

告警聚合

项目介绍

告警聚合基于新华三统一运维平台，是使用运维平台中的告警数据，对告警数据做聚合操作，生成聚合告警。解决痛点是运维平台中的海量告警数据，无法挖掘出用户真正关心，有价值的告警。

技术方案

- 产品定位：作为统一运维平台的一个组件，以zip包的形式发布，安装部署于容器化平台（该容器化平台基于k8s，并对k8s的api做了一些封装，向上层应用提供服务）
- 产品架构：使用了微服务架构，前后端分离，各自以pod提供服务进行解耦。告警聚合主要分为3个服务，分别是前端ui界面服务，后端api服务，和聚合流处理服务。
 - 前端采用spring boot+静态资源的方式构建部署微服务。其中静态页面中又通过引入vue对象和iview组件的形式进行相关开发。前端涉及到了中英文，权限校验和iview组件的使用。使用axios进行网络请求
 - 后端使用spring boot构建api的微服务。使用到了mysql，mybatis，swagger，logback日志记录。微服务之间使用feign client, rest api在线文档使用swagger，支持国际化。还使用了k8s api
 - 流处理采用flink 架构，使用flink-clients，flink-streaming 和flink-scala进行flink程序开发。
 - 使用一个Stream接收不断变化的数据（比如我们的配置数据），然后把这些数据广播到flink的所有task中，这样主Stream中的数据就能动态的从广播流中获取所需要的配置，然后根据动态的配置来处理数据。首先从mysql中读取规则流，使用broadcast算子构造广播流形式
 - 另外从kafka获取告警数据流，作为主流。两者使用connect算子，做双流join
 - 对join后的双流，在process算子内做规则过滤逻辑，如果过滤命中，则输出命中的告警
 - 如果在窗口内命中了则执行自定义sink操作，使用snmp4j 包发送trap，作为聚合告警输出。
 - 部署采用容器部署，前后端使用安装了jdk的基础镜像，flink使用安装了flink的基础镜像，都可以从官方hub下载。将对应的jar包拷贝，执行java命令或者flink run
 - 部署分为x86版本和arm版本
- 关键技术：
 - spring boot
 - vue
 - kafka
 - redis
 - flink

全景拓扑项目

项目介绍

最终目标为将全域资源拓扑的统一展示，作为拓扑框架主要提供导入资源和根据需求扩展业务菜单。如根目录以及视图菜单扩展，右键菜单，toolbar工具栏，tip扩展，节点样式数据扩展，拓扑试图内容数据扩展。

采用扩展的方式，便于各组件解耦，便利数据自定义展示和个性化定制。

技术方案（需要再回顾）

- 前端使用twaver 做拓扑的画图和右键，tip，云层，分组等功能。
- 后端的核心数据库有试图表，本地节点表，节点类型表。扩展围绕核心数据进行扩展。节点类型用于存储该节点对应的不同图标菜单等属性，节点表存储位置信息。
- 节点和关系存储与CMDB配置管理库，通过配置管理库拿到节点和关系，再计算位置信息返给前端。

运维自动化

项目介绍

目标为构建运维的自动化，比如一个告警事件可以通过事先编排好的流程图进行自动化处理，最终实现告警的闭环。

技术方案

- 前端使用bpmn库做流程图编辑
- 使用codemirror库做脚本录入从而作为流程图的某一个原子操作或者流程图的某一步功能
- 后端使用flowable对前端传过来的流程图xml或者json格式做解析，获取事件的先后顺序。
- 动作实现，使用stackstorm架构理念，使用ansible做python脚本或者ssh远程执行命令

分析器的堆叠双主隔离特性

项目介绍

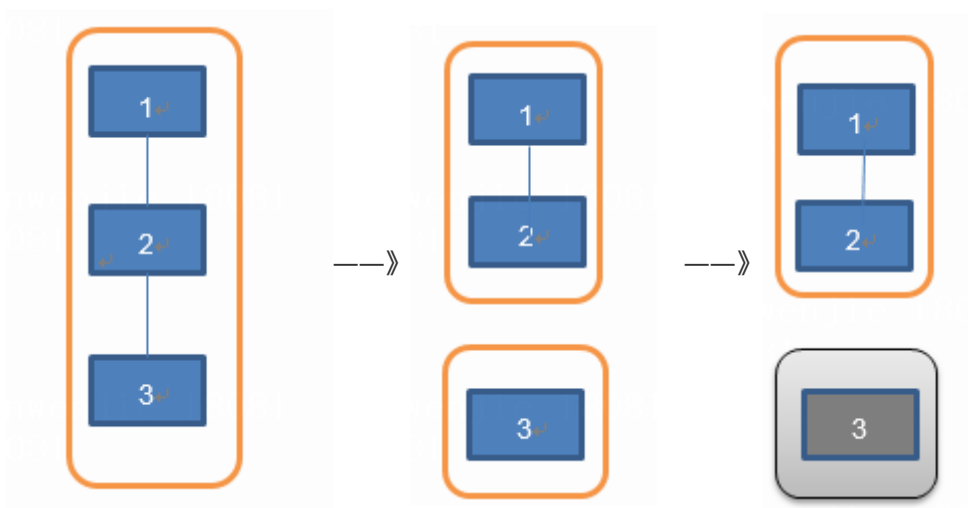
在可靠性网络场景中，一般为了保证冗余，会对两个设备组合成堆叠设备。类似于集群，两个设备之间存在堆叠链路，对外显示堆叠ip。当网络故障时，会发生堆叠分裂，此时就有可能业务受损。为了解决堆叠双主分裂问题，使用协议进行堆叠双主检测，并使用流处理程序实时判断，检测到分裂后通过kafka发送异常告警，通知前端进行隔离动作的下发。

技术方案

- 用户增加设备时，分析器检测该设备是否满足正常堆叠设备的状态，如果满足，检测通过，当做一个堆叠设备处理，并存库，预置为初始状态
 - 成员设备role和id，通过role=master，得出master id
 - 成员设备有个member id列表，可以得出堆叠包含哪些设备
 - 成员设备的接口，接口名称，接口状态

以上三点与master一致则认为是正常的堆叠设备

- 开启对设备netconf的周期性采集堆叠设备的状态数据，并进行比较，如果满足分裂条件，认为分裂，进入分裂处理，并发送kafka消息
 - 采集各设备的role 和member id ，如果不一致，则说明分裂，有两个role，拿到了master id1 和master id2
 - 采集各设备的成员编号，可判定当前分裂成双主的组成，{member id 1，member id2}，{member id3}
 - 采集各成员的IRF端口状态，可以判断哪个端口down了
- 前端收到kafka消息后闪烁提示，通过预先配置的策略，可以自动进行隔离处理，也可以等待用户处理，只做提示。
 - 根据各成员设备的优先级字段判定留下的master设备，和被隔离的master设备，优先级大的留下，优先级小的端口被shutdown
 - 根据数据库中的member id选择小的id对应的主设备留下，大的主设备端口shutdown



应用组件部署

项目介绍

作为统一运维平台的部署管理模块，提供了各组件包安装在该平台上的服务。提供上传包，包的有效性检测，包的解析，包的依赖，以及定制该应用包的场景选择，网络配置，方案配置以及和业务具体相关的一些配置。

技术方案

- 前端使用angular架构，使用语言为typescript，主要是在前端写逻辑写模块，展示。

OS兼容性测试

项目介绍

主要负责华为各种服务器产品（不同cpu，架构）上的OS兼容性测试，包括主流linux，虚拟化OS和windows，测试其能否正常安装，cpu的压力测试，硬盘的读写测试，网卡读写测试，内存压力测试，还有一些常见工具的安裝，ftp，nfs等

技术介绍

- 涉及到rhel，centos，suse，Ubuntu，vmware，xen，hyper-v等操作系统的安裝
- 网络相关测试工具，ifconfig，ipaddress，iptable，firmware，systemctl，网络配置文件，apt源的安裝等
- cpu的压力测试，内存的压力测试，ptu，stress
- 硬盘读写，网卡读写测试

BMC自动化测试

项目介绍

BMC为服务器上一块带外管理芯片，内置了服务器的管理程序，可以通过BMC对服务器做硬盘，cpu，内存等的管理，还可以通过kvm查看服务器的业务系统界面

技术介绍

- python自动化脚本开发
- mysql
- ipmi 接口，snmp接口，cli接口，rest接口

- mkdir
- wget
- 安装部署linux上各种工具，如配置FTP，NFS，kafka，zookeeper