文章编号：2095-2163(2020)07-0034-06 中图分类号：TP393.08 文献标志码：A

BGP 安全事件快速检测框架的设计与实现

霍峻杰，张 宇，方滨兴
（哈尔滨工业大学 计算机科学与技术学院，哈尔滨 150001）

摘 要：边界网关协议(BGP)在路由交换、网络互连领域具有不可替代的重要意义，是世界网络互通的基础核心。BGP 安全事件具有影响力广、危害大等特点，需要及时发现 BGP 安全事件并进行排查修复。但是目前针对其设计的检测框架，数据处理慢，检测时延长，安全事件无法复现。基于以上需求，本文首先设计并提出了 BGP 安全事件快速检测框架，从结构、功能、运行和维护四个方面进行模块化设计，数据以管道流的形式流经各个模块；其次提出了基于分布式共享内存的 BGP 安全事件检测方案，并设计了可扩展的分布式历史数据库，用于对已发生的安全事件进行取证和复现；最后，对整个方案设计进行了初步实现部署，并对功能实现和性能指标进行了评估。

关键词：BGP 安全；分布式；模块化

Design and Implementation of BGP Security Incident Fast Detection Framework

HUO Junjie, ZHANG Yu, FANG Binxing
(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

【Abstract】 Border Gateway Protocol (BGP) plays an irreplaceable role in the field of routing exchange and network interconnection, and it is the basic core of world network interconnection.. BGP security incidents have the characteristics of wide influence and great harm, so it is necessary to find BGP security incidents in time and then conduct troubleshooting and repair them. However, the current detection framework is slow in data processing, extended in detection time, and unable to reproduce security events. Based on the above requirements, a fast BGP security incident detection framework is designed and proposed. This framework is designed to modularity from four aspects: structure, function, operation and maintenance. Data flows through each module in the form of pipeline flow. Secondly, a scheme of BGP security event detection based on distributed shared memory is proposed, and an extensible distributed history database is designed for forensics and recurrence of security events. Finally, the whole scheme design is preliminarily deployed, and the function realization and performance index are evaluated.

【Key words】 BGP Security; Distribution; Modularization

0 引 言

随着科技的发展和社会的进步，大到学术研究、国际合作，小到企业办公、个人出行，网络互联的意义变得尤为重要。在这个万物互联的时代，互联网络的安全性被推到了一个新的高度。随着国际互联网的发展与普及，越来越多的安全问题暴露在研究者的视野中。

BGP 是全球互联的核心基础，BGP 安全事件一旦发生，会快速造成全球范围的互联影响，甚至发生网络黑洞，网络风暴等损失巨大的安全问题。因此，亟需设计一套框架来提高 BGP 安全事件的检测、报警速度，同时还需拥有 BGP 安全事件还原复现的能力。本文的通过对数据源、数据量、输入格式、输出格式、检测方案、检测速度等多方面的思考比较，设计出一套针对 BGP 安全事件的，能够快速检测、运行稳定、可扩展性强，而且可复现分析的分布式流计算框架。

1 相关工作

BGP 安全攻击手段简单，成本不高，但造成的影响巨大，随着 AS 不断地增加，BGP 安全事件也越来越多。BGP 安全事件的发生有多元化的原因，如：攻击者恶意攻击，管理员错误配置，甚至商业利益驱动等等，都会引发 BGP 安全事件。目前，对 BGP 安全防护的研究主要分为两个方面。

制定 BGP 安全标准，完善 BGP 协议，使 BGP 在互相交换信息的同时，能保证保密性，完整性和可认证性等安全策略。最为著名的 BGP 路由信息安全解决方案是美国 BNN 公司在 2000 年提出的 S-BGP。S-BGP 使用公钥基础设施(PKI)，确认模块以及 IPSec 来共同建立 BGP 安全认证体系[1]。

基金项目：国家重点研发计划(2018YFB1800702,2016YFB0801303,2016QY01W0103)。

作者简介：霍峻杰(1996-)，男，硕士研究生，主要研究方向：BGP、网络拓扑测量；张 宇(1979-)，男，博士，副教授，主要研究方向：网络测量、网络安全、未来网络；方滨兴(1960-)，男，博士，院士，主要研究方向：网络与信息安全、网络分析、网络应急技术等。

收稿日期：2020-05-21

2003 年，Cisco 提出了安全起源 BGP 方案（SoBGP），旨在解决起源 AS 和 AS 路径的认证问题。SoBGP 采用类似 S-BGP 的 PKI 公钥链签名思想，但是使用的是网状的信任模型[2]。2005 年，Kerlin 等人提出了 Pg-BGP 方案，该方案的思路是对异常的 BGP 路由信息进行延迟使用和向下传播，这样可以缓解 BGP 攻击者的威胁程度[3]。

利用 BGP 相互通信的信息，对域间路由系统进行监测分析，从而监测响应 BGP 安全事件。国内有国防科技大学的 Rousseau 安全监测系统，国外有 BGPmon 安全响应系，Cyclops 监测系统，ARTEMIS 前缀劫持检测方案。这种方式不需要对现有的 BGP 系统进行大规模修改，易于部署，方便移植等优点，目前是业界较为流行的 BGP 安全检测防护方式。

第一种方案需要对已有的 BGP 系统进行重新部署，耗费的精力巨大，且对软硬件及网络资源需求很大，已经部署的 AS 管理者大多不希望使用这种 BGP 安全加强协议对 BGP 系统进行重新部署，所以这种方案在实际的 BGP 安全防护中并不被 AS 管理者所接受[4]。

域间路由系统的安全检测不需对 BGP 协议进行修改，不需 AS 管理者重新部署 BGP 系统，搭建成本低，可扩展性强，是目前主流的 BGP 安全防护解决方案[5]。

2　设计及实现

2.1　系统结构与模块设计

2.1.1　系统结构

BGP 安全事件快速检测框架基于分布式流处理的思想进行设计与实现。本文从设计原则，结构设计，功能设计和实现要点四个方面，结合结构、功能、运行和维护四个维度进行思考和设计，得到了 BGP 安全事件快速检测框架的基本系统结构。

（1）设计原则。结构上实现模块化设计，数据标准化处理，和资源合理分配；功能上实现快速检测 BGP 安全事件，且可对其进行复现，同时要保证框架在功能上的可扩展性，包括垂直扩展和水平扩展；运行上需要整个系统可靠稳定，能进行实时检测；维护上保证系统的鲁棒性，框架的可扩展性和数据的安全性。

（2）结构设计。结构上实现结构分层运行，检测分布式，以及资源调度的合理设计；功能上实现分布式流处理，多级平行扩展，和实时记录历史数据；运行上实现模块分离运行，互不干扰，分布式进行检测和数据的存储；维护上实现多机部署系统，简化从机的部署流程，保证数据的冗余度，从而提高数据恢复能力。

（3）功能设计。结构上实现不同功能分离，利于功能稳定性，数据流标准化预处理，使用主从模式工作和资源调度；功能上实现数据流任务分发，来分析检测，从机水平化部署，由主机统一管理调度，并维护一个历史数据库，持久化历史数据；运行上模块间使用数据管道相互通信，检测使用集群的方式对流数据进行检测，历史数据库均匀的分布在多机上，以实现负载均衡；维护上考虑容错容灾，应对突发情况，功能部署利于扩展，加入新需求或扩展性能等，提供数据恢复功能，在需要时保证数据的安全性。

（4）实现要点。结构上明确模块间的通信方式，使用数据流的管道模式，分析检测采用任务分发模式；功能上实现数据流转化为任务流，使用主从模式的部署管理方式，并且将历史数据以合理的方式持久化到数据库中；运行上实现检测任务容器化，不相互影响和干扰，历史数据存取时满足快速准确的需求；维护上使用投票模式，选举出 leader，在某一结点宕机时，可以重新选举，保证了系统的鲁棒性，从机部署要实现插件化，保证易于扩展，数据需要备份，创建多个副本，实现快速恢复。

2.1.2　模块设计

基于系统结构设计，并结合 BGP 安全事件的发生特性，本框架分为五大模块，分别是数据采集模块、流处理模块、分析模块、检测模块和入库模块。模块之间有着紧密的联系，构成一个完整的分布式检测框架。

数据采集模块负责定期下载 BGP 路由更新记录的 MRT 文件，将下载的文件路径传入流处理模块；流处理模块负责维护两个流队列：将已下载的文件路径传递给分析模块；将下载的 MRT 文件进行解析，解析成可读文本，传入流处理队列，构成源数据流，源数据流流入检测模块和入库模块；分析模块对 MRT 文件进行分布式计算，分析后的数据流入入库模块；检测模块使用分布式流处理，对源数据流进行处理和计算，得到的结果流入入库模块；入库模块流入的数据流有 3 种：分析模块的分析结果，写入分布式关系型数据库；检测模块的检测结果，写入时序数据库；流处理模块的源数据流，作为历史数据，实时写入历史数据库。

2.1.3　技术架构

BGP 安全事件快速检测框架的主要技术架构

分为 6 个部分:分布式处理、流处理、集群管理、数据库、数据压缩和可视化。每个部分需要安装部署相应的软件,如图 1 所示,为各个机器部署一系列技术支撑。

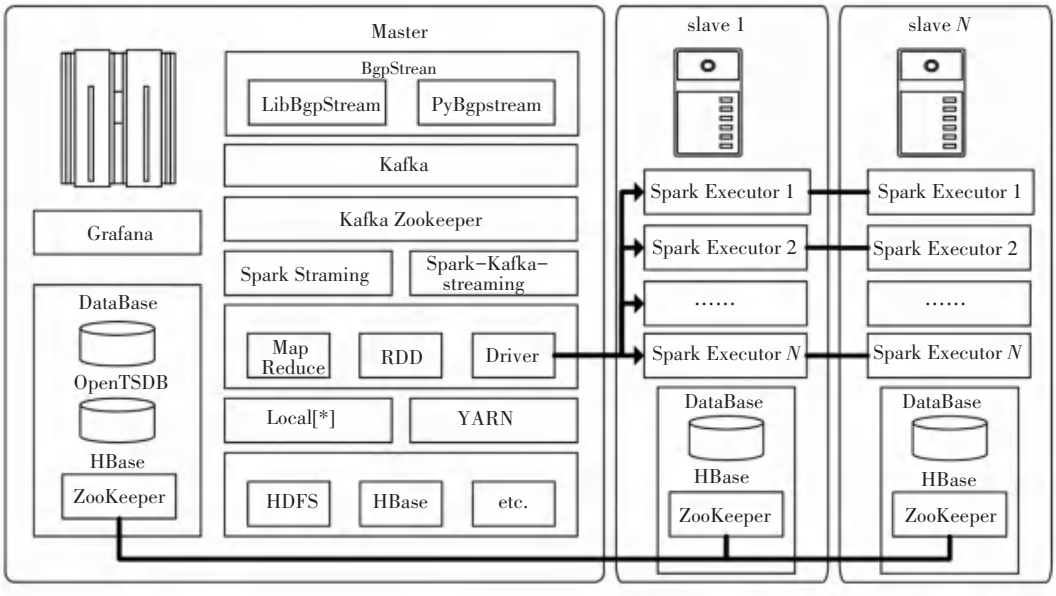

图 1 技术架构图

Fig. 1 Technical framework

(1)分布式处理:Hadoop, Spark, Spark Streaming

(2)流处理: Kafka, BGPStream

(3)集群管理:Zookeeper,Yarn

(4)数据库:Hbase, HDFS, OpenTSDB, Mysql, Mycat

(5)数据压缩:Snappy

(6)可视化:Grafana

2.2 BGP 前缀劫持检测

2.2.1 基准前缀树的构建

基准前缀树用来表示 BGP 网络中正常的前缀声明情况,是 BGP 前缀劫持检测的基础。基准前缀树基于 Trie 结构,Trie 树利用字符串的公共前缀减少查询时间,最大限度地减少无谓的字符串比较,查询效率十分高效,优于哈希树等查找树。Trie 树的核心思想是:使用空间换时间,利用字符串的公共前缀来降低查询时间的开销,达到提高查询的目的。由此可见,Trie 树结构在查找公共祖先方面有着得天独厚的优势,本框架中的基准前缀树采用 Trie 树进行设计,可达到快速获取前缀的公共祖先的目的。

本文使用 BGP 采集点的 RIB 数据作为构建基准前缀树的基础数据源。将源数据流入分布式共享内存,在内存中的数据处理分为 3 个阶段,如图 2 所示。

(1)阶段 1:根据设计的处理流程绘制构建 DAG 图;

(2)阶段 2:根据 DAG 图,对流入的数据进行前缀归并操作,获取前缀声明的 AS 列表;

(3)阶段 3:构建 Trie 结构的基准前缀树。

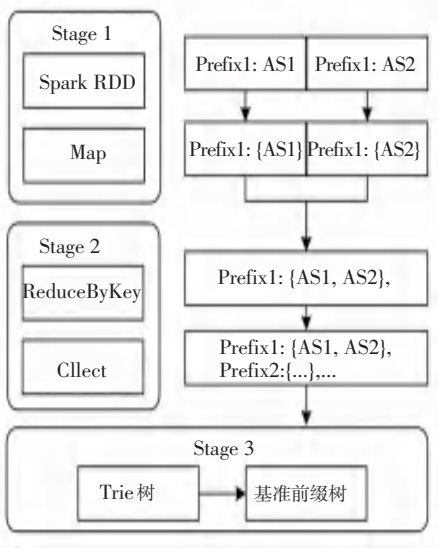

图 2 构建基准前缀树的三个阶段

Fig. 2 Three stages of constructing the benchmark prefix tree

使用基于分布式共享内存计算构建基准前缀树的优点很多。首先,在内存中构建 DAG 图,数据管道化流式进入,流经每一个 DAG 单元,不需要对中间结果进行保存和反复提取,减少了 IO 操作,显著提升计算速度;其次,数据流管道化可以避免同步等待,消除了不必要的进程等待时间;最后,构建的 DAG 图细化了每个操作计算单元,为分布式计算任

务分配和调度提供了设计支持。基于以上的优点，基准前缀树的生成速度将显著提升，构建效率呈指数增加。

2.2.2　BGP 前缀劫持检测方案

本文设计了一种基于基准前缀树的分布式流处理的快速检测 BGP 前缀劫持的方案，使用设计的检测策略，结合实时准确的基准前缀树，快速对发生的 BGP 前缀劫持事件进行检测和报警。

首先，从数据管道中的流数据（BGP 路由更新记录）提取 AS 及其声明的前缀；其次，从基准前缀树中提取此前缀及其母前缀；最后，将当前 AS 与基准前缀树中提取的 AS 列表进行对比，若存在其中，则为疑似前缀劫持事件。

基于基准前缀树 BGP 前缀劫持检测方案流程，如图 3 所示。

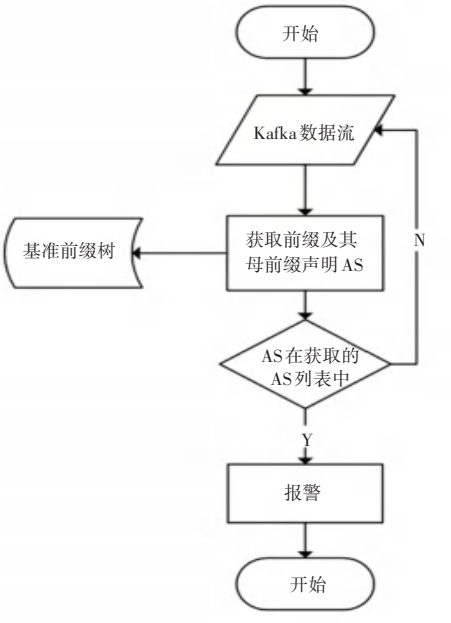

图 3　BGP 前缀劫持检测流程

Fig.3　BGP prefix hijacking detection process

2.3　历史数据库的存储设计

历史数据库存储历史 BGP 路由更新数据，为检测 BGP 安全事件提供辅助数据支持，同时可以针对特定的 BGP 安全事件定点复现。由于历史数据库的存储数据量较大，且对数据写入、数据查询、数据扩展以及数据鲁棒有着较高的需求，所以需要选择合理的数据存储方式，使用可靠的数据仓库，设计合理的存储结构，来满足 BGP 安全事件检测框架的历史数据库需求。

2.3.1　分布式列式存储

本框架使用的历史数据存储方式为分布式列式存储。分布式存储具有存储量大、负载均衡、可扩展性强、数据鲁棒性强的特点，是历史数据存储的合适选择。分布式列式存储使用以下技术来提高数据存取的效率，保证存储系统的稳定性：

（1）写入数据时，先将总体数据划分为几组数据，每一组数据单独在内存中建立 B+树后集中刷入到磁盘，减少了磁盘 IO，提高了资源利用率，数据写入速度大大提升。此外，刷入新数据时，采用了“追加写入形式”，这样的写入方式，在数据量激增时也不会产生写入性能瓶颈。

（2）数据组织使用 LSM 树形结构，定期对写入磁盘的 B+树进行合并操作，按照字典序对 RowKey 进行排序，在处理查询请求时，减少了磁盘寻道时间，大大提高了查询数据的速度。

2.3.2　存储格式设计

最重要的环节就是针对存储的信息格式，设计 HBase 数据库。HBase 中有一项叫做 RowKey，是 HBase 写入、查找等操作所依赖的索引列，可理解为 HBase 的主键。RowKey 设计的好坏，直接影响写入和查询的效率，需要根据写入数据的格式和查询的需求对 HBase 的 RowKey 进行合理的设计。

由于同一时间戳，可能会包含多条 BGP 路由更新记录，他们的 RowKey 不能相同（HBase 中 RowKey 不能重复，否则会覆盖有效的数据），所以设计 RowKey 在包含时间戳的基础上，还包含这条 BGP 路由更新记录的 MD5 值前 4 位，以区分同一时间戳内不同的 BGP 路由记录。

HBase 根据 RowKey 的字典序排列，而以时间戳开头的 RowKey 必然会导致写入数据的“热点问题”。即在数据写入时，大量的写入操作集中在某一块区域，导致写入操作集中在某一块磁盘的某一片区域，写入速度大幅下降，同时数据写入失衡，出现一块磁盘满转，其他磁盘空闲的状态。因此，本框架设计了分区模式来解决“热点问题”。首先在创建 HBase 数据表时进行预分区（00--FF）；然后在 RowKey 前面随机加上 00--FF 范围的字符串，这样数据的写入就随机分配到这 256 个分区中，成功解决了数据写入的热点问题。

2.3.3　数据压缩存储

为了最大化利用数据存储空间，需要对存储的数据进行压缩存储。难点在于数据被压缩后，数据库的读写性能不能受到影响。针对上述需求，提出了使用 Snappy 对数据热备份压缩的方案。

历史数据库的存储结构设计为列式存储，每一列的数据类型相同，存储的数据相似，彼此之间的相

关性更大,所以压缩效率很高,适合数据的压缩存储。热备份操作压缩和解压的速度很快,对数据库读写性能影响很小,是理想的压缩方式。

Snappy 是热备份压缩的一种可用方案,具有压缩解压速度快,压缩率高,占用 CPU 资源少,兼容性好,鲁棒性强等特点,被广泛应用于数据热备份领域,本文也是采用 Snappy 方案对数据进行热备份,提高存储空间的利用率。

3 实验评价

3.1 实验环境

(1)系统:Ubuntu 16.04

(2)部署情况:1 台 master,2~n 台 slave

(3)硬件:内存 32G,硬盘 2T

3.2 BGP 前缀劫持检测性能

3.2.1 构建基准前缀树

实验方法:使用 36 个不同测量点 0 点(2020.04.12)的 BGP RIB 原始二进制 MRT 文件进行基准前缀树的构建。实验通过对比的方式进行,分别使用基于多进程计算的原始方法和基于内存的分布式构建方法(3 台分布式集群)对 BGP RIB 数据进行处理构建,并记录所用时间。

实验结果及统计:根据实验结果反映,原始方法构建基准前缀树效率低、速度慢、资源利用率低;使用基于内存的分布式构建速度显著提升,且资源利用率很高,最大化利用了现有的计算资源。

通过测试可知,基于内存的分布式方案在速度方面,相较于原始方案构建速度提升 15 倍;CPU 利用率也提升到原来的 4 倍,充分利用了分布式的计算资源;同时,内存利用率也提升到原始方案的 9 倍,内存利用率越高,与其他慢速 IO 交换数据的频率就越少,计算速率和性能也随之提升,测试结果见表 1。综上,使用基于内存的分布式方案构建基准前缀树,可以在充分利用计算资源的情况下,使构建的性能和速度得到显著的提升。

表 1 基准前缀树构建测试结果

Tab. 1 Benchmark prefix tree construction test results

测试项 \ 构建方式	基于多进程的原始方案	基于内存的分布式方案
所用时间/s	6313	431
CPU 利用率/%	27.78	98.6
内存利用率/%	5.21	51.25

3.2.2 前缀劫持检测测试

实验方法:使用 linx 测量点在 2020.04.12 的 08:45 的 BGP 路由更新记录文件,基于之前构建的基准前缀树,进行 BGP 前缀劫持事件的检测,本实验针对 08:45:44 发生的疑似 BGP 前缀劫持事件(AS 7908 劫持 AS 7409)进行检测。使用对比试验的方式,对比了多进程检测的原始方案和分布式流式检测方案在检测速度,CPU 资源利用率,内存利用率 3 个方面的性能对比,测试结果见表 2。

表 2 前缀劫持检测测试结果

Tab. 2 Prefix hijacking detection test results

测试项 \ 检测方式	多进程检测的原始方案	分布式流式检测方案
所用时间/s	10.57	0.91
CPU 利用率/%	16.67	98.2
内存利用率/%	5.83	66.38

从测试结果来看,分布式流式检测方案检测速度为秒级,相较于之前的多进程检测方案,速度性能提升很多;机器资源利用率上,分布式检测方案充分利用了可用的计算资源提升计算速度,减少了计算资源的浪费,最大化的利用了 BGP 安全事件检测框架的硬件资源。

3.2.3 前缀劫持实时监控系统

实验方法:BGP 前缀劫持检测结果写入时序数据库,使用 Grafana 系统对时序数据库进行可视化监控图呈现。本实验检测 AS 7049 在 2020.04.12 的 8:45 到 9:00 的 BGP 前缀劫持情况,监测结果如图 4 所示,其中高于报警红线的时间点为疑似攻击事件发生的时间点。

从监测结果来看,AS7049 在这段时间内可能被 BGP 前缀劫持,劫持的 AS 为 AS7908,监控系统的报警规则设置为,当 10 min 内有一半以上时间点数值超过安全红线,则很可能是发生 BGP 前缀劫持事件,予以报警。

3.3 历史数据库性能

(1)列式存储分布式数据库存取性能及存储占用测试。

实验方法 1:使用 rrc00 在 2020.4.12 日 00 点到 01 点的 BGP 路由更新数据作为数据源,数据量 2.7 G。采用对比实验的方式,将数据分别写入原框架关系型数据库(Relational DataBase, RDB)和基于列式存储的分布式历史数据库(Distributed History DataBase, DHDB)中,分别记录写入数据需要的时间,使用条件查询语句分别对上述数据库进行查询性能测试。实验分两部分进行:第一部分实验使用空数据库;第二部分使用已存储大量数据(200 G)的数据库,通过对比得到实验结果,见表 3。

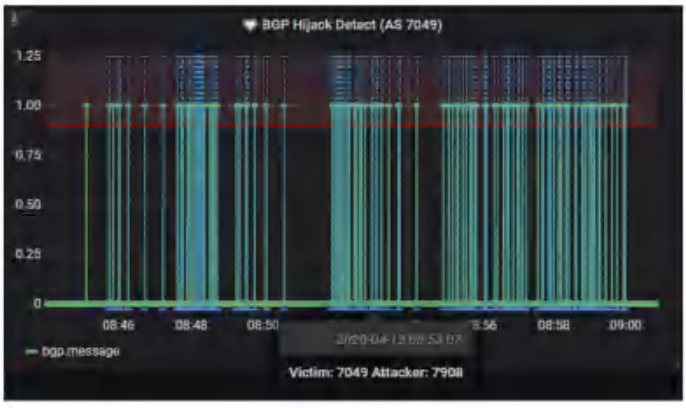

图 4　AS7049 前缀劫持监测结果

Fig. 4　The results of AS7049's prefix hijacking monitoring

表 3　历史数据库存取性能及存储占用测试结果

Tab. 3　Results of access performance and storage occupancy of historical database

测试项 \ 检测方式	空数据库		非空数据库	
	RDB	DHDB	RDB	DHDB
写入时间/s	3957.9	2674.7	4363.5	2718.1
查询时间/s	248.3	2.4	305.4	2.3
存储占用-正常/GB	12	9.3	12	9.3
存储占用-压缩/GB	12	1.8	12	1.8

测试在初始数据库为空的情况下，关系型数据库和分布式历史数据库的存取性能对比。可以看到，存入 1h 的数据，关系型数据库需要的时间超过 1 h，不能满足历史数据的存储性能要求，而分布式历史数据库则只需要 2/3 的时间，写入性能提升很大。在数据查询时，同样的查询请求，关系型数据库需要 4 min 左右，而分布式历史数据库则只需要 2.4 s，在查询性能上也是远远超过关系型数据库。除此之外，存储同样的数据，分布式历史数据库的存储空间占用也减少到 1/6，大大提升了存储资源利用率。

在数据库中已有 200G 数据的情况下，进行对比试验。可以看到关系型数据库的读写性能受到了影响，随着数据量的增大，甚至会出现性能瓶颈，而分布式历史数据库则不受影响，因为使用了列式存储，数据写入是追加形式，且存储有序，读写性能依然很高。

实验方法 2：使用 rrc00 在 2020.4.12 日 15 min 的数据作为测量数据。分别记录是否使用 Snappy 压缩算法对数据进行热备份，并记录存储空间占用情况。

由实验结果可见，热备份后存储空间占用减少到正常存储时的 1/4，大大提升了空间利用率，节约了系统资源。

(2)分布式历史数据库的扩展性能

实验方法：使用 rrc00 在 2020.4.12 日 15 min 的数据作为测量数据。分别记录集群部署 2 台机器和 4 台机器对分布式历史数据库的读写性能影响，测试结果见表 4。

表 4　历史数据库存取性能受集群数量影响测试结果

Tab. 4　Results of historical database access performance affected by the number of clusters

测试项 \ 检测方式	2 台机器	4 台机器
写入时间/s	165	135
查询时间/s	2.4	1.3

由实验结果可知，增加集群的规模可以有效地提升分布式历史数据库的读写性能，分布式历史数据库可扩展性很强。

4　结束语

本文设计并实现了一种基于分布式流处理的 BGP 安全事件快速检测框架；提出并设计了一个高性能、强扩展、事件可复现的历史数据库，并设计实验，与原始方案进行对比，在检测速度方面有着显著的性能提升；从性能、空间、集群规模等多个方面进行对比，得到了很好的实验结果。BGP 安全影响大、范围广、危害性极强，本框架具备快速检测，报警和展示 BGP 安全事件的能力，同时可以对已发生的事件进行复现和取证，为 BGP 的安全做了一份努力。

参考文献

[1] SEO K, LYNN C, KENT S. Public-key infrastructure for the secure border gateway protocol (S-BGP)[D]//Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01. IEEE, 2001, 1: 239-253.

[2] Ng James. Extensions to BGP to Support Secure Origin BGP(SoBGP)[S]. Draft-ng-sobgp-bgp-extensions-01.txt. November 2002.

[3] 王太红. 互联网前缀劫持检测与防御研究[D]. 北京：清华大学, 2015.

[4] Sanchez F, Duan Z. Region-based bgp announcement filtering for improved bgp security [C]//Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security. 2010: 89-100.

[5] 向阳. 互联网域间路由前缀劫持监测预防与研究[D]. 北京：清华大学，2013