doi:10.3969/j.issn.1000-436x.2014.z1.004

基于 SQL-on-Hadoop 的网络日志分析

章思宇[1]，姜开达[1]，韦建文[1]，罗萱[1]，王海洋[2]
（1. 上海交通大学 网络信息中心，上海 200240；2. 上海交通大学 电子信息与电气工程学院，上海 200240）

摘 要：当今网络带宽、设备和应用数量急剧扩张，日志管理面临数据量爆炸式增长的挑战。基于 SQL-on-Hadoop 构建网络日志分析平台，实现千亿级日志存储和高效、灵活查询。利用真实 TB 级数据集对多种 Hadoop 列存储格式及压缩算法进行性能测试，并对比 Hive 和 Impala 引擎日志扫描及统计查询效率，选用 Gzip 压缩的 Parquet 格式可将日志体积压缩 80%，且将 Impala 查询性能提升至 5 倍。基于该平台已开发 6 种安全事件响应、攻击检测和预警应用并发挥良好效果。

关键词：日志分析；大数据；Hadoop；SQL；网络安全
中图分类号：TP393.08 **文献标识码**：A **文章编号**：1000-436X(2014)Z1-0014-06

Network log analysis with SQL-on-Hadoop

ZHANG Si-yu[1], JIANG Kai-da[1], WEI Jian-wen[1], LUO Xuan[1], WANG Hai-yang[2]
(1. Network and Information Center, Shanghai Jiaotong University, Shanghai 200240, China;
2. School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: With the rapid expansion of network bandwidth, devices and applications, log management is facing the challenge of exploding data volumes. Log analysis platform built on SQL-on-Hadoop is capable of storing and querying hundreds of billions of log entries effectively. Columnar and compressed data formats for Hadoop are benchmarked with real-world multi-TB dataset. Conditional and statistical querying efficiency of Hive and Impala is tested. With gzipped parquet format, log data can be compressed by 80%, and querying with impala is 5 times faster. On this platform, six security incident analysis and detection applications are already deployed.

Key words: log analysis; big data; Hadoop; SQL; network security

1 引言

网络日志在网络运行管理中发挥着重要的作用，尤其在安全领域，日志是安全事件追溯、取证分析的重要依据，并且为入侵检测和漏洞挖掘提供了支撑。目前已较成熟的日志集中管理系统解决了各类设备、服务器和应用日志的采集和格式统一问题，日志分析也从最初简单的正则匹配，向结构化查询、报表和预测演进[1]。

当今网络带宽迅速扩容使日志量爆炸式增长。上海交通大学网络信息中心采集的 IP 流、DNS 和 HTTP 日志，每日合计有近 12 亿条，体积超过 500 GB。仅上述 3 类日志存储一年就将产生近 200 TB 的 4 000 亿条日志，若接入更多设备和操作系统日志，数据体量则更大。如何存储和使用如此规模的日志数据，成为我们面临的重大挑战。

Wei 等人[2]设计的 Analysis Farm 摒弃了传统的关系型数据库（RDBMS），利用 NoSQL 数据库 MongoDB 构建了可横向扩展的日志分析平台，以支撑 NetFlow 日志存储和查询。Rabkin 等人[3]设计了基于 Hadoop 的日志收集和分析系统 Chukwa，日志处理程序在 MapReduce 框架上开发。

Hadoop 为大数据存储和分析提供了理想的平台，但是 MapReduce 程序较长的开发周期无法适应灵活变化的日志分析需求。本文在 Hadoop 基础上

收稿日期：2014-10-18
基金项目：国家自然科学基金资助项目（61371084）
Foundation Item: The National Natural Science Foundation of China (61371084)

研究基于 SQL-on-Hadoop 引擎构建网络日志分析平台，能够使用应用广泛的 SQL 语言快速、灵活查询。本文利用 TB 级日志数据对存储、查询性能进行测试和优化，构建的平台解决了数百 TB 容量、千亿级日志管理的难题，为众多网络安全大数据分析应用提供支撑。

2　Hadoop 结构化数据处理

设备和服务器软件生成的原始日志是一种半结构化数据，日志管理系统的采集器对不同格式的日志进行标准化处理，从而以结构化的形式进行日志存储和分析。

2.1　HDFS 数据采集

网络日志的生成是分布式的，与任何日志管理系统一样，日志采集是本文平台的基础。本文平台采集的日志直接存储在 Hadoop 文件系统（HDFS）中，基于 Hadoop 众多衍生项目，分析实际需求，实现了 3 种日志采集方式。

1) 文件导入：对已分布在各服务器磁盘的日志文件，经网络文件系统挂载，直接将日志文件导入 HDFS。该方式允许日志文件批量可靠导入，可在网络利用率低谷时段进行传送。

2) 流数据导入：基于 Apache Flume[4]构建，实现多个日志源数据实时汇聚，接收网络设备、服务器发送的 Syslog 日志。

3) RDBMS 导入：为实现与现有日志系统兼容，基于 Apache Sqoop[5]实现与 MySQL、PostgreSQL 等 RDBMS 对接，直接导入已存储在这些数据库中的记录。Sqoop 同时可将 SQL-on-Hadoop 处理结果输出到 RDBMS，供现有的日志分析系统进行报表和可视化处理。

2.2　SQL-on-Hadoop 引擎

SQL 是查询、处理结构化数据最常用的语言，强大的需求催生出多种在 Hadoop 上运行 SQL 的解决方案。

Apache Hive 是最早的一个 SQL-on-Hadoop 方案，最初由 Facebook 开发[6]。Hive 提供一种类似 SQL 的语言——HiveQL 进行数据查询，它将 HiveQL 查询转化为 MapReduce（MR）任务执行。Hive 表结构则由 Hive 元存储（Metastore）维护，存储在传统数据库中。Hive 基于 MR 的设计导致一个明显的缺点，即 MR 任务的初始化耗时较长。

Impala[7]是 Cloudera 开发的一个开源的 MPP（massively parallel processing）SQL 引擎，它在 Hadoop 所有计算节点上运行守护进程，SQL 查询对 HDFS 的访问直接由守护进程操作本地磁盘文件。由于没有 MR 开销，以及磁盘 I/O、查询语句编译等一系列优化，Impala 的查询性能通常要数倍优于 Hive [8]。Impala 共享 Hive 元存储，可直接与 Hive 管理的数据互操作。

本文只针对上述 2 个 SQL-on- Hadoop 引擎进行研究，因为 Hive 和 Impala 是开源、可获得的，当前使用最为广泛的引擎。

2.3　结构化数据存储与压缩

目前，很多研究提出在 Hadoop 中优化结构化数据存储的方法。Floratou 等人[9]提出的 RCFile 格式旨在提高数据导入和处理效率，它首先将数据水平分割为多个行组（row-group），然后对每个行组内的数据垂直分割作列存储。列存储将数据表同一列的数据连续存放，当查询只涉及部分列时，可大幅减少所需读取的数据量。ORC（optimized RC）是对 RCFile 的改进，解决其在数据类型和性能上的多个局限性，改善查询和空间利用效率。

Parquet[10]是另一种为 Hadoop 设计的列存储格式，最初由 Twitter 开发。Parquet 的设计支持复杂的嵌套数据结构，文件组织同样按照行组和列存储两级划分，它是 Impala 推荐的存储格式。

表 1 比较了 Hive 和 Impala（1.4.0 版本）对上述文件格式的支持。Text 是原始的文本数据，通常为 CSV 或其他特定字符分隔。Hive 的格式支持更为全面，由于 Impala 和 Hive 共享元存储，因此本文平台实际应用中通常由 Hive 导入数据而后使用 Impala 查询。

表 1　　Hive 与 Impala 文件格式支持

文件格式	Hive		Impala	
	查询	插入/导入	查询	插入/导入
Text	√	√	√	√
RCFile	√	√	√	
ORC	√	√		
Parquet	√	√	√	√

数据压缩是另一种性能优化方法。压缩一方面节省存储空间，另一方面在相同磁盘 I/O 速度下可读写更多记录。Hive 和 Impala 均支持直接查询压缩的数据文件，常用压缩算法有 Gzip/Zlib 和侧重于

解压速度的 Snappy。ORC 格式本身已内嵌轻量级的压缩机制。

3 系统部署

基于 Hadoop 的网络日志分析平台在上海交通大学网络信息中心的部署使用 28 台服务器，其中 4 台作为管理节点运行 HDFS NameNode、Hive 元存储、Impala 目录及状态存储等服务。24 个计算节点，每个具有两路八核 Intel Xeon E5-2670 处理器、128 GB 内存和 12 个 2 TB 硬盘，并配置 240 GB SSD 用于 MapReduce 加速，所有节点通过 10 Gbit 以太网互联。Hadoop 部署采用 Cloudera 的发行版，版本为 CDH 5.0.2，HDFS 总容量超过 450 TB。

实验阶段，接入日志分析平台的数据为采集自校园网和数据中心出口的 IP 流和 HTTP 日志，以及在 DNS 服务器采集的 DNS 日志。IP 流日志以 Syslog 形式输出，每天记录数约 4 亿，体积 350 GB。HTTP 日志每天约 5.5 亿条，体积 165 GB；DNS 日志尺寸相对较小，每天 2 亿条记录消耗 15 GB 空间。

4 性能评估

为了评估本文基于 SQL-on-Hadoop 的网络日志分析平台的性能，并比较列存储格式和压缩对查询性能的优化效果，使用在网络出口采集的 404 亿条 HTTP 日志作为测试数据，原始日志尺寸 9.35 TB。真实日志上的测试结果能准确反映系统实际运行中的处理效率。

4.1 存储与日志扫描效率

对原始文本格式的 404 亿条 HTTP 日志，转储为 Parquet、RCFile 和 ORC 格式，并且启用 Snappy、Gzip 或 Zlib 压缩。表 2 首先比较了格式转换和压缩后的数据尺寸。采用 Parquet 紧凑的列存储格式并 Gzip 压缩，相对原始文本日志节省了超过 80% 的空间，而 Zlib 压缩的 ORC 文件体积仅为原始日志的 9.3%。

数据压缩后，Hive 和 Impala 查询时需对数据进行在线解压，消耗一定额外的 CPU 资源。因此，测试 Hive 和 Impala 执行全表扫描的性能，全表扫描是日志检索中最基本的操作。对于 Hive 查询，ORC 格式具有明显优势，最快 6 min 完成 404 亿条日志的扫描。Impala 对各种格式数据的扫描，性能均明显优于 Hive，尤其是 Gzip 压缩的 Parquet 格式，仅 2 min 15 s 即完成 404 亿条日志扫描，比 Hive 的最快时间缩短 2/3，同时也仅为 Impala 扫描原始 Text 日志耗时的 1/5。

表 2 存储格式和压缩算法比较

格式/压缩	体积/TB	Hive 查询/s	Impala 查询/s
Text	9.35	1 146.1	662.6
Parquet	6.78	1 106.2	489.7
Parquet / Snappy	3.10	713.9	231.2
Parquet / Gzip	1.73	726.2	134.6
RCFile	9.21	1 044.7	672.3
RCFile / Snappy	3.58	461.2	302.4
ORC	2.42	505.8	—
ORC / Snappy	1.15	357.9	—
ORC / Zlib	0.87	359.4	—

为了分析 Hive 和 Impala 性能差异的原因，监视查询过程集群资源使用情况（如表 3 所示），包括计算节点平均 CPU 使用率、总磁盘 I/O 峰值。通过优化 HDFS 块尺寸，最小化了网络开销，网络带宽对任一查询都不是瓶颈。

表 3 查询硬件资源使用率比较

格式/压缩	Hive		Impala	
	CPU	磁盘读取 /(GB·s^{-1})	CPU	磁盘读取 /(GB·s^{-1})
Text	99.9%	9.8	19.5%	16.9
Parquet	99.9%	8.4	26.0%	17.1
Parquet / Snappy	99.5%	5.5	56.3%	17.3
Parquet / Gzip	98.7%	3.0	99.9%	14.3
RCFile	99.2%	10.6	22.0%	16.8
RCFile / Snappy	99.2%	9.8	44.3%	17.2
ORC	99.3%	5.9	—	—
ORC / Snappy	98.7%	4.0	—	—
ORC / Zlib	98.8%	3.3	—	—

通过表 3 的结果可见，Hive 基于 MR 的查询是 CPU-bound 的，对未压缩文本扫描也耗尽了所有 CPU 资源，启用压缩后的额外 CPU 开销成为负担，处理 Gzip/Zlib 压缩数据的速度仅为未压缩时的 1/3。基于 C++的 Impala 执行效率要高得多，查询未压缩时仅 20%左右的 CPU 使用率，使其有足够的空闲 CPU 用于在线解压。测试中仅 Gzip 算法使 CPU 使用率达到 100%，其余查询 Impala 均能充分利用硬盘的读取能力，总的磁盘读取超过 17 GB/s，

查询耗时基本与文件尺寸成正比。

通过空间使用和查询性能比较，Parquet 格式结合 Gzip 压缩是本系统日志存储的最佳方案。Hive 和 Impala 均支持该格式的查询和导入，压缩后数据尺寸仅为原始日志的 20%，使平台所能存储的日志量提高至 5 倍，同时 Impala 查询性能也提升至原来的 5 倍，优势显而易见。

4.2　列存储效率分析

如 2.3 节分析，列存储格式在 SQL 查询只涉及部分字段时具有优势。本文设计对特定列扫描的查询语句，并由 Impala 在 Gzip 压缩的 Parquet 表上进行测试。

查询语句具体为 SELECT $column_1$, $column_2$, …, $column_n$ FROM *table* WHERE $column_1$ = *condition*，SQL 引擎将对语句中涉及的 *n* 列数据进行扫描。图 1 显示了查询涉及列数从 1 增加到 12 时，Impala 读取数据量及耗时的变化。日志中各列数据尺寸并不相同，例如仅扫描域名字段时，Impala 只需读取 64.8 GB 数据，花费 13.42 s，比扫描整个 1.73 TB 的表快了近 10 倍。

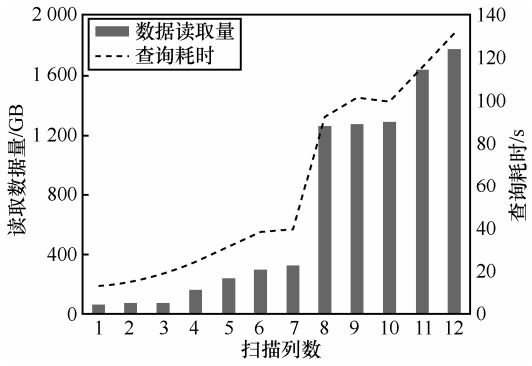

图 1　列存储查询测试

4.3　统计查询效率

4.1 节和 4.2 节对 SQL-on-Hadoop 引擎进行日志匹配筛选的性能进行了分析，实际的日志处理中另一大需求是统计报表，常用 SQL 分组统计查询实现。相对于之前测试的日志扫描，统计查询对计算性能要求更高。本文基于 HTTP 日志设计了 3 个常用的统计任务。

查询 1 统计访问学校主页域名的客户端数量，使用 COUNT DISTINCT 计算。

查询 2 统计各 HTTP 方法（如 GET、POST）的请求比例，使用 GROUP BY 分组统计。

查询 3 统计 Top 1000 的 User-Agent，同样使用 GROUP BY，与查询 2 的区别在于 HTTP 方法只有有限的几个，而 404 亿条 HTTP 日志中 User-Agent 有 1 700 万种，GROUP BY 产生的分组数巨大，且查询需对结果排序。

测试中 Impala 使用 Gzip 压缩的 Parquet 格式，而 Hive 使用 Zlib 压缩的 ORC，分别为两者处理最快的格式。测试结果如图 2 所示，前 2 个查询 Impala 优势明显，耗时分别为 Hive 的 16.7%和 43.9%。查询 3 的执行 Impala 花费了比 Hive 更长的时间，主要原因在于 Impala 对 JOIN 和聚合的实现仍为单线程，千万级 User-Agent 分组聚合凸显了 Impala 在此类计算中的瓶颈。

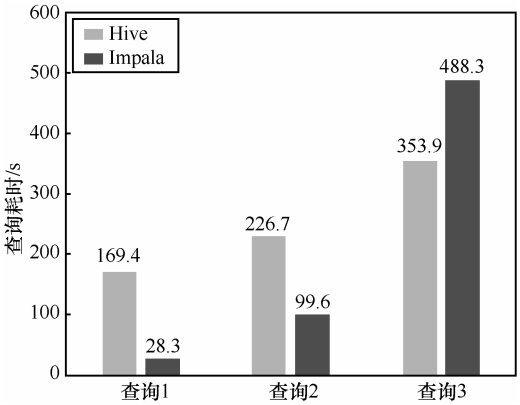

图 2　统计查询性能测试

5　实际应用

在本文实现的 SQL-on-Hadoop 日志分析平台上，已存储近 50 TB 的流量日志，并开发和移植了多种网络安全应用。

5.1　安全事件响应与追踪

根据长期存储的历史日志，可对已经发生的网络安全事件进行追溯、取证分析和影响评估。而对于最新公布的安全漏洞，本平台提供了以下 2 种关键的安全保障能力。

1) 攻击代码分析与追踪

以 2014 年 7 月 2 日公开的 Discuz!论坛程序 SQL 注入漏洞为例，基于漏洞信息从 HTTP 日志获取漏洞利用代码，提取特征片段后可持续对利用代码的传播和变形进行跟踪。

图 3 显示了本平台监测到的攻击者数量，与过去许多影响面较广的漏洞发展态势相似，漏洞公开后攻击代码和工具被迅速制作传播，首日就有近百个攻击者尝试入侵，公开后的 48 h 是攻击最集中的

阶段。日志反映漏洞公开前已有少量知晓者进行探测和利用，这也为基于日志发掘 0day 漏洞创造了条件。

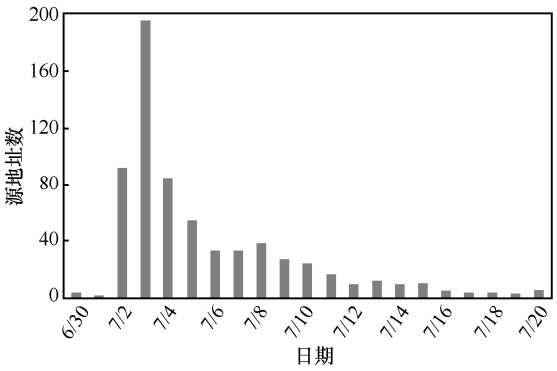

图 3 漏洞攻击者数量统计

2) 漏洞快速响应

漏洞公开后如此迅速地被传播和利用，对网络管理者的响应能力提出了很高的要求。对于类似的 Web 漏洞，如何快速定位校园网内存在风险的网站？

基于网站备案，日常管理成本很高且通常数据不完整、更新不及时；利用搜索引擎存在一定的盲区，无法保证内网全部站点都被索引；内网地址段全面扫描耗时长，且遗漏非常端口和路径上的网站。HTTP 日志检索是快速定位漏洞站点的最佳途径，利用本文平台查询入校流量日志，2 min 内即得到了受影响的网站列表，且该列表涵盖了搜索引擎和扫描器所能检测的范围，因为爬虫和扫描器的请求都被流量日志记录。

5.2 攻击检测与预警

日志分析更有价值的应用在于发现尚未知晓的攻击行为，或更进一步地发掘未知漏洞。通过大数据平台对多种日志和攻击模式关联分析，可实现复杂 APT 攻击检测。基于本文平台，以下 4 个检测和预警应用得以实现。

1) Web 攻击检测

在 SQL-on-Hadoop 平台上，将攻击特征匹配代码转化为 SQL，复杂的正则表达式匹配通过平台分布式处理，充分利用所有 CPU 并行计算。表 4 统计了该方法应用到数据中心服务器上半年流量日志检测到的攻击数。目前检测规则分 5 类，设计既保证很高的准确率（precision），又能较为通用地涵盖这类攻击已知和未知的利用代码。

表 4 Web 攻击检测结果

类型	源地址	请求数
SQL 注入	1 034	611 136
文件包含/下载	276	42 237
WebShell 尝试	297	197 183
路径探测	14 933	1 023 730
扫描器	799	1 037 806

2) 日志驱动的漏洞挖掘

日志驱动的漏洞挖掘分 2 类。基于攻击日志驱动，对 Web 攻击检测，尤其是 SQL 注入、文件下载和 WebShell 事件的 URL，主动发起探测，验证相关页面是否存在，并调用 sqlmap 等自动化工具加以验证。日志中包含大量扫描和探测，主动验证可剔除不成功的攻击企图，精确定位真实存在漏洞的页面。

另一类挖掘基于站点特性驱动，在 Web 日志中提取具有上传、下载功能及数值和字符串注入的页面，主动进行相关漏洞有用尝试。该方法优势在于能够挖掘尚未被发现和利用的漏洞，消除潜在安全隐患。

3) 数据中心出站连接审计

对于数据中心服务器而言，对外发起连接的数量比入站连接少得多，通常是软件安装和更新、外部 API 调用、代理服务器等产生。通过审计出站连接，已成功发现多起服务器入侵事件。表 5 显示了

表 5 被入侵服务器出站请求日志

时间	网址	User-Agent	内容尺寸
3/8 11:15	61.160.221.*:88/fuck	Wget/1.10.2(RedHatmodified)	6 772
3/8 11:16	61.160.221.*:88/xz32	Wget/1.10.2(RedHatmodified)	1 351 181
3/9 2:54	61.160.221.*:88/fuck3.sh	Wget/1.10.2(RedHatmodified)	404
3/9 2:54	61.160.221.*:88/fuck	Wget/1.10.2(RedHatmodified)	6 772
3/9 2:55	61.160.221.*:88/sshd8	Wget/1.10.2(RedHatmodified)	1 513 570
3/9 2:57	61.160.221.*:88/ssh.py	Wget/1.10.2(RedHatmodified)	1 324
3/9 2:57	61.160.221.*:88/mafix.tar.gz	Wget/1.10.2(RedHatmodified)	446 713

一个检测案例，入侵者在服务器上通过 Wget 下载多个后门和 Rootkit 程序。出站连接审计结合黑白名单、IP 和域名声望等参考数据，大幅降低人工分析的工作量。

4) Google Hacking 分析

搜索引擎关键词分析[11]对提升网站安全具有积极作用，移植到本平台后，可处理更长期的历史日志，并显著提高关键词提取和分析的效率。对 2014 年上半年入校 HTTP 请求 Referer 的分析显示，0.71%的访问来自搜索引擎结果，0.24%的搜索使用了高级语法。提取的 6 万多个关键词中，不乏对特定网站程序，以及对 Struts 框架 action 等后缀的搜索。文献[11]同时提出主动 Google Hacking 检测网站风险的方法。

6　结束语

本文研究了 SQL-on-Hadoop 技术在网络日志分析中的应用，利用 TB 级数据集对多种为 Hadoop 优化的存储格式及压缩性能进行测试，并对比了 Hive 和 Impala 引擎进行日志扫描和复杂统计查询的性能差异。应用 Parquet 列存储和 Gzip 压缩，日志存储占用的空间可压缩 80%，Impala 查询性能则提升至原来的 5 倍。实际部署后，本平台数百 TB 的容量可承载千亿级日志存储分析，为众多网络安全应用提供支撑。本文分析了 6 种安全事件响应、攻击检测和预警技术在平台上的部署使用效果。

参考文献：

[1] OLINER A, GANAPATHI A, XU W. Advances and challenges in log analysis[J]. Communications of the ACM, 2012, 55(2): 55-61.

[2] WEI J, ZHAO Y, JIANG K, *et al*. Analysis farm: a cloud-based scalable aggregation and query platform for network log analysis[A]. 2011 International Conference on Cloud and Service Computing[C]. Hong Kong, China, 2011.354-359.

[3] RABKIN A, KATZ R H. Chukwa: a system for reliable large-scale log collection[A]. 24th International Conference on Large Installation System Administration[C]. San Jose, CA, USA, 2010.163-177.

[4] The apache software foundation. Apache flume[EB/OL]. http://flume.apache.org/, 2014

[5] The apache software foundation. Apache sqoop[EB/OL]. http://sqoop.apache.org/, 2014.

[6] THUSOO A, SARMA J S, JAIN N, *et al*. Hive-a petabyte scale data warehouse using hadoop[A]. 26th IEEE International Conference on Data Engineering Long Beach[C]. CA, USA, 2010.996-1005.

[7] Cloudera. Impala[EB/OL]. http://impala.io/,2014.

[8] FLORATOU A, MINHAS U F, OZCAN F. SQL-on-Hadoop: full circle back to shared-nothing database architectures[J]. Proceedings of the VLDB Endowment, 2014, 7(12): 1295-1306.

[9] HE Y, LEE R, HUAI Y, *et al*. RCFile: a fast and space-efficient data placement structure in MapReduce-based warehouse systems[A]. 27th IEEE International Conference on Data Engineering[C]. Hannover, Germany, 2011.1199-1208.

[10] Parquet. Parquet[EB/OL]. http://parquet.io/,2014.

[11] 姜开达, 李霄, 孙强. 基于搜索引擎关键词的校园网安全分析[A]. 中国教育和科研计算机网 CERNET 第十九届学术年会[C]. 太原, 中国, 2012. 83-87.
JIANG K, LEE X, SUN Q. Campus network security analysis with search engine keywords[A]. 19th CERNET Annual Conference[C]. Taiyuan, China, 2012.83-87.

作者简介：

章思宇（1989-），男，上海人，上海交通大学助理工程师，主要研究方向为网络与信息安全。

姜开达（1980-），男，安徽池州人，上海交通大学工程师，主要研究方向为网络与信息安全。

韦建文（1986-），男，壮族，广西百色人，上海交通大学高性能计算中心计算专员，主要研究方向为高性能计算、高通量数据分析。

罗萱（1979-），男，江西丰城人，上海交通大学网络信息中心工程师，主要研究方向为云计算、网络虚拟化、软件定义网络和数据挖掘。

王海洋（1990-），男，黑龙江哈尔滨人，上海交通大学电子工程系博士生，主要研究方向为海量空间数据的分析与挖掘。