss_acks	Total number of ack packets received after losses
14	max_retrans	Maximum number of retransmissions
15	min_retr_time	Minimum time between any two (re)transmissions
16	q1_data_ip	First quartile of total bytes in IP packet
17	med_data_control	Median of control bytes in packet
18	q1_data_control	First quartile of control bytes in packet

时后，随机抽取 100 个连续的批处理作业 (每个批处理作业包含 12 500 个网络流样本)，分别统计其整体准确率，结果如图 4 所示。从图中可以看出，logistic 回归算法的整体准确率主要集中在 92% 到 95% 之间，波动幅度不大，具有良好的分类准确率和稳定性。

通过 Spark 提供的可视化 Web UI 可以监控 Spark Streaming 网络流量分类程序的运行状态，从中获取这 100 个流量分类批作业的处理时间，如图 5 所示。可以发现，批作业处理时间在 300 毫秒附近小幅度震荡，最多也不超过 400 毫秒，远小于 batch size (1 秒)，说明分类程序运行非常稳定，完全能够胜任每秒 12 500 个网络流的分类工作。假设网络流的平均大小为10KB (保守估计)，则实验相当于模拟了 1Gbps 网络环境中的准实时流量分类任务，实验结果表明，本文采用的方法具有很好的实时分类能力，可以满足实际应用需求。

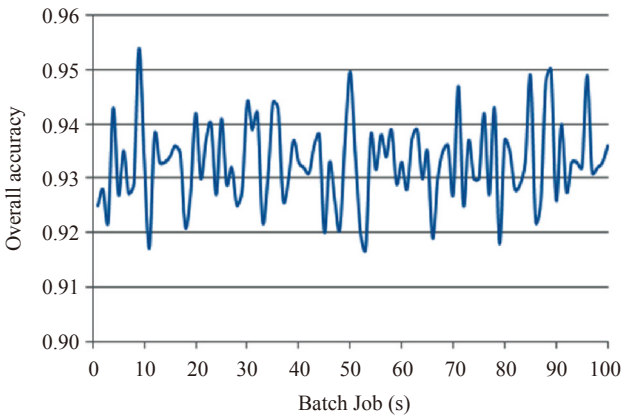

图4　整体分类准确率

Fig. 4　Overall accuracy of network traffic classification

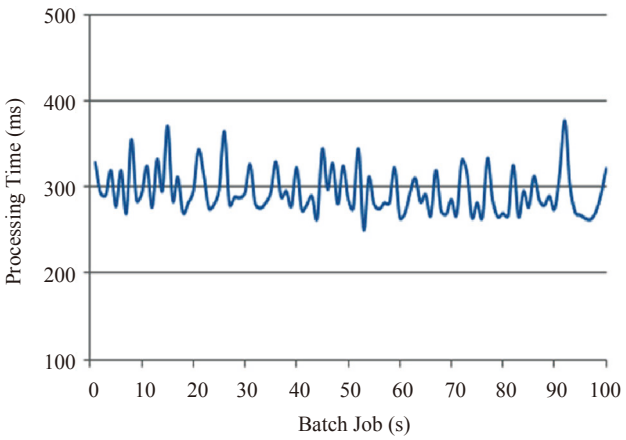

图5　流量分类批作业处理时间

Fig. 5　Batch processing time of network traffic classification

4　结束语

在互联网快速发展和高速网络日益普及的今天，网络流量规模呈爆发式增长，这给传统的流量分类方法带来了巨大的挑战。目前，国内外针对高速网络环境的 (准) 实时流量分类研究成果较少，如何权衡分类准确率和分类速度仍然是有待突破的难题，本文提出一种基于 Spark 分布式计算平台的大规模网络流量准实时分类方法，将流处理框架 Spark Streaming 与机器学习算法库 MLlib 有机结合起来，大幅提高分类效率，极具实际应用价值。此外，分布式系统自身的特性为该方法带来了良好的扩展性，可以通过增加计算节点的数量提升数据处理能力，以适应高速网络的发展趋势。值得注意的是，为了应对实际网络中不断涌现的新型流量应用类别，需要持续从流数据中学习并调整分类模型，从而保证预测准确率和稳定性，但当前 Spark 提供的流式机器学习算法种类很少且都不适用于网络流量分类任务，这将是下一步的工作重点。

参考文献

[1] Callado A, Kamienski C. A survey on internet traffic identification[J]. IEEE Communications Surveys and Tutorials, 2009, 11(3): 37-52.

[2] Nguyen T T T, Armitage G. A survey of techniques for internet traffic classification using machine learning[J]. IEEE Communications Surveys and Tutorials, 2008, 10(4): 56-76.

[3] 柏骏, 夏靖波, 吴吉祥, 等. 实时网络流量分类研究综述[J]. 计算机科学, 2013, 40(9): 8-15.

[4] Pentreath N, Machine Learning with Spark[M]. [S.l.]: Packt Publishing, 2014.

[5] M Zaharia, Das T, Li H, et al. Discretized streams: fault-tolerant streaming computation at scale[C]. The 24th ACM

Symposium on Operating Systems Principles, Farmington, PA, 2013, 423–438.

[6] Sun Meifeng, Chen Jingtao. Research of the traffic characteristics for the real time online traffic classification [J]. The Journal of China Universities of Posts and Telecommunications, 2011, 18(3): 92-98.

[7] Li Jun, Zhang Shunyi, Lu Yanqing, et al. Real-time P2P traffic identification[C]. IEEE Global Telecommunications Conference, New Orleans, LA, 2008, 2474-2478.

[8] Li Zhu, Yuan Ruixi, Guan Xiaohong. Traffic classification-towards accurate real time network applications[J]. Lecture Notes in Computer Science, 2007, 4553: 67-76.

[9] Li W, Abdin K, Dann R, et al. Approaching real-time network traffic classification, RR-06-12[R]. London: Queen Mary University of London, 2006.

[10] Tavallaee M, Lu W, Ghorbani A A. Online classification of network flows[C]. The 7th Communication Networks and Services Research Conference, Moncton, NB, 2009, 78-85.

[11] 王涛, 程良伦. 基于快速 SVM 的大规模网络流量分类方法 [J]. 计算机应用研究, 2012, 29(6): 2301-2305.

[12] 乔媛媛. 基于 Hadoop 的网络流量分析系统的研究与应用 [D]. 北京: 北京邮电大学, 2014.

[13] 刘勇, 雒江涛, 邓生雄, 等. 基于 Hadoop 的网络分流和流特征计算 [J]. 电信科学, 2014, 12: 76-81.

[14] Yu Lei, Liu Huan. Feature selection for high-dimensional data: a fast correlation-based filter solution[C]. The 20th International Conference on Machine Learning, Washington, DC, 2003, 856-863.

[15] Tsai DB. Multinomial logistic regression with Apache Spark[EB/OL]. (2014-06-20) [2015-12-30]. http://www.slideshare.net/dbtsai/2014-0620-mlor-36132297.

[16] Das T, Zaharia M, Wendell P. Diving into Spark Streaming's execution model[EB/OL]. (2015-07-30) [2015-12-30]. https://databricks.com/blog/2015/07/30/diving-into-spark-streamings-execution-model.html.

[17] M Zaharia, Das T, Li H, et al. Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters[C]. The 4th USENIX Conference on Hot Topics in Cloud Computing, Berkeley, CA, 2012, 10-10.

[18] Moore A W, Zuev D, Crogan M L. Discriminators for use in flow-based classification, RR-05-13[R]. London: Queen Mary University of London, 2005.

收稿日期：2015 年 12 月 20 日

杨晨光：中国科学院大学，硕士研究生，主要研究方向为数据挖掘、分布式计算。

E-mail: *yangchenguang13@mails.ucas.ac.cn*

马永征：中国科学院计算机网络信息中心，副研究员，博士研究生，主要研究方向为大规模网络数据处理、分布式计算。

E-mail: *myz@cstnet.cn*