基于大数据技术的企业信息安全警报系统设计与实现

曾德真

南宁学院人工智能学院，南宁，530200

摘　要　使用 Prometheus 开源监控解决方案，结合 Grafana 可视化工具以及睿象云平台搭配自定义告警服务搭建一套基于大数据技术的企业信息安全警报系统。在 Prometheus 提供的能力基础之上，通过配置信息的同步更新从而完成 Prometheus 组件创建，采集任务生成，告警任务生成的能力；再借助 Prometheus 的 PromQL 数据查询功能集成 Ganfana 实现数据可视化展示；并通过 Ganfana 与睿象云平台的集成实现自定义告警通知。

关键字　Prometheus，Grafana，睿象云，可视化

Design and Implementation of Enterprise Information Security Alert System Based on Big Data Technology

Zeng Dezhen

School of Artificial Intelligence
Nanjing University
Nanning 530200, China;
984374899@qq.com

Abstract—An enterprise information security alert system based on big data technology is built by using Prometheus open source monitoring solution, Grafana visualization tool and Ruixiang cloud platform with custom alarm service.Based on the capabilities provided by Prometheus, the capabilities of Prometheus component creation, collection task generation and alarm task generation are completed through the synchronous update of configuration information, and the data visualization display is realized by integrating the PromQL data query function of Prometheu with Ganfana. Through the integration of Ganfana and Ruixiang cloud platform, custom alarm notification is achieved.

Key words—Prometheus, Grafana, Ruixiangyun, Visualization

1　引　言

企业系统架构的部署也日益复杂。信息化带来快速迭代的产品，业务的稳定可靠保证是企业存亡的关键，这时运维人员将面临新的挑战[1]。传统的监控方式与业务分离，无法直接监控业务内部情况等；传统的监控系统数据采集模式单一，不够灵活。线上系统任何故障和运维的任何失误都可能会为企业带来巨大的经济损失，这就需要一个高效稳定的监控警报系统，为企业运维人员提供行之有效的服务[2]。

监控系统在目前的环境下分为开源系统、商用系统、自研系统[3]。分别在互联网有一定的使用占比，受众群体不同，也都有相应的特点优势。商用系统一般是某些大型专业企业研发的商业版监控系统，对于部分小型互联网企业，自己没有能力去自行研发，开源系统又需要投入大量人力成本、运维成本，直接购买商用系统，省时又放心，可以全身心投入产品迭代。开源监控系统一般是针对有余力的企业结合开源组件搭建的监控系统，可减少开发量，快速提供监控能力，为后续定制化监控系统做铺垫。自研系统则是用于有能力去付出时间和人力成本进行研发，定制特定场景需求的企业，开源和商用版都不足以满足企业的监控需求，只能通过自行研发的方式满足特定需求[3]。

Ganglia 是加州大学伯克利分校发起的一个开源监控项目[4]，可用于测量数以千计的节点。Ganglia 的核心包含 Gmond、Gmetad 以及一个 Web 前端，主要是用来监控系统性能，如：CPU、MEM、硬盘利用率、I/O 负载、网络流量情况等，通过曲线很容易见到每个节点的工作状态，对合理调整、分配系统资源，提高系统整体性能起到重要作用[5]。但是 Ganglia 没有警告和消息通知机制。

CAT（Central Application Tracking）是一个提供实时监控警报，应用性能分析诊断的工具。它是基于 Java 开发的实时应用监控平台，所以侧重于对 Java 应用的监控。CAT 为美团点评提供了全面的实时监控告警服务，主要包括移动端监控、应用侧监控、核心网络层监控、系统层监控等。CAT 最大的优势是：它是一个

实时系统，大部分系统是分钟级统计，但是从数据生成到服务端处理结束是秒级别，整体报表的统计粒度是分钟级。第二个优势是：其监控数据是全量统计，客户端预计算，链路数据是采样计算。

Zabbix 是一个企业级的开源分布式监控解决方案[8]。其核心组件包含 Zabbix server 与 Zabbix agent。Zabbix server 作为整个服务的服务端，自身可以采集服务器的状态信息，也可以结合 Zabbix agent 组件，通过服务端主动拉取或者采集端主动上报的方式获取监控数据，最后通过 web 的形式提供各服务器的运行状态[9]。在 Zabbix 系统中所反映出的监控数据以及元数据之间的独立性，为并行、可扩展性提供了基础，可以说 Zabbix 监控系统非常适合基于数据分组的并行化和微服务化。但是面对数据量大时，展示直接读取数据库，会有卡顿。

上述的介绍的是目前在国内用得比较多的开源监控组件，各自有优缺点。目前在企业中，一般是根据自身实际需求以及自身技术实力进行技术选型。也有许多国内互联网企业也比较普遍地使用商用的监控系统，例如阿里云监控、腾讯云监控等。这些商用的监控系统可以为部分企业提供一个较为全面的监控告警能力，减少了企业的大量运维成本。

对于自主研发监控系统，一般是用于自己企业内部的资源监控，并且充分考虑企业业务所处场景，高适配的监控体系，定制化应对企业业务所面临的监控问题，可能会针对数据采集或者数据存储方面进行自行研发，从而使得运维更加方便，快捷。

本文设计实现的监控警报系统是基于 Prometheus 开源监控组件、Grafana 可视化工具以及睿象云平台，结合企业中的实际业务场景开发的，对于企业的各业务线都有不错的适配性。Prometheus 采用时序数据库，大大的节省其存储空间，并且提升其查询效率；同时本系统采用 Granfana 于睿象云的集成，可以实现实现数据可视化多样展示与警告通知立即响应等效果。本监控警报系统正在逐步完善其监控功能，为企业提供高效稳定的监控服务，并为后续企业向自研监控系统的搭建提供了一个基础。

2 系统设计

本文系统的设计思路是：在开源 Prometheus 的基础上，结合 Grafana 技术与睿象云平台的集成来实现企业安全信息监测。其系统的物理架构如图 1 所示，其系统架构主要包括数据采集、数据存储、监控告警与数据可视化四大模块。

2．1 数据采集模块

主要采用 Prometheus 扩展组件 Node-exporter 采集底层服务器的各种运行参数，如 CPU、内存使用率，主机运行时长，分区使用率，网络状态，数据 IO 吞吐量等信息。例如对于 Flink 任务，需要采集 Flink 的 JobManager 服务器状态、Checkpoint 情况、程序运行时长、Taskmanager 内存，流量等指标。

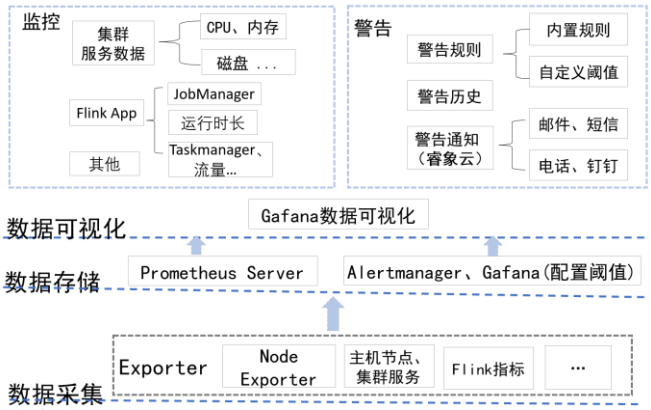

图 1 系统的功能设计

当 Prometheus 通过 Node-exporter 采集到相应的监控指标样本数据后，于是就基于采集到的数据特征维度，通过 PromQL 技术进行过滤、聚合和统计，从而产生新的计算后的一条时间序列的数据并将此数据进行存储操作。例如利用 PromQL 时间序列的标签匹配模式来对时间序列进行过滤，目前主要支持两种匹配模式有完全匹配和正则匹配。一般描述样本特征的标签(label)在并非唯一的情况下，可以通过 PromQL 查询数据，返回多条满足这些特征维度的时间序列。

2．2 数据存储模块

为数据的可视化呈现提供有效的数据支持，需要对采集到的数据进行存储，Prometheus 将采集到的监控数据按照时间序列的方式存储在本地磁盘当中。

2．3 监控告警模块

当监控指标出现异常情况时，Prometheus 首先推送单条异常告警，如果异常情况大于 1 条，会同时触发过滤规则，此时 Prometheus 会将所有的告警合并成一条 JSON 数据推送出来。根据这一特点，Prometheus 可以有效减少告警数量。最后，再通过睿象云平台以短信，邮件，钉钉或电话的形式通知运维管理员，如图 2 所示。

同时，警告警报还提供智能化分析建模预警，即通过成熟的人工智能算法对系统运行数据和用户行为数据进行建模后最终达到精准智能化预警的效果。

2．4 数据可视化模块

数据可视化模块采用 Grafana 可视化工具以及结合 Prom QL 语法，为用户提供面向集群级的数据展示服务，展示图形可定制化，兼容多种数据源，便于用户监控管理。

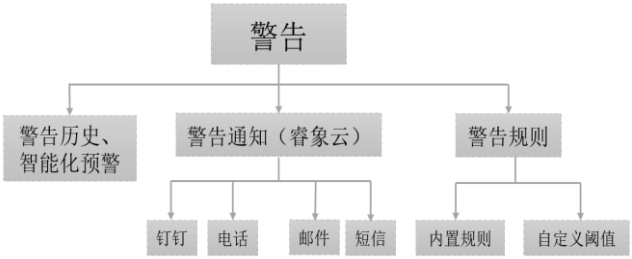

图 2 警告通知信息图

3 系统实现

针对技术方案确定系统物理架构图分步骤、按模块实现整个系统的功能。以下分别对数据采集、数据存储、监控告警与数据可视化这四个模块的实现过程进行详细介绍。

3．1 数据采集模块

该模块主要收集主机数据、系统数据、容器数据等，然后将收集到的数据进行规范化，并进行存储。具体的操作步骤如下：

Step1：根据企业实际业务及资源情况需求，搭建好 Hadoop、Flink 分布式集群，把集群节点、Linux 服务等作为监控目标。

Step2：在集群内安装 exporter，实现对集群性能数据的获取。如 CPU、内存、磁盘、网络、程序运行时长和流量等资源数据信息。

Step3：通过 exporter 采集不同维度的监控指标，并通过 Prometheus 支持的数据格式暴露出来，Prometheus 定期 pull 数据，并用 Grafana 可视化工具和睿象云平台进行展示与警告通知运维管理员。

Step4：采集 Flink 的 JobManager 服务器状态、Checkpoint 情况、程序运行时长、Taskmanager 内存，流量等指标数据，并通过暴露的 metrics 接口用 Prometheus 抓取。

Step5:通过 Prometheus-node-exporter 采集主机节点的性能指标数据，并通过暴露的 metrics 接口用 Prometheus 抓取。

Step6：当 Prometheus 通过 Node-exporter 采集到相应的监控指标样本数据后，于是就基于采集到的数据特征维度通过 PromQL 技术进行过滤、聚合和统计，从而产生新的计算后的一条时间序列的数据并将此数据进行存储操作。例如利用 PromQL 时间序列的标签匹配模式来对时间序列进行过滤，目前主要支持两种匹配模式有完全匹配和正则匹配。

一般描述样本特征的标签(label)在并非唯一的情况下，可以通过 PromQL 查询数据，返回多条满足这些特征维度的时间序列。可以用 PromQL 来对时间序列数据进行聚合操作处理，从而形成一条新的时间序列。对于范围数据分析处理，我们采用三种表达式进行处理：一是直接通过类似于 PromQL 表达式 httpreqeststotal 查询时间序列时，返回值中只会包含该时间序列中的最新的一个样本值，这样的返回结果称之为瞬时向量；二是对过去一段时间范围内的样本数据时，需要使用区间向量表达式；三是区间向量表达式和瞬时向量表达式之间的差异在于：在区间向量表达式中，需要定义时间选择的范围，时间范围通过时间范围选择器进行定义。

3．2 数据存储模块

该模块主要是通过部署时编写好的yaml文件内的告警规则语言，将数据收集层获取到的数据进行规格化和过滤处理，提取需要的数据到监控告警模块，Prometheus把收集到的数据通过exporter保存统一格式的数据存储到Prometheus自带的时序数据库，用于Grafana调用。

（1）Prometheus搭建安装

具体实现操作如下六个步骤：

Step1：把Prometheus镜像打包好并且放到集群镜像仓库中，用于后面Prometheus的安装。

Step2：在搭建好的Hadoop集群中创建名字为monitoring的命名空间，主要用于存放Prometheus运行的容器。

Step3：给monitoring分配集群的读取权限，用于Prometheus可以通过Kubernetes的API获取集群的资源相关信息。

Step4：在monitoring创建Config Map用来存储Prometheus容器的一些配置以及Kubernetes集群中动态发现pod和运行中的服务的配置。

Step5：创建Deployment模式的Prometheus，通过yaml文件安装Prometheus。

Step6：连接Prometheus，通过yaml文件把Prometheus内部端口映射成外部端口，用于Hadoop集群自动连接到Prometheus，即Prometheus部署成功。

（2）Prometheus工作流程

Prometheus工作流程如图3所示，主要分为四步：

Step1：Prometheus server 定期从配置好的 jobs 或者 exporters，或者从Pushgateway，或者从其他的 Prometheus server 中拉metrics。默认使用的拉取方式是pull，也可以使用Pushgateway提供的push方式获取各个监控节点的数据。

Step2：Prometheus server 在本地存储收集到的 metrics，并运行已定义好的 alert.rules，记录新的时间序列或者向Alertmanager推送警报。

Step3：Alertmanage根据配置文件，对接收到的警报进行处理，发出告警。

Step4：在图形界面中，可视化采集数据。

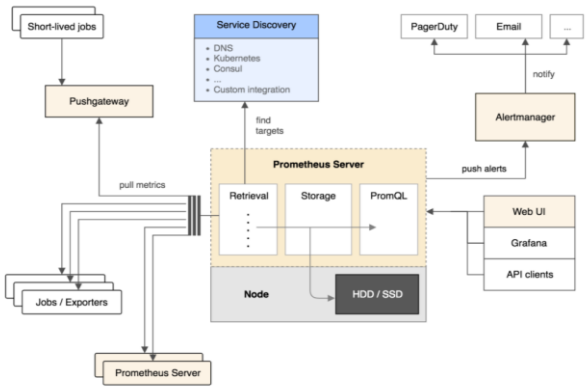

图 3 Prometheus工作流程图

3．3 监控警告模块

该模块主要分为主机、工作负载和群资源等监控，监控的内容具体有：主机监控，主机数量、内存（node_memory_MemTotal_bytes， node_memory_MemFree_bytes，node_memory_Buffers_bytes）、CPU使用、磁盘使用量、磁盘一个小时内的 I/O、一个小时内的网络 I/O；工作负载监控，CPU使用率；群资源监控，Flink集群的JobManager服务器状态（flink_jobmanager_job_uptime）、checkpoint情况、程序运行时长、Taskmanager内存，网络延时（flink_ jobmanager_job_uptime），重启次数（flink_jobmanager_job_numRestarts）、流量等指标。

告警规则配置层主要是根据第三层获取到的数据进行告警规则设置、告警阀值设置、告警联系人设置和告警方式设置等。该功能主要通过Grafana进行配置，具体为以下四个步骤操作：

Step1：连接登录Grafana；

Step2：打开设置面板，选择预警接收类型，睿象云配置的警告通知类型；

Step3：设置预警值的范围；

Step4：预警配置成功，当资源到达设置的预警范围，即能发送预警通知。

最后再通过Grafana与睿象云平台集成，将告警事件进行实时记录以及通知用户，睿象云平台上以web界面的形式展示给用户，如监控统计结果、告警故障结果等进行统一展示。

3．4 数据可视化模块

数据可视化模块主要提供的前端Web展示界面，并实现快速和灵活的客户端图形，将数据收集层获取到的数据进行统一展示，展示的方式可以是饼图、曲线图、柱状图等多种形式图形展示，通过将数据图形化，可以帮助运维人员了解一段时间内主机或网络的运行状态和运行趋势，并作为运维人员排查问题或解决问题的依据。同时在同一个图中可混合不同的数据源，还可以根据每个查询指定数据源，甚至适用于自定义数据源。

实现数据展示层主要通过Grafana可视化工具，其连接Grafana流程如图 4所示，具体操作步骤如下：

Step1：连接Grafana，通过yaml文件把Grafana内部端口映射成外部端口，用于集群自动连接到Grafana。

Step2：使用管理员账号登录Grafana，并且配置Prometheus的数据源。

Step3：编辑好需要图表类型的JSON文件，导入到Grafana，用于调用各个图表的样式，显示各个数据类型的图表。

Step4：连接Grafana，即可看到相关默认模式的监控数据，即Grafana部署成功。

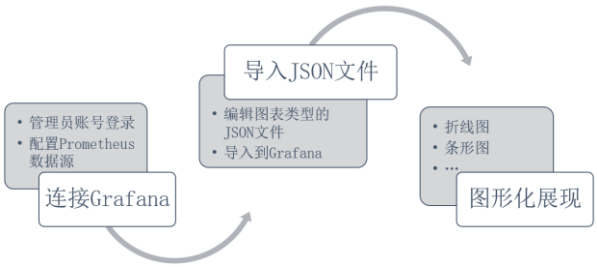

图 4 连接Grafana流程图

4 系统性能测试

针对技术方案确定系统物理架构图分步骤、按模块实现整个系统的功能。以下分别对数据采集、数据存储、监控告警与数据可视化这四个模块的实现过程进行详细介绍。

4．1 测试环境

测评主要使用了三台虚拟机搭建 Hadoop 分布式集群和 Flink 分布式集群。同时将 Prometheus、Alertmanager、Pushgateway 和 Grafana 部署在 hadoop01 上，Node-exporter 服务部署到 3 台 CentOS-7 虚拟机，Node-exporter 服务主要是采集 3 台主机的运行指标如 CPU、内存、磁盘、Flink 等信息。

4．2 监控警报模块测试

（1）Hadoop集群资源监控

Hadoop 集群资源监控总览页面，主要监控集群的运行时间、CPU、内存、磁盘、网络等资源数据信息。运维管理员可以对自己的主机列表进行监控，在页面中可以选择是否开启自动刷新，手动刷新，通过节点范围进行过滤等。监控的主要指标包括：节点状态、CPU 总核数、CPU 使用核数、内存总量、内存使用

量、磁盘总量、磁盘使用量等。查看具体的监控指标可以看到该节点的资源概览，主要指标包括：CPU 使用率、内存使用率、系统平均负载、磁盘使用量、磁盘 I/O、网络流入流出数据包数、网络流入流出数据速率等。图 5 到图 8 给出了资源监控详细页面中的各个指标详细信息。

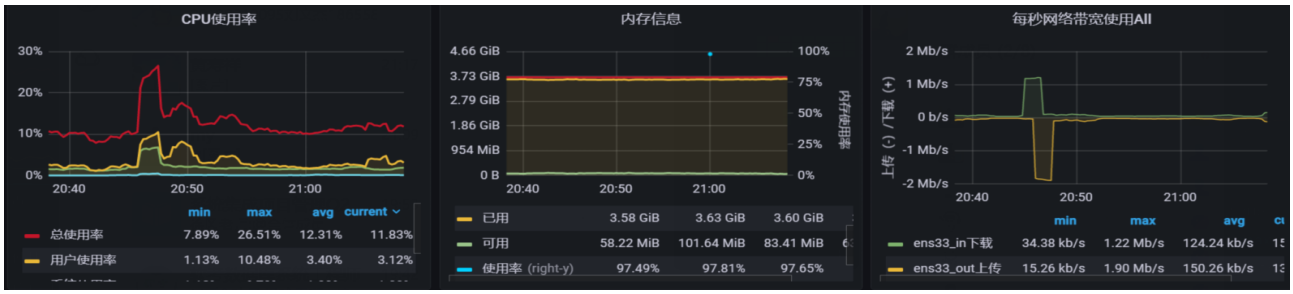

图 5 集群资源监控明细页面 1

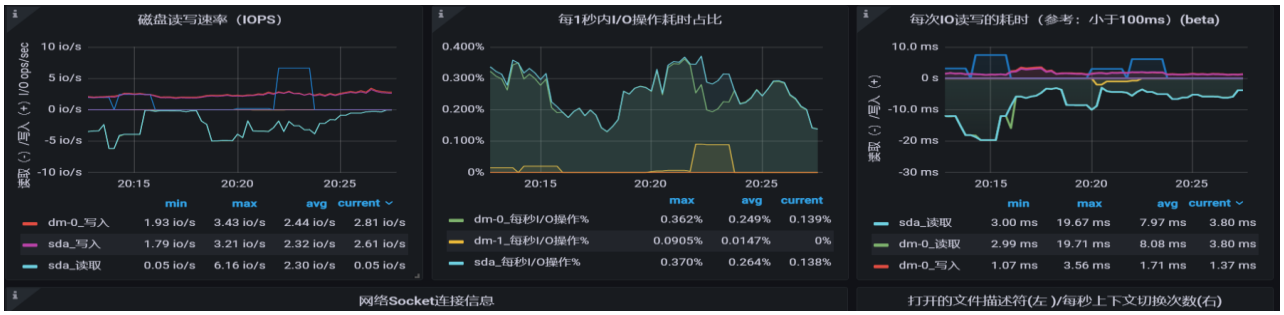

图 6 集群资源监控明细页面 2

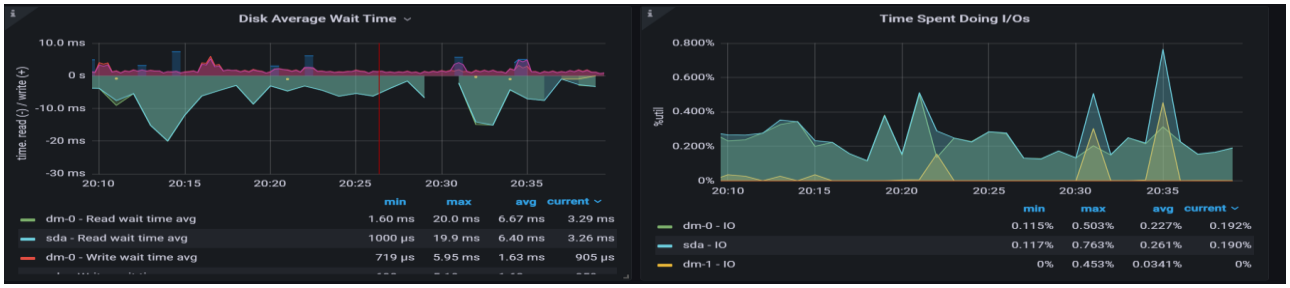

图 7 集群资源监控明细页面 3

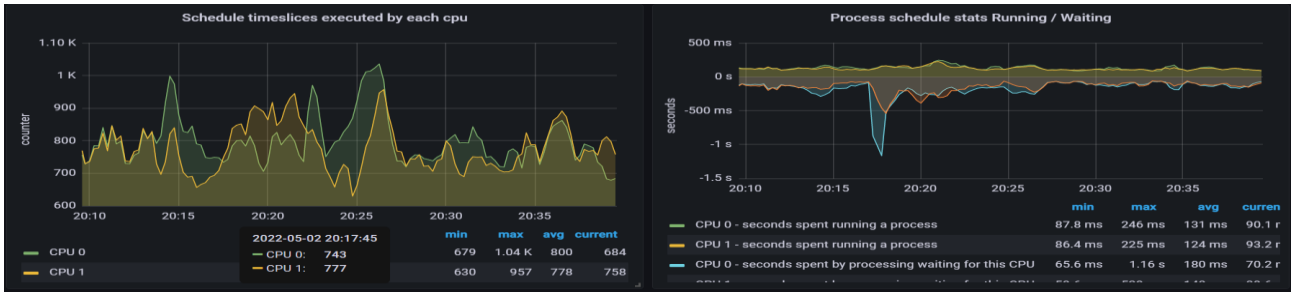

图 8 集群资源监控明细页面 4

（2）Flink 任务失败监控指标

而图 9 是监控 Flink Job 激活、Flink Job 网络延迟及任务重启指标的详细信息。Flink 任务失败监控指标主要是基于 flink_jobmanager_job_uptime 这个指标进行监控，具体如图 10 所示。从图 10 中得知数据上报到 Promgateway 频率为 30s，当 job 任务失败后数 Flink 上报的 Promgateway 的 flink_jobmanager_job_uptime 指标值不会变化。

其原理是在 Job 任务存活时，会按照配置 metrics.reporter.promgateway.interval 上报频率递增。基于这个特点，当任务失败后这个数值就不会改变，就能监控到任务失败，((flink_jobmanager_job_uptime)- (flink_jobmanager_job_uptime offset 30s))/100 值就会是 0。

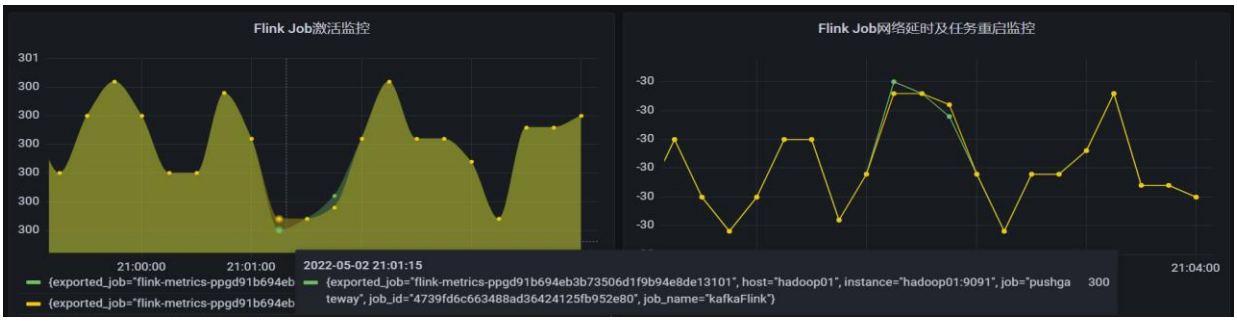

图 9 Flink 集群资源监控明细页面

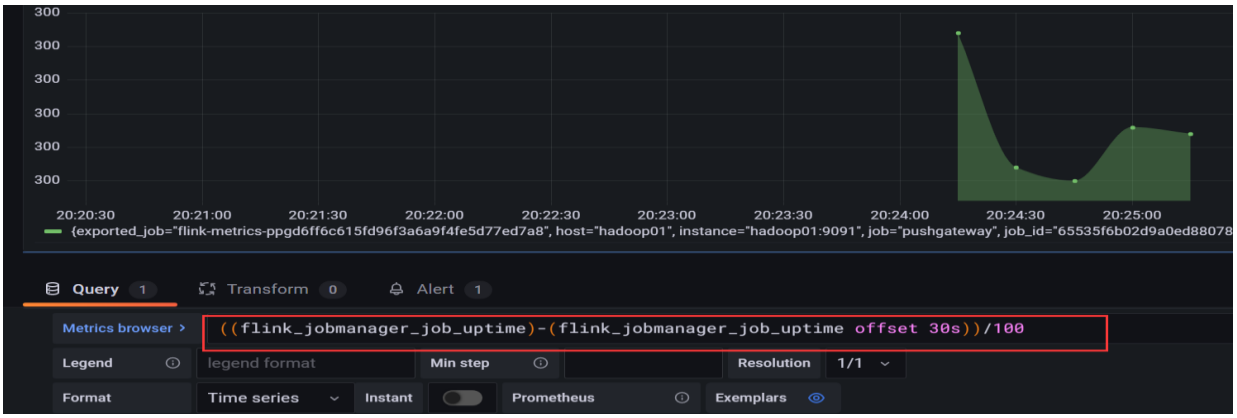

图 10 Flink 任务监控指标

（3）Flink网络延时或任务重启监控

Flink网络延时或任务重启告警是基于flink_jobmanager_job_uptime指标的监控，当出现网络延时或者重启后进行监控通知，监控指标((flink_jobmanager_job_uptime offset 30s)-(flink_jobmanager_job_ uptime))/1000。告警时机如下：

①延时导致值突然小于-30（正常情况为-30）时；

②重启会导致 flink_jobmanager_job_uptime指标清零从新从 0 值上报，导致查询公式值突然大于 0（正常情况为-30），

（4）Flink重启次数监控

Flink重启次数监控配置是基于flink_jobmanager_job_numRestarts指标的监控，表示flink job的重启次数。设置重启策略后，在任务异常重启后这个数值会递增一。可以单纯的监控重启次数，也可以每次重启都进行告警（差值）。利用当前值减去30秒前的值，如果其值等于1证明重启次数为一次。

4.3 集成第三方告警平台睿象云实现警告通知

针对告警通知出现接收不及时的问题，为确保通知信息能被及时接收，可通过配置Prometheus或者Grafana与第三方平台告警平台（例如睿象云）集成，进而通过第三方平台提供的多种告警媒介（例如电话，短信，邮件，钉钉）等发送告警信息。

本文针对通知出现接收不及时的问题，采用的方案是Grafana与第三方告警平台睿象云的集成，效果良好，警告通知能达到秒级别响应。具体操作如下：

（1）集成睿象云之前，须在其官网进行注册并登录，注册时需填入个人手机号和电子邮箱，其官方网站为： https://www.aiops.com。

（2）依次点击CA智能告警平台、集成，选择Grafana，填入应用名称，并点击“保存并获取应用key”。得到App Key之后，配置Grafana，在Grafana中创建Notification channel，配置channel。Test测试后会接到电话以及邮件，测试成功后需要保存其配置。

（3）Test、save测试成功后，需再睿象云中配置分派策略，分派策略可以配置，哪些应用的告警信息，发送给哪些用户。例如Flink_Job激活监控的告警信息发送给某个管理员。其中分派策略配置流程：依次点击“配置”，“分派策略”，“新建分派”。配置具体分派策略如图11所示，主要配置分派策略的名称、分派条件、分派人。分派条件与分派人可以是一个或者多个，当警告发生后，立即通知分派人。

（4）配置通知策略。通知策略通知策略主要配置告警状态、告警级别、通知方式、时间设置、延迟策略以及通知人等等。配置通知策略流程如下：一是依

次点击“配置”，“通知策略”，“新建通知”。同时通知策略可以配置一个或者多个，具体配置按实际应用场景进行相关配置；二是配置具体的通知策略，如图12所示。图中，如果提供一个多维度的通知方式，即可以通过电话、短信、邮件、微信、钉钉以及其它app等方式进行通知。当通知人时，可以配置一个或者多个。告警级别主要提供提醒、警告和严重三个级别；延迟策略可以设置为立刻、1分钟后、2分钟后、10分钟后，同时也可以设置为1个小时后、5小时后等设置选项。警报ton通知时间可设置为任何时间、工作时间和非工作时间三个选项，同时工作时间于非工作时间也是可以自己设置；警告状态可以选则发生时、认领时、关闭时或者是全部选择，本文警告状态配置为全部选择。

完成测试配置后，若停止Flink服务，随后即可触发睿象云平台的动作，进而根据我们配置的分派策略和通知策略，发送告警信息。告警信息以邮件、短信和电话的方式发送到注册时填入的邮箱地址或手机号码。效果如图13与图14所示。其中图13提供给运维管理员一个可视化的管理页面，管理员可以通过此页面了解到最近的警告事件、警告事件的级别以及警告事件分类等情况，以便更好的做出维护决策，甚至在一定程度上能达到预测警告事件的功能。

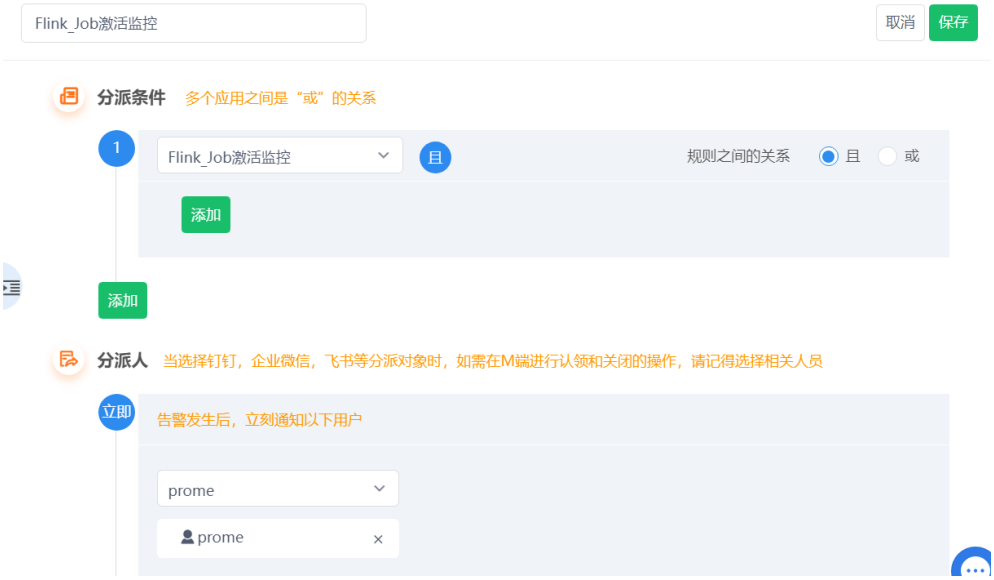

图 11 配置具体分派策略

图 12 配置具体的通知策略

图 13 警告通知面板展示

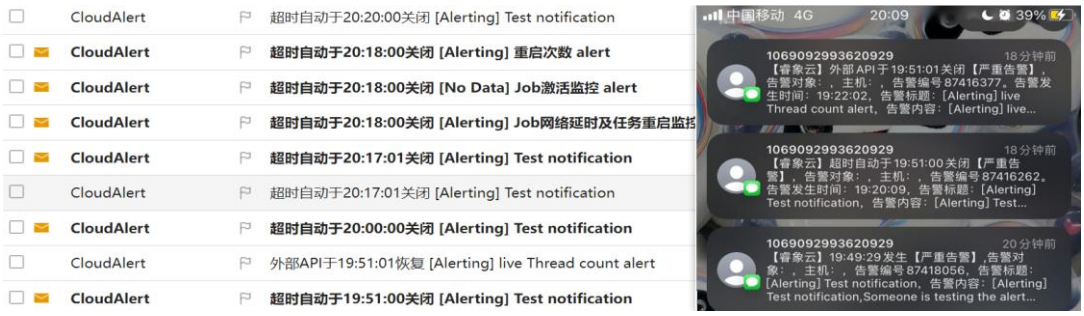

图 14 FlinkJob 宕机邮件与短信通知效果

5 结束语

本文介绍了互联网企业随着技术的高度实践带来的与日俱增的监控问题，对比分析了现有较为完善的监控系统以及互联网上优秀的开源监控组件。然后结合现企业中实际的运维状况，选择 Prometheus 为基础，结合 Grafana 可视化工具以及睿象云平台，搭建实现了一套稳定高效的企业级监控警报系统。目前，本系统由数据采集、数据存储、监控告警以及数据可视化四大模块组成。系统性能测试结果表明，系统设计实现达到预期的目标，实现了采集任务生成、告警任务生成、数据可视化展示和自定义告警通知等功能。本文的技术方案对中小型企业服务器架构部署一定的参考价值。

参 考 文 献

[1] MohaiminulIslam,ShamimReza.The Rise of Big Dataand Cloud Computing[J]. Internet of Things and Cloud Computing,2019,7(2).
[2] 郝鹏海,徐成龙,刘一田.基于 Kafka 和 Kubernetes 的云平台监控告警系统[J].计算机系统应用,2020,29(8):121-126.
[3] 段立明.面向大型分布式系统的智能监控系统设计与实现[D].沈阳：中国科学院沈阳计算技术研究所,2018.
[4] 张松.容器微云的监控系统的研究与实现[D].硕士学位论文,合肥：中国科学技术大学,2016.
[5] [澳]詹妈斯·特恩布尔(James Turnbu)著 史天 张媛 肖力译. Prometheus 监控实战, 2017.