基于大数据的网络日志分析技术

应 毅[1] 任 凯[2] 刘亚军[3]

(三江学院计算机科学与工程学院 南京 210012)[1] (南京大学金陵学院 南京 210089)[2]

(东南大学计算机科学与工程学院 南京 210096)[3]

摘 要 传统的日志分析技术在处理海量数据时存在计算瓶颈。针对该问题,研究了基于大数据技术的日志分析方案:由多台计算机完成日志文件的存储、分析、挖掘工作,建立了一个基于 Hadoop 开源框架的并行网络日志分析引擎,在 MapReduce 模型下重新实现了 IP 统计算法和异常检测算法。实验证明,在数据密集型计算中使用大数据技术可以明显提高算法的执行效率和增加系统的可扩展性。

关键词 大数据,Hadoop,MapReduce,日志分析,异常检测

中图法分类号 TP393 **文献标识码** A

Network Log Analysis Technology Based on Big Data

YING Yi[1] REN Kai[2] LIU Ya-jun[3]

(College of Computer Science and Technology, Sanjiang University, Nanjing 210012, China)[1]

(Jinling College, Nanjing University, Nanjing 210089, China)[2]

(School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)[3]

Abstract There exists a calculation bottleneck when traditional log analysis technology processes the massive data. To solve this problem, a log analysis solution based on big data technology was proposed in this paper. In this solution, the storage and analysis, mining tasks of Log files will be decomposed on multiple computers. The open source framework Hadoop is used to establish a parallel network log analysis engine. IP statistics and outlier detection algorithm was realized with MapReduce model. Empirical studies show that the use of big data technology in data-intensive computing can significantly improve the execution efficiency of algorithms and the scalability of system.

Keywords Big data, Hadoop, MapReduce, Log analysis, Outlier detection

1 引言

经过多年的信息化建设,大型企业在内部网络中积累了大量的软硬件资源,这些计算机及网络设备在运行过程中产生了大量的日志数据。作为软件系统、硬件设备和用户行为的记录工具,日志文件在监控网络情况、检查硬件故障、保护软件安全等方面起着重要的作用。通过分析日志文件,能及时发现用户异常行为和软硬件故障,保证网络运行的稳定性和安全性。

网络日志的分析、挖掘技术普遍应用于信息安全领域,可以进行计算机取证工作[1]、发现异常的网络访问[2]、检测泛洪攻击[3]、进行防火墙的安全测评。当前,在大型局域网内部,日志文件的种类众多、格式不一、体量庞大(达到 TB、PB 级别),传统的基于关系数据库的日志分析方法性能急剧下降,无法满足海量数据的处理需求。在大数据领域,日志是广泛使用的数据采集方法之一[4],具有 4V 特征:数据体量巨大(Volume)、数据类型繁多(Variety)、有价值但密度低(Value)、处理速度快(Velocity)。因此,大数据技术是解决日志分析的更有效手段。本文提出基于 Hadoop 框架的日志分析引擎,在 MapReduce 模型下重新实现了 IP 统计算法和异常检测算法,实验证明,基于大数据的日志分析技术具有更高的执行效率和良好的可扩展性。

2 大数据技术在日志分析中的应用

2.1 大数据处理技术与工具

根据应用类型的不同,大数据的处理模式可以分为流处理和批处理两种,实时计算领域使用流处理技术,其他大部分应用(如数据挖掘、推荐系统)都依赖批处理技术[4]。Hadoop 是由 Apache 开源的大数据处理框架,它是基于批处理技术的,默认 MapReduce 是其并行计算引擎,并由 HDFS[5] 负责数据存储。由于具备数据海量存储、数据并行处理及资源调度、负载平衡、容错处理等底层管理功能,Hadoop 极大地降低了分布式程序开发的难度,得到了工业界的青睐,已经成为大数据领域事实上的标准[6]。

Hadoop 的核心是 MapReduce。MapReduce[7] 依托于无共享大规模集群系统,将计算工作分布到集群中的众多节点并行运行,它的计算依靠用户定义的 Map 函数和 Reduce 函数实现,Map 函数负责分块数据的处理,Reduce 函数对中间结果进行归约并得到最终结果。MapReduce 编程模型易于理解、易于使用,结合 Hadoop 平台合适的查询优化和索引技术,在大数据环境下,仍能保持良好的数据处理性能。因此,MapReduce 被广泛应用于海量数据的搜索、分析、挖掘和机器学习。

本文受江苏省高等学校自然科学研究面上项目(17KJB520033)资助。

应 毅(1979—),男,硕士,副教授,主要研究方向为大数据处理与数据库,E-mail:907635255@qq.com(通信作者);任 凯(1979—),女,硕士,讲师,主要研究方向为分布式计算与数据库;刘亚军(1953—),女,教授,硕士生导师,主要研究方向为软件工程与数据库应用。

2.2 日志分析引擎

本文设计了基于 Hadoop 框架的并行日志分析引擎，其中节点分为两类：MainCtrlNode(主控节点)、WorkerNode(工作节点)。两类节点的模块组成及模块作用如表 1 所列。

表 1 MainCtrlNode 和 WorkerNode 的组成及作用

	存储工作	计算工作
MainCtrlNode 的组成	NameNode 日志数据集	JobTracker 日志分析算法库
WorkerNode 的组成	DataNode	TaskTracker

NameNode 是 HDFS 的管理者，负责管理集群配置信息、文件系统命名空间、存储块复制等；DataNode 负责数据的实际存储，它将文件块保存在本地系统中。JobTracker 和 TaskTracker 采用 M/S 结构。JobTracker 负责启动、跟踪、调度各 WorkerNode 上的任务执行；TaskTracker 负责在本节点完成数据处理，并将状态和完成信息上报给 JobTracker。日志分析算法库存储已编写完成的分析数据、挖掘信息的算法，为了利用 Hadoop 并行执行的特点，这些算法必须使用 MapReduce 模型重新设计实现。日志分析引擎的整体架构如图 1 所示。

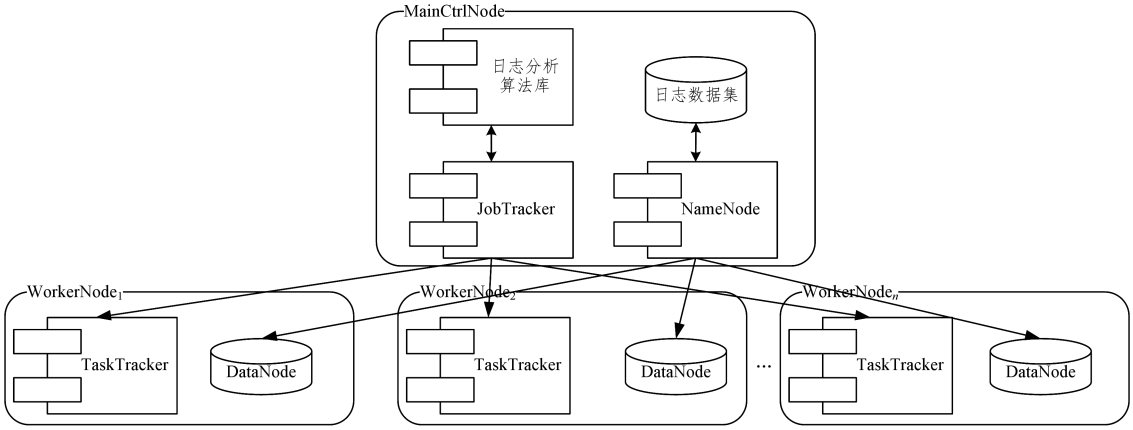

图 1 基于 Hadoop 的日志分析引擎架构

3 并行日志分析算法

作为通用的大数据处理框架，运行在其上的传统的统计分析、数据挖掘、机器学习等算法需要作出调整以适应并行计算的特点。

3.1 源 IP 统计算法

DoS 攻击及 DDoS 攻击会使服务器高负荷运转，最终导致正常服务瘫痪。对服务器的访问日志中各 IP 地址的请求次数进行统计，获取请求次数频繁的 IP 地址是检测攻击源、防御攻击的较有效方法。服务器的访问日志包含较多数据信息，日志文件通常达到 GB 数量级，传统的单机模式统计算法的时效性很差。这里借助于 MapReduce 并行计算的特点，将传统单机算法改进为并行算法。

日志数据以文件形式存入 HDFS 中，Map 函数对每一行日志数据进行分析，提取申请访问服务器的源 IP，输出的 Key/Value 是：SourceIP/1。Reduce 函数输入的 Key 是相同的 SourceIP，for 循环将 Value 数值累加，输出的 Key/Value 是：SourceIP/n，它表示同一个 IP 对服务器的请求次数。该算法步骤如算法 1 所示。

算法 1 统计源 IP 的 MR 算法

```
1. map(key,value)
2.    emit(value.SourceIP/1)
3. reduce(key,values[v1;v2;…])
4.    i=0
5.    for val in values[v1;v2;…]
6.       i++
7. emit(key/i)
```

3.2 基于密度的异常检测算法

网络中的流量日志记录了用户或设备的行为，当一个设备遭遇攻击或遭到病毒、木马入侵时，它会表现出与被入侵前不同的行为状况。分析设备的流量日志，可以鉴别异常流量，发现入侵和被攻击行为。

异常检测是发现那些行为不符合期望的数据对象，它们与大部分数据不相似、不一致，这种对象被称为异常、噪声或离群点。计算机入侵检测中的异常可能意味着入侵行为的发生。基于密度的异常检测算法[8]的思想是：对整个数据空间的密度分布进行计算，存在于低密度区域的数据会被认为是异常数据。

在对象集 D 中，对象 o 的 k 距离记为 $dist_k(o)$，是 o 与另一个对象 $p\in D$ 之间的距离 $dist(o,p)$，使得：至少有 k 个对象 $o'\in D-\{o\}$，使得 $dist(o,o')\leqslant dist(o,p)$；至少有 $k-1$ 个对象 $o''\in D-\{o\}$，使得 $dist(o,o'')<dist(o,p)$。

o 的 k 距离领域包含其到 o 的距离不大于 $dist_k(o)$ 的所有对象，记为：

$$N_k(o)=\{o'\mid o'\in D,dist(o,o')\leqslant dist_k(o)\} \tag{1}$$

对于两个对象 o 和 o'，定义 o' 到 o 的可达距离为：

$$reachdist_k(o\leftarrow o')=\max\{dist_k(o),dist(o,o')\} \tag{2}$$

对象 o 的局部可达密度为：

$$lrd_k(o)=\frac{|N_k(o)|}{\sum\limits_{o'\in N_k(o)} reachdist_k(o'\leftarrow o)} \tag{3}$$

o 的局部偏离因子(Local Outlier Factor)定义为：

$$LOF_k(o)=\frac{\sum\limits_{o'\in N_k(o)}\frac{lrd_k(o')}{lrd_k(o)}}{|N_k(o)|} \tag{4}$$

考虑到不同用户有不同的网络使用习惯，进行流量分析时，应以时间轴进行同一设备历史数据的纵向考量，通过异常检测算法判断非正常的流量情况。

每一条流量日志为日志文件中的一行，它记录设备 IP、时间、1 分钟内产生的流量、1 分钟内与之交互的 IP 个数，即：$IP,time,traffic,interactionIP$。将一条流量日志记录看作一个点，并将 $traffic$ 和 $interactionIP$ 投影到二维空间，相当

于点的横坐标和纵坐标,点之间的距离通过欧氏距离 $d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$ 衡量,使用基于密度的异常检测方法(见式(4))计算该时间点的局部偏离因子 lof。

大型企业内网的流量日志总量是非常庞大的,假设每分钟采样一次,一个工作日(8 小时)一个 IP 就会产生 480 条记录,一个具有 5000 台网络设备的企业一天就会产生 240 万条记录。传统的 LOF 算法无法处理如此海量的数据,本文对 LOF 算法进行改进,通过在集群中进行 MapReduce 并行计算完成异常检测,称之为基于 MapReduce 的 LOF 算法(Local Outlier Factor base MapReduce,LOF-MR)。

Map 函数的输入是流量日志文件,一行为一条记录:IP,$time$,$traffic$,$interactionIP$;输出的 Key/Value 是:$IP/time$,$traffic$,$interactionIP$。Reduce 函数汇聚相同 IP 的数据,输入的是 $IP/list[time,traffic,interactionIP]$;输出的 Key/Value 是:$IP$,$time/lof$,其中 lof 是 IP 所标识的设备在 $time$ 时间点的偏离因子值。该算法步骤如算法 2 所示。

算法 2　LOF-MR 算法

```
1. map(key,value)
2.    emit(value.IP/value.time,value.traffic,value.interactionIP)
3.    reduce(key,values[v1;v2;…])
4.    for val in values[v1;v2;…]
5.       lof=LOF(val.traffic,val.interactionIP) //根据流量和交互 IP
         数量计算每个时间点的局部偏离因子
6.       emit(key,val.time/lof)
```

4　实验与效果评价

网络日志分析引擎由 6 台普通 PC 组成(1 台 MainCtrlNode、4 台 WorkerNode、1 台日志服务器),PC 硬件配置为:Intel i5-6500 四核 CPU、8GB RAM。安装软件为:CentOS 5.5,Hadoop 1.0.2。

实验 1　分别在单机和 Hadoop 集群环境下运行源 IP 统计算法,输入的日志文件大小为:2 G,4 G,8 G,10 G,20 G,40 G。记录各次处理的运行时间情况,其结果如图 2 所示。可以看出,随着日志文件的逐步增大,并行日志分析算法的耗时并未很快上升,其快速且高效的特点逐渐明显,说明它在处理大数据量文件时,要比传统的单机方式具有更好的性能优势。

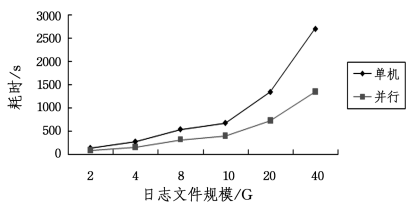

图 2　源 IP 统计算法在单机和并行模式下的运行时间比较

实验 2　LOF-MR 算法的等效度量实验。等效度量指标(ISO-efficiency)评估增大问题规模对并行算法的性能影响,如式(5)所示:

$$E=\frac{1}{1+\frac{T_0}{T_1}} \tag{5}$$

其中,T_1 是只启动 1 个 WorkerNode 时 LOF-MR 算法的运行时间,T_0 是系统并行处理所引起的额外开销,主要包括节点空转和各节点之间的通信、同步、调度等时间代价。

启动不同个数的 WorkerNode(2 个、3 个或 4 个 WorkerNode)来对不同规模的日志文件(2.6G,4G,5.5G)运行 LOF-MR 算法。实验结果如图 3 所示。WorkerNode 的个数和等效度量指标成反比;日志文件规模和等效度量指标成正比。2 个 WorkerNode 处理 2.6G 日志文件、3 个 WorkerNode 处理 4G 日志文件、4 个 WorkerNode 处理 5.5G 日志文件时,等效度量指标 E 都保持在 0.76 左右。数据显示:随着 WorkerNode 个数、日志文件规模同时增加,指标 E 基本保持不变。这说明当日志分析引擎处理的日志文件规模变大时,可以使用扩充节点的方法来弥补性能的耗损,即 LOF-MR 算法表现出了良好的可扩展性。

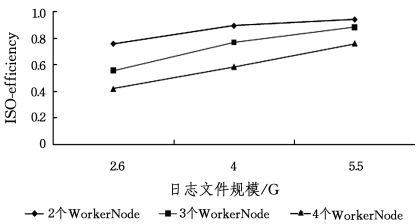

图 3　LOF-MR 算法的等效度量曲线图

结束语　在信息安全领域,网络日志的分析与挖掘技术有着广泛的应用。但在当前大数据环境下,传统算法暴露出了诸多问题。本文依托大数据技术,提出基于 Hadoop 的日志分析引擎架构,在 MapReduce 模型下重新实现了 IP 统计算法和异常检测算法。实验证明,大数据平台能有效解决数据处理中数据量大的问题,并具有良好的性价比和可伸缩性。

参 考 文 献

[1]　国光明,洪晓光. 基于日志挖掘的计算机取证系统的分析与设计[J]. 计算机科学,2007,34(12):299-303.

[2]　WINDING R,WRIGHT T,CHAPPLE M. System Anomaly Detection: Mining Firewall Logs[C]// Securecomm and Workshops,2006. IEEE,2006:1-5.

[3]　SANDFORD P J,PARISH D J,SANDFORD J M. Detecting security threats in the network core using data mining techniques[C]// 10th IEEE/IFIP Network Operations and Management Symposium,2006(NOMS 2006). IEEE,2006:1-4.

[4]　李学龙,龚海刚. 大数据系统综述[J]. 中国科学:信息科学,2015,45(1):1-44.

[5]　SHVACHKO K,KUANG H,RADIA S,et al. The hadoop distributed file system[C]// 2010 IEEE 26th symposium on Mass storage systems and technologies (MSST). IEEE,2010:1-10.

[6]　孟小峰,慈祥. 大数据管理:概念、技术与挑战[J]. 计算机研究与发展,2013,50(1):146-169.

[7]　DEAN J,GHEMAWAT S. MapReduce:simplified data processing on large clusters[J]. Communications of the ACM,2008,51(1):107-113.

[8]　HAN J W,KAMBER M,PEI J. 数据挖掘:概念与技术(3 版)[M]. 北京:机械工业出版社,2012.