Apache Storm 日志分析探索

Presented by:孟诚
唯品会信息安全部

Presentation将包含以下内容

- STORM简单介绍。
- 为何会开始研究STORM？
- 我们使用STORM计算什么？
- 搭建STORM平台过程中遇到的问题。
- STORM性能优化算法。

日志对于安全来说是什么?

我认为它是安全的感官

日志分析

非常有价值

- 几乎所有的异常都能从日志中发现

很难

- 海量的数据
- 准确发现

STORM简单介绍

- 分布式实时计算框架(**distributed realtime computation system**)

✓可扩展(scalable)

✓可容错(fault-tolerant)

✓确保每条数据都会被处理(guarantees your data will be processed)

我们使用STORM做什么？

Nginx日志

APACHE STORM™

Distributed • Resilient • Real-time

攻击监测

业务安全

攻击监测

- 大量正则规则匹配

  e.g script./script*

- 攻击自动化验证确认攻击有效性

业务安全

各接口监控
登录、注册、短信、购物车等

各活动监控
派券、抢购

各礼券使用监控
礼品卡、优惠券

STORM日志匹配规则热配置

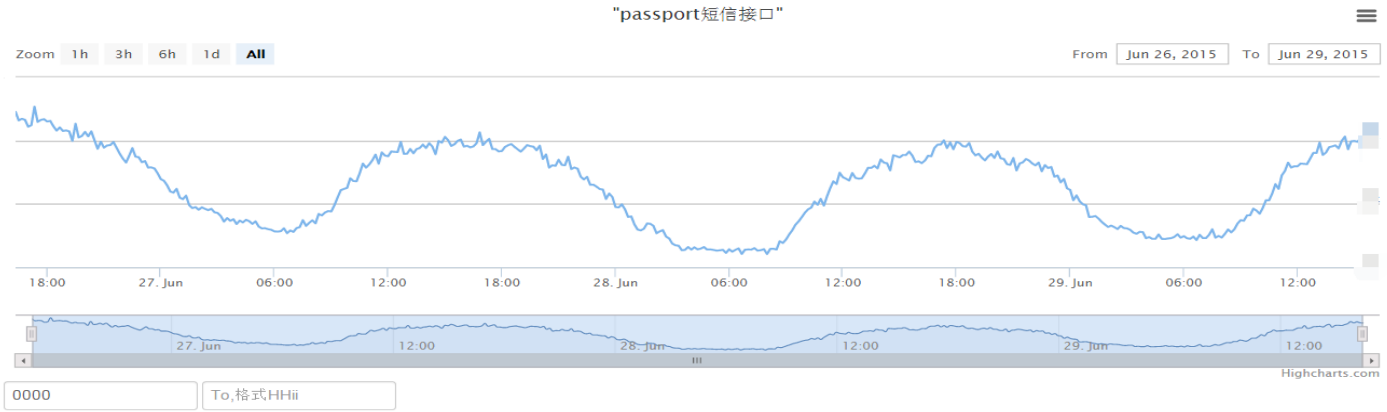
- 作用：无需更改任何代码实现日志匹配规则改变

STORM规则数据页面

Storm动态查询页面

首页 规则页面 概览页面 HIVE CDN节点查询

passport.vip.com
/register/ phone erify



0000 To,格式HHii

06-29 00:00 ~ 06-29 16:21

IP_ABC段	次数
120.4 3.0	52127
121. 49.0	30671
58.2 .0	30468
112.9 31.0	26597

IP段	标签	访问次数	日志
120.43.2	-	51159	日志
121.225.	-	30671	日志
58.246.5	-	30468	日志
119.184	-	21868	日志

STORM日志查询

passport短信接口

查询IP:120.43.[redacted],时间:1506290000 - 1506291621,总计51221个结果

回包大小分布： 72(4997) 0(2) 148(1)

HTTP状态码分布： 200(4998) 408(2)

暂时限制5000个结果

下载符合此规则的所有日志

时间	域名	请求	方式	回包	状态	remote_ip	forwarded_for	UA	机器	referer
1506290059	passport.vip.com	/regist[redacted]phone_verify	POST	72	200	[redacted]	120.43.[redacted]	Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.10 Safari/537.36	gd6-[redacted]	https://passport.vip.com/register/send_phone_verify
1506291433	passport.vip.com	/regis[redacted]hone_verify	POST	72	200	[redacted]	120.43.[redacted]	[redacted]ozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.10 Safari/537.36	gd6-[redacted]	https://passport.vip.com/register/send_phone_verify
1506290912	passport.vip.com	/regist[redacted]hone_verify	POST	72	200	[redacted]	120.43.[redacted]	Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.10 Safari/537.36	gd6-[redacted]	https://passport.vip.com/register?src=http%3A%2F%2Fwww.vip.com%2F%3Futm_source%3D2345%26utm_medium%3Dtextlink%26utm_term%3D20140522

无JAVA基础

前后端全写

对大数据
一无所知

挑战？

历时8个月

Sure, but we enjoy it!

不到20台服务器
（中等性能）

2位工程师

大数据流量

10,000,000,000

每日流量

18,000,000

峰值流量（每分钟）

2015.05.21 10am

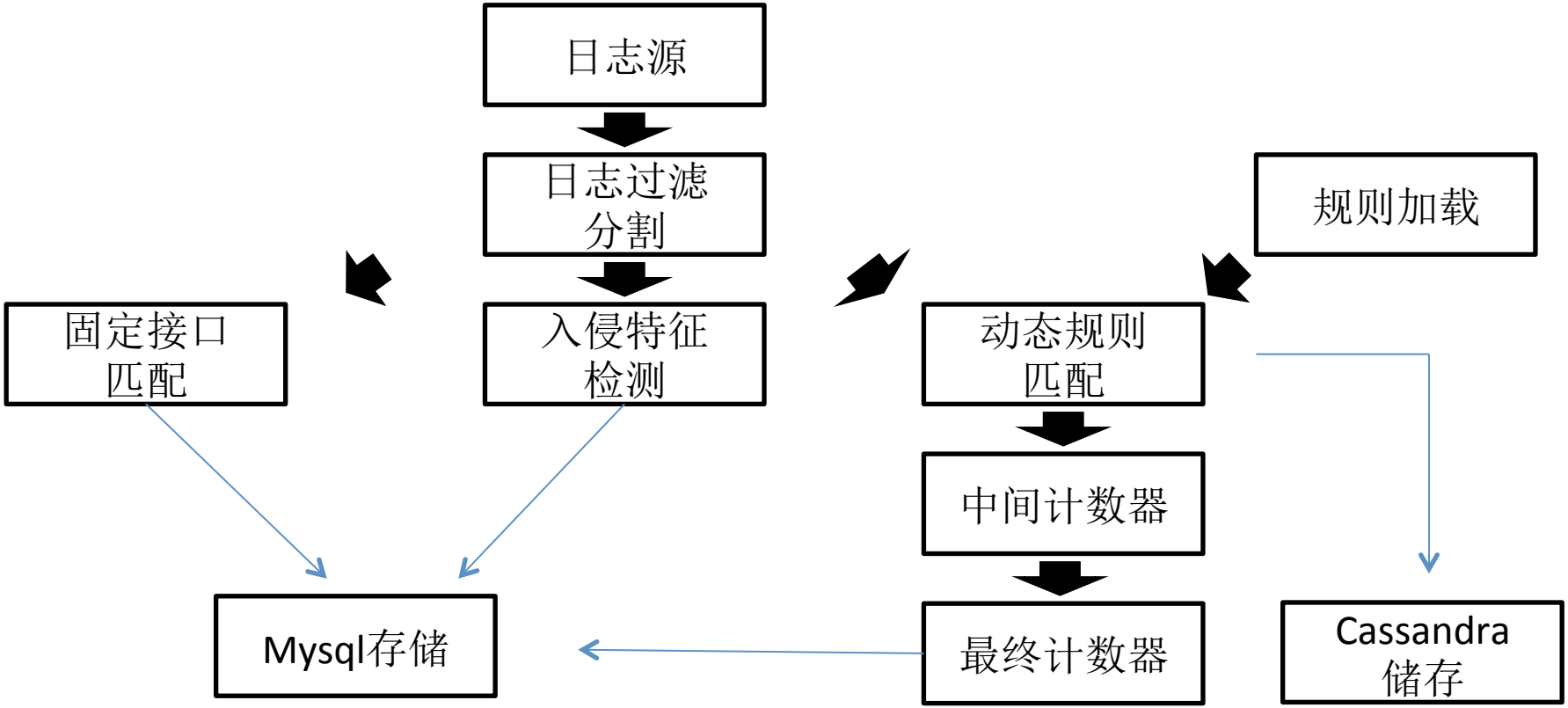
1,000,000

写入（每分钟）

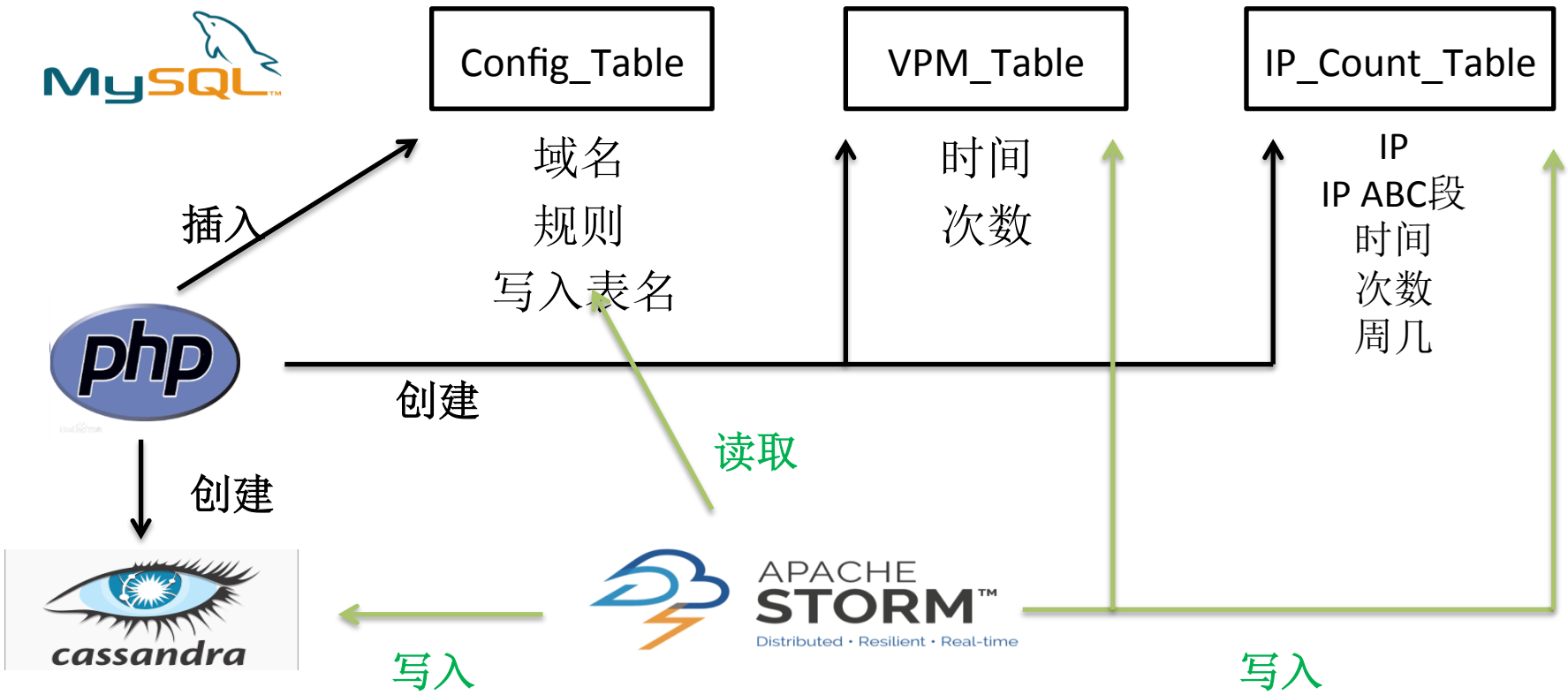
复杂的计算

	PV计算	UV计算	HTTP_Code	攻击匹配	动态匹配
特征匹配	无	无	无	**是**	**是**
数据结构	简单	中等	简单	简单	**复杂**
匹配规则确定性	确定	确定	确定	确定	**不确定**
数据写入量	低	低	低	**中**	**高**

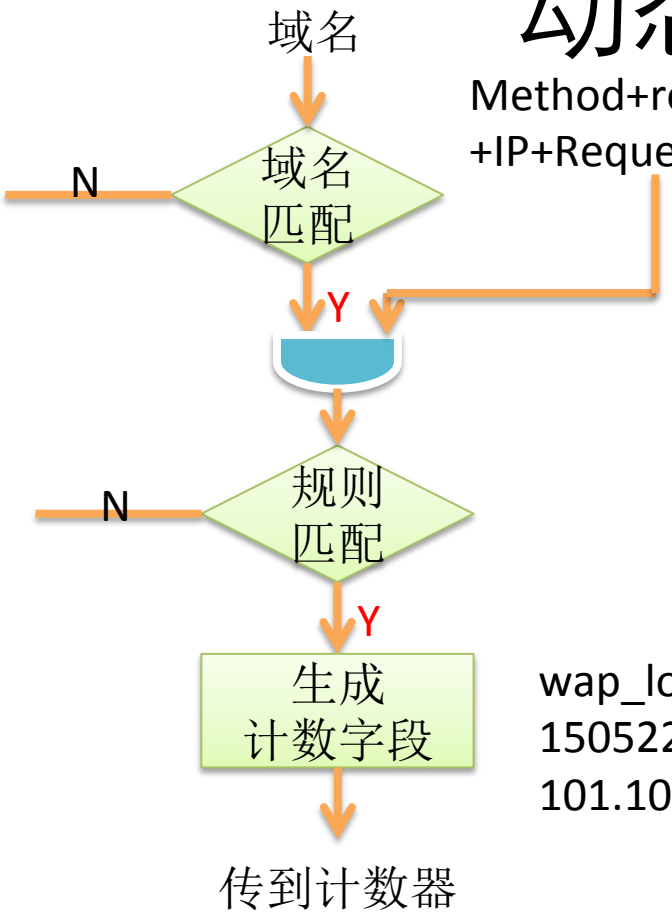
架构概述


日志源
日志过滤
分割
规则加载
固定接口
匹配
入侵特征
检测
动态规则
匹配
中间计数器
Mysql存储
最终计数器
Cassandra
储存

动态规则匹配


MySQL
Config_Table
VPM_Table
IP_Count_Table
域名
规则
写入表名
时间
次数
IP
IP ABC段
时间
次数
周几
插入
php
创建
读取
创建
cassandra
APACHE
STORM™
Distributed · Resilient · Real-time
写入
写入

动态规则匹配



Method+request+UA
+IP+Request_Data

wap_login,
1505221735,
101.102.135.1

域名	规则	表名	标记
www .vip.com	detail	W_1	T
www .vip.com	.*	W_2	F
cart. vip.com	cartadd	C_1	T
cart .vip.com	viewcart	C_2	F
mapi .vip.com	login_id=.*&passwd	Wap _login	T
sapi .vip.com	.*	S_1	T

预聚合

wap_login,1505221735,101.102.135.1
wap_login,1505221735,101.102.135.1
wap_login,1505221735,101.102.135.1

中间计数器

Key	value
wap_login, 1505221735, 101.102.135.1	1

中间计数器

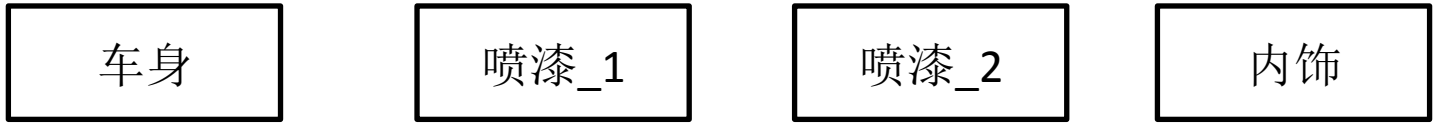
Key	value
wap_login, 1505221735, 101.102.135.1	2

最终计数器

Key	value
wap_login, 1505221735, 101.102.135.1	3

流水线处理

5分钟　　10分钟　　5分钟



5分钟　　5分钟　　5分钟　　5分钟

一小时能造几辆车？

1+(60-20)/10 = 5辆

1+(60-20)/5 = 9辆

流水线优化：正则

- SQL注入

Select.*from|from\\W+information_schema|/\\char\\(\\d*\\)|chr\\(\\d*\\)……

扫描器

sqlmap|Acunetix……

- 跨站脚本

script>.*</script……

- 敏感文件遍历

\\.\\./|\\.\\. %2f|etc/passwd|……

- WEBSHELL

phpshell\\.php|webshell\\.php……

Stucts

s|%20)+\\{.*\\|getRuntime|redirectAction……

流水线优化：避免使用正则

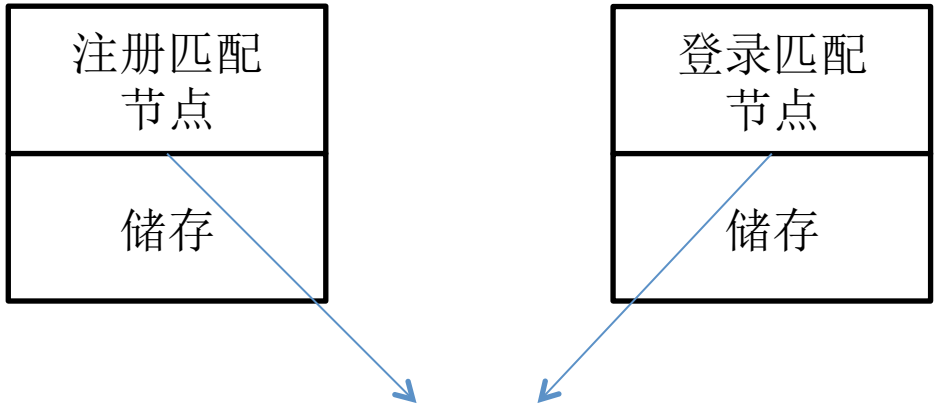
- 基于：多数访问流量是正常的
- 粗规则 与 细规则

String[] pre_sqlinj= {“select”,“information_schema”,”char”,”chr”};

sqlinj = Pattern.compile("Select.*from|from\\W+information_schema|/\
\char\\(\\d*\\)|chr\\(\\d*\\)",Pattern.CASE_INSENSITIVE);

流水线优化：统一化储存节点

- 处理节点更快处理下一数据
- 储存节点资源共享
- 代码可读性、维护性更强



Mysql优化：自动化分区

- 加快查询速度
- 瞬间删除大量数据

```
PARTITION BY RANGE (weekday)
(PARTITION Sun VALUES LESS THAN (1) ENGINE = InnoDB,
 PARTITION Mon VALUES LESS THAN (2) ENGINE = InnoDB,
 PARTITION Tue VALUES LESS THAN (3) ENGINE = InnoDB,
 PARTITION Wed VALUES LESS THAN (4) ENGINE = InnoDB,
 PARTITION Thu VALUES LESS THAN (5) ENGINE = InnoDB,
 PARTITION Fri VALUES LESS THAN (6) ENGINE = InnoDB,
 PARTITION Sat VALUES LESS THAN (7) ENGINE = InnoDB,
 PARTITION error_day VALUES LESS THAN MAXVALUE ENGINE = InnoDB)
```

MySQL优化：批量写入

- 批量化插入(Batch Insertion)

INSERT INTO yourtable VALUES (1,2);
INSERT INTO yourtable VALUES (5,5);
INSERT INTO yourtable VALUES (1,2),(5,5),...;

mysql_table_name	mysql_command	batch_no
Wap_login	Insert into Wap_login values('1','2'),('3','4');	2
www_table	Insert into www_table values('4','2'),('3','4'), ('3','8');	3
…	…	…
…	…	…
…	…	…

内存泄漏

MySQL连接

Sql=“insert into table (`a`,`b`) values (‘1’,’2’);”

Statement statement

=conn.createStatement();

statement.executeUpdate(sql);

statement.close();

statement = null;

数据结构

ArrayList<String> mysql_table_name =new ArrayList <String>();

ArrayList<String> mysql_command =new ArrayList <String>();

ArrayList<Integer> batch_no =new ArrayList <Integer>();

…

mysql_table_name.clear();

mysql_command.clear();

batch_no.clear();

Takeaways

- 日志是信息安全的传感器，很重要，但很难准确分析
- 平衡各个节点的处理时长非常重要。
- MySQL很脆弱，但是很好用。
- 大数据程序与小数据程序非常不同。
- 正则很强大，但也很累赘。
- 处理内存需要小心，依赖JAVA GC并不靠谱

问题？评论？建议？

请关注唯品会应急响应中心

消息系统

处理引擎

储存系统

检索引擎

自动验证流程