

基于微服务的通用性系统日志评估方案及应用^①



邵 瑛¹, 徐 斌²

¹(上海电子信息职业技术学院 电子技术与工程学院, 上海 201411)

²(上海诺基亚贝尔股份有限公司 固网事业部, 上海 201206)

通信作者: 徐 斌, E-mail: bxu006@sina.com

摘 要: 针对系统日志类信息譬如系统运行日志、命令行文本输出等信息的分析评估, 提出了一种基于微服务的通用性评估方案. 该方案中的框架采用微服务构架, 单控制节点/多工作节点模式, 各个节点都具备相同的能力. 节点角色控制节点或工作节点是按需进行配置, 工作节点的数量可灵活扩展以支持大数据分析能力, 支持 TB 级的系统日志分析评估. 相对于目前流行的 ELK 等开源技术, 本方案除具有类似日志提取、分析和图表呈现等功能外, 还设计和实现了基于配置驱动的系统日志的通用性评估模型. 该模型采用规则与引擎分离的原则, 适用于各种系统日志类信息的处理. 基于提出评估方案, 采用 Spring Boot/Spring Cloud 微服务框架实现了一套 PON 网络健康检查系统. 该系统在全球多个 PON 网络评估服务中得到了很好的应用, 极大地提高了服务效率.

关键词: 微服务; 系统日志; 通用性; 评估方案; 配置驱动

引用格式: 邵瑛, 徐斌. 基于微服务的通用性系统日志评估方案及应用. 计算机系统应用, 2022, 31(1): 124-131. <http://www.c-s-a.org.cn/1003-3254/8334.html>

Log Evaluation Scheme and Application of General System Based on Microservice

SHAO Ying¹, XU Bin²

¹(School of Electronic Technology and Engineering, Shanghai Technical Institute of Electronics and Information, Shanghai 201411, China)

²(Fix Network Business Group, Nokia Shanghai Bell Co. Ltd., Shanghai 201106, China)

Abstract: This study proposes a general evaluation scheme based on microservice for the analysis and evaluation of system log information such as system operation log and command-line text output. The framework of this scheme adopts a microservice architecture and a model of one single master node and multiple worker nodes. Each node has the same ability. The node role, namely, whether the node is a master or worker one, is configured on demand. The number of worker nodes can be flexibly expanded to support the big data analysis ability and the TB-level system log analysis and evaluation. Compared with the popular open-source technologies such as elasticsearch logstash kibana (ELK), this scheme not only has similar functions of log extraction and analysis and chart presentation but also designs and implements a general evaluation model of system logs based on the configuration driver. The model, adopting the principle of separating rules from the engine, is suitable for processing many kinds of system log information. Under the proposed evaluation scheme, a PON network health examination system is built with the Spring Boot/Spring Cloud microservice framework. The system has been applied in many PON network evaluation services around the world and has greatly improved service efficiency.

Key words: microservice; system log; versatility; evaluation scheme; configuration drive

① 基金项目: 中国高校产学研创新基金 (2019ITA03035)

收稿时间: 2021-04-04; 修改时间: 2021-04-29, 2021-05-28; 采用时间: 2021-06-08; csa 在线出版时间: 2021-12-17

当前,计算机已经融入到了我们的日常生活中,生活的各个方面都会涉及到各种计算机系统,为了监视这些系统运行是否正常并能在系统出现故障时帮助用户及时定位,系统中通常会设计日志系统来输出一些有用的日志信息,很多系统还提供了各种人机接口命令(CLI)用来获取系统的配置、告警和状态等信息,这些信息也是以一定的文本格式输出.通过对这些信息的充分分析和评估,可以有效地帮助用户对系统目前的运行状况譬如安全性^[1]、负荷信息^[2]等进行深度了解并采取必要的优化措施.许多科研技术人员在这方面也做了很多研究工作^[3,4],并开发了ELK^[5-8]等类似的系统文本日志处理框架,但这些研究或框架在功能上普遍缺乏通用性研究,特别是缺乏通用的可配置的日志评估功能及对微服务的支持.本文基于该需求,设计了一种系统日志的通用性评估方案,该方案包括架构的设计和系统日志通用性评估模型的设计.在架构设计上,采用了微服务架构,实现是基于Spring Boot/Spring Cloud^[9]来实现,部署支持Docker Stack方式和K8S方式部署.在系统日志通用性评估模型方面,采用了引擎与规则分离的原则,来实现评估的通用性.

1 基于Spring Boot/Spring Cloud的系统评估框架的设计

2014年“微服务架构”^[10,11]被提出后,其思想经由Martin Fowler阐述后,在近几年持续受到重视,理论与相关实践都不断发展,目前它已经成为了主流

软件架构模式.微服务将一个大的系统提供的功能分解成了许多小的服务,这些小的服务都是独立实现、这些小的服务之间一般采用轻量级的通信协议,小的服务之间是松耦合且独立部署,这些小的服务就被称为微服务.微服务架构具有如下的一些特点:

- (1) 每个服务足够内聚,足够小,代码容易理解;
- (2) 服务独立测试、部署、升级、发布;
- (3) 每个服务可以各自进行扩展,每个服务可以根据自己的需要部署到合适的硬件服务器上;
- (4) 容易扩大开发团队,可以针对每个服务(service)组件开发团队;
- (5) 提高容错性(fault isolation),一个服务的内存泄露并不会让整个系统瘫痪.

1.1 评估框架总体架构设计

在图1总体架构中,控制节点和工作节点在系统微服务能力上完全相同,区别在于启用的微服务能力不一样.工作节点开启的微服务能力包括日志的上传、集成性检查、解析、组织入库,而控制节点则负责汇总各个工作节点的数据,并根据配置的健康检查规则对数据进行健康检查,同时根据评估规则生成项目的评估报告

1.2 节点上微服务架构的设计

日志的分析一般的流程是收集、解析、入库和分析,节点上微服务构架如图2,也是基于该流程来设计的,除此之外,本框架中还加入了评估功能.

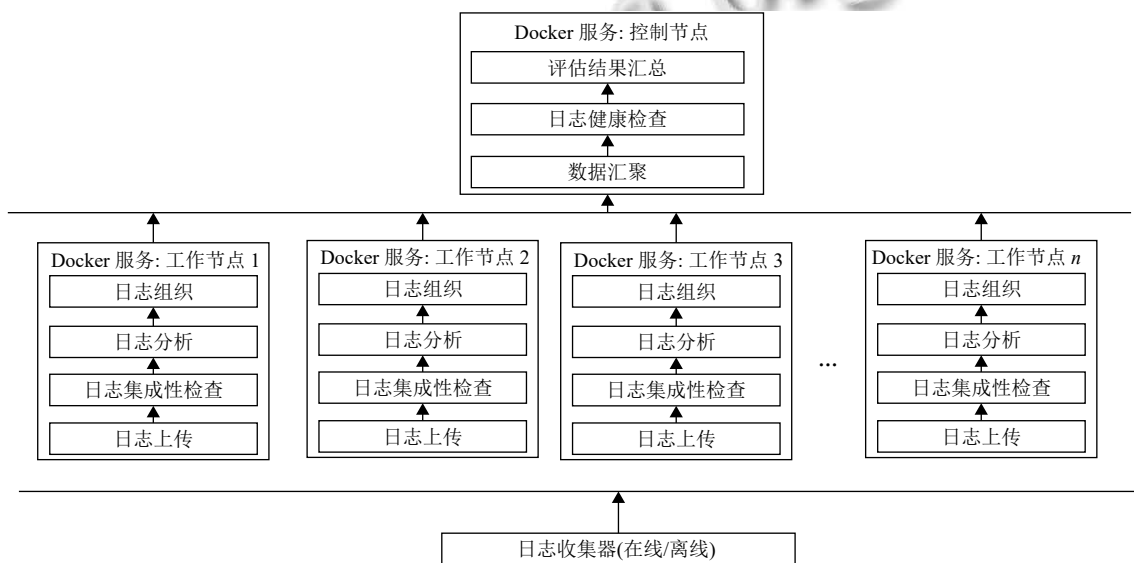


图1 评估框架总体架构

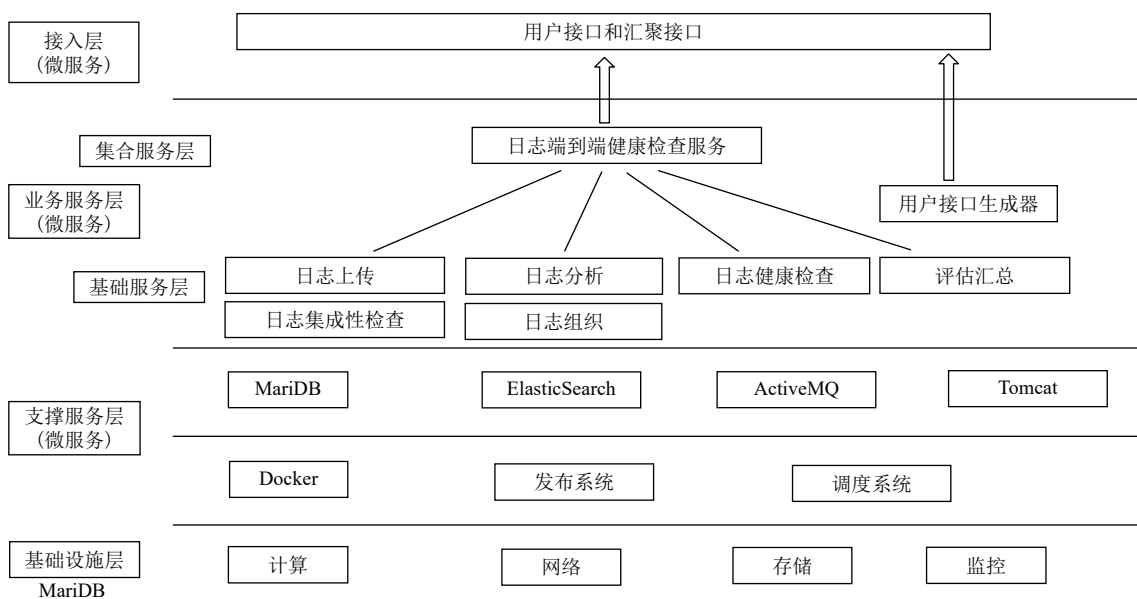


图2 节点上微服务架构的设计

在图2的微服务架构中,支撑服务层包括 MariaDB, ES 和 ActiveMQ, ES 用作原始 log 分析后的存储, MariaDB 用作组织后的数据的存储, ActiveMQ 用作前后端以及基础服务层之间的消息通信的中间件. 基础服务的各个部分都作为独立的服务来运行, 这些服务都是基于 Spring Boot/Spring Cloud 框架来开发. 利用 Spring Boot/Spring Cloud, 不但可以方便地实现微服务的开发, 而且还能方便地集成对 MariaDB, ES, ActiveMQ 等服务的访问.

2 系统日志通用性评估模型设计

方案中设计了两种模型.

(1) 通用日志分析模型

日志分析模型拆分成了分析引擎和分析规则, 通过在分析规则中配置数据源来实现分析引擎与数据源的松耦合, 分析引擎不针对某个特定的数据, 只根据分析规则来对数据信息进行处理.

(2) 通用日志评估模型

日志评估模型拆分成了评估引擎和评估规则, 通过在评估规则中配置数据源来实现分析引擎与数据源的松耦合, 评估引擎不针对某个特定的数据, 只根据评估规则来对数据信息进行处理.

通用日志分析模型包括分析引擎和分析规则, 在模型中, 分析引擎通过分析分析规则来实现对系统日志的分析检查.

2.1 通用日志分析模型

(1) 模型工作的前置条件

- 1) 被分析数据存放在关系数据表中;
- 2) 每张数据表需要一个额外的字段来存储分析结果, 这里暂命名为“checkrs”.

(2) 相关定义

1) 数据集: 将要被检查的同一张关系数据表中的所有数据称为一个数据集. 假设数据集有 l 列 m 行, 其表示如下:

$$D = \{d_1, d_2, \dots, d_l\} = \begin{bmatrix} d_{1,1} & \dots & d_{1,l} \\ \vdots & \ddots & \vdots \\ d_{m,1} & \dots & d_{m,l} \end{bmatrix}$$

$$d_i = \begin{bmatrix} d_{1,i} \\ \vdots \\ d_{m,i} \end{bmatrix}, 1 \leq i \leq l$$

2) 有效数据集: 数据集中具有对应检查规则的列的数据组成的数据集. 假设有效数据集有 n 列, 其表示如下:

$$V = \{v_1, v_2, \dots, v_n\} = \begin{bmatrix} v_{1,1} & \dots & v_{1,n} \\ \vdots & \ddots & \vdots \\ v_{m,1} & \dots & v_{m,n} \end{bmatrix}$$

$$v_i = \begin{bmatrix} v_{1,i} \\ \vdots \\ v_{m,i} \end{bmatrix}, v_i \in D, 1 \leq i \leq n$$

3) 分析规则集: 用来分析有效数据集 V 的规则集

集合, 规则集里规则的列数必须与有效数据集 V 的列数一致. 假设分析规则集有 p 行规则, 其表示如下:

$$R = \{r_1, r_2, \dots, r_n\} = \begin{bmatrix} r_{1,1} & \dots & r_{1,n} \\ \vdots & \ddots & \vdots \\ r_{p,1} & \dots & r_{p,n} \end{bmatrix}$$

$$r_i = \begin{bmatrix} r_{1,i} \\ \vdots \\ r_{p,i} \end{bmatrix}, 1 \leq i \leq n$$

4) 分析结果标记集: 当有效数据 $v_{ji} (1 \leq i \leq n,$

$1 \leq j \leq p)$ 不能满足规则时输出的结果标记, 其中 e_i 与 r_i 一一对应, 其表示如下:

$$E = \{e_1, e_2, \dots, e_n\}$$

(3) 分析算法

分析算法见图 3. 该算法中, 输入包含: 分析规则集、规则中关联的有效数据集、及对应的分析结果标记集, 输出为分析结果数组.

(4) 分析规则的设计

在本模型中, 设计了 4 种分析规则.

```

Input: V,R,E
Output: checksr[]
for k=0;k=m;k++ do
    checksr[k]=敏感;
    **检查数据是否满足某行规则
    for i=0;i=p;i++ do
        isRowMatched=True;
        for j=0;j=n;j++ do
            if(vkj not matchrij) then
                isRowMatched=False;
                break;
            end
        end
        if isRowMatched then
            break;
        end
    end
    if isRowMatched then
        checksr[k]="OK";
    else
        **检查各列数据是否满足对应规则中的任意规则
        for i=0;i=n;i++ do
            isColMatched=False;
            for j=0;j=p;j++ do
                if(vkj matchrij) then
                    isColMatched=True;
                end
            end
            if not isColMatched then
                checksr.append(ei);
            end
        end
        end
        **处理对于所有的列都能找到匹配规则但却无法匹配行规则的情况
        if checksr[k] == 敏感 then
            checksr[k] = "NoMatched";
        end
    end
end
    
```

图 3 分析算法

第 1 种规则: 相等规则. 分析规则中配置数字或字符串, 分析引擎对有效数据与规则中配置的数据进行相等性判断;

第 2 种规则: 正则表达式规则. 规则以“!”开始表示正则表达式规则, 分析规则中配置正则表达式, 分析引擎判断有效数据是否满足规则中提供的正则表达式;

第 3 种规则: 数字范围规则. 分析规则中包含数字范围, 输入方式为 $[X-Y]$, 表示数字在 X 与 Y 之间都是有效的. 例如规则“vlan[1-100]”, 那么 vlan1, vlan2, ..., vlan100 都是合法的.

第 4 种规则: SQL 条件规则. 规则中以“?”开始表示 SQL 规则. 该规则以 SQL 的条件子句为分析规则. 例如: “?F1>100 and F2=1000”, 表 1 是一条分析规则的例子.

表 1 分析规则

F1	F2	F3	F4
SH	![0-9]+	vlan[100-300]	?F4>100 and F4<200

这条规则将对有效数据进行分析, 检查 F1 字段是否是字符串“SH”, 判断 F2 字段是否是数字, 判断 F3 字

符串是否是 vlan100 或 vlan101 或...或 vlan300, 判断 F4 字段是否大于 100 且小于 200.

(5) 模型的实现例子

下面是一个对宽带网络用户利用率进行分析检查的例子. 数据集如表 2, 有效数据集如表 3, 分析规则集和分析结果标记如表 4. 表 4 中第 1 行是对应的有效数据集中的字段名; 第 2 行是分析规则集; 第 3 行是分析结果标记集. 分析结果如表 5. 通过检查, 发现用户的可利用带宽的配置都不正常, 分析结果见 checksr 列中的结果标记.

2.2 通用日志评估模型

通用日志评估模型同样包括评估引擎和评估规则, 在模型中, 评估引擎同样通过评估规则来实现对日志的评估.

(1) 相关定义和解释

数据集: 根据某个或某些特征组合在一起数据, 这些数据存储多个关系数据表中. 其表示如下:

$$\begin{cases} D = \{t_1, t_2, \dots, t_n\}, t_i = \{f_{i1}, f_{i2}, \dots, f_{im}\} \\ t_i \in D (1 \leq i \leq n) \end{cases}$$

其中, t_i 为一张二维数据表的所有数据, $f_j(i_1 \leq j \leq i_m)$ 为表 t_i 中的一列数据.

规则. 其表示如下:

$$R = \{r_1, r_2, \dots, r_k\}$$

评估规则: 一组用来对数据中的数据进行评估的

其中, r_i 作用在指定表的指定列(字段)的数据上.

表2 数据集

ip_address	pon	mc_avail_bw	mc_used_percent (%)	uc_avail_ds_percent (%)	uc_avail_us_percent (%)
10.17.41.23	1/1/1/3	1 600 000	1.00	100.00	100.00
10.17.41.23	1/1/1/4	1 600 000	5.20	100.00	100.00
10.17.41.23	1/1/1/5	1 600 000	0.00	100.00	100.00

表3 有效数据集

mc_avail_bw	mc_used_percent (%)	uc_avail_ds_percent (%)	uc_avail_us_percent (%)
1 600 000	1.00	100.00	100.00
1 600 000	2.00	100.00	100.00
1 600 000	0.00	100.00	100.00

表4 分析规则集和分析结果标记集

mc_avail_bw	mc_used_percent	uc_avail_ds_percent	uc_avail_us_percent
300 000	?replace(mc_used_percent, '%', '')+0.0 < 80	?replace(uc_avail_ds_percent, '%', '')+0.0 > 40	?replace(uc_avail_us_percent, '%', '')+0.0 > 40
OK-Wrong mc_avail_bw	NOK-Wrong mc_used_percent	NOK-Wrong uc_avail_ds_percent	NOK-Wrong uc_avail_us_percent

表5 分析结果

mc_avail_bw	mc_used_percent (%)	uc_avail_ds_percent (%)	uc_avail_us_percent (%)	checkrs
1 600 000	1.00	100.00	100.00	NOK-Wrong mc_avail_bw
1 600 000	5.20	100.00	100.00	NOK-Wrong mc_avail_bw
1 600 000	0.00	100.00	100.00	NOK-Wrong mc_avail_bw

排除键值 (EV): 数据评估的时候, 用来清洗被评估数据. 当被评估数据包含排除键值时, 那么该数据将被过滤掉. 在规则中设置.

期望键值 (KV): 被评估数据的期望值. 在规则中设置.

评估权重 (W): 该项占百分值评估的分值权重. 在规则中设置.

有效数据: 如果被评估数据包含期望键值, 该数据即为有效数据.

负面数据: 如果被评估数据不包含期望键值, 该数据即为负面数据;

作用类型 (PN): 配置被评估对象的数据取值为期望键值时, 获取的评估分值是正还是负. 在规则中设置.

有效范围 (vScope): 有效策略为计数统计策略和平均值策略的时候, 统计得到的数据计算评估得分时的判断依据.

评估方法策略 (M): 本模型中定义了 3 种评估策略:

- 百分占比策略: 计算有效数据数占总数 (有效数据加负面数据) 的百分比, 然后根据计算出的百分比来计算该项的评估得分.

- 计数统计策略: 按照指定条件分组并统计各分组中不同数据出现的次数. 并把得到的这些次数与某个设定范围 (有效范围 vScope) 进行比较来计算该项的评估得分.

- 平均值策略: 按照指定条件分组并统计各分组中数据的平均值. 并把得到的这些平均值与某个设定范围 (有效范围 vScope) 进行比较来计算该项的评估得分.

评估方法策略在规则中设置.

(2) 评估算法

评估算法见图 4. 该算法中, 输入包含: 评估规则及规则中关联的数据, 输出为评估得分数组 score.

(3) 模型的实现例子

1) 规则的定义

由于规则数百条,表6是小部分规则例子.

2) 评估结果

根据数据得到的评估结果如表7所示,见score(得分)列.

```

Input: D,R
Output: score[]
for i=0;i<len(R);i++ do
  score[i]=0;
  m=getPolicy(ri);
  tableName=getTablename(ri);
  fieldName=getFieldname(ri);
  kv=getExpectValue(ri);
  ev=getExcludeValue(ri);
  w=getWeight(ri);
  pn=getPositiveNegtiveFlag(ri);
  vScope=getVScope(ri);
  if (m is "count_percentage") then
    count=0;
    allData[]=getData(D,tableName,fieldName);
    data[]=filter(allData,ev);
    for j=0;j<len(data);j++ do
      if data[j] in kv then
        count++;
      end
    end
    score[i]=(count/len(data))*w*pn;
  else
    if(m is "count_distinct") then
      allCountDataGKey=getDataByGroupKey(D,
      tableName,fieldName,kv);
      (data[],gKey[])=filter(allCountDataGKey,ev);
      vmap={};
      for j=0;j<len(data);j++ do
        if(vmap has key:gKey[j] then
          vmap{gKey[j],data[j]}++;
        else
          vmap{gKey[j],data[j]}=1;
        end
      end
    end
  end
  score[i]=w*pn
  for(key,value) in vmap then
    if value not in vScope then
      score[i]=0;
      break;
    end
  end
end
end
end

```

图4 评估算法

表6 评估规则

分类	评估方法策略	数据库表名	字段名	期望键值	排除键值	作用类型	有效范围	权重
Performance	count_average	go_ont_check	ont_tx_sinal_level	columns=ont_tx_sinal_unknow&unsupportt _level ed&invalid&nodata		Positive	[2.5,3.5]	5
Performance	count_percentage	go_ont_check	temperature_check	Ont-temp- high&Ont0temp-low	NA	Negative	[20,70]	7
Performance	count_average	go_ont_check	ont_temperature	Columns=ont_temperat_unknow&unsupportt ure ed&invalid&nodata		positive	—	5
Performance	count_percentage	go_ont_check	ont_tx_seg	NOK&warning	NA	Negative	—	8
Inventory	count_percentage	go_cpe_inventory	checkrs	NOK-Wrong m_vendor_name	NA	Negative	—	3
Inventory	count_percentage	go_cpe_inventory	checkrs	NOK-Wrong s_vendor_name	NA	Positive	—	3
Inventory	distinct_type	go_cpe_inventory	s_vendor_name	Columns=s_vendor_na me	NA	Negative	[1,5]	5

3 基于微服务的系统日志通用性评估方案的 PON 网络健康检查系统的实现

PON 是一种典型的无源光纤网络,是指(光配线网中)不含有任何电子器件及电子电源,ODN 全部由光分路器(splitter)等无源器件组成,不需要贵重的有

源电子设备.一个无源光网络包括一个安装于中心控制站的光线路终端(OLT),以及一批配套的安装在用户场所的光网络单元(ONUs)目前 PON 网络作为接入网在全球得到了广泛的应用,小的 PON 接入网可能只有几十台 OLT,大的可能有数万甚至数十万台 OLT,

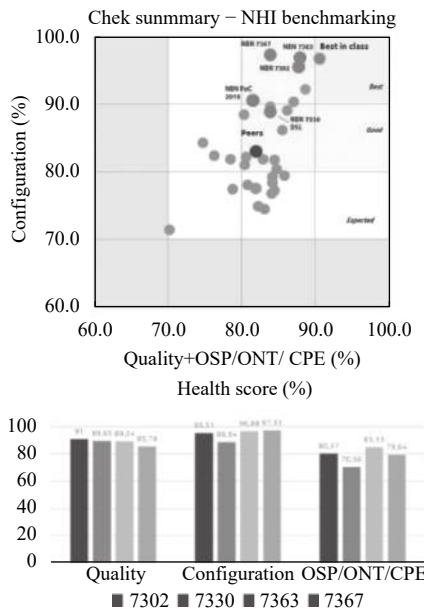
ONU 的数量相对于 OLT 来说,更是成倍甚至数十倍,如何维护和管理这些 OLT 和 ONU,特别如何对正在服务的 PON 网络进行一个健康情况的评估并进行优化就变得特别重要。

基于本文介绍的基于微服务的系统日志通用性评估方案,作者等开发了一套 PON 网络健康检查系统,目前该系统已经在多个 PON 网络的健康评估中得到了应用。处理过的最小的网络只需要建立一个节点来处理,该节点既是工作节点,又是控制节点;最大网络

建立了 7 个工作节点,1 个控制节点来处理,处理的日志约 1 TB,该 PON 网络有 25 000 多个 OLT 上百万个 ONT。从日志开始检查,分析到最后输出数百张图表及最终的评估报告,总的耗时约 7 小时。该系统在实际的项目中对 PON 网络的健康服务起到了非常重要的作用,极大地提高了服务的效率,得到了用户的高度评价。图 5 为某外国电信运营商利用基于本文介绍的分析评估模型开发的系统进行网络评估得到的评估结论。

表 7 评估结果

分类	评估方法策略	数据库表名	字段名	期望键值	排除键值	作用类型	有效范围	权重	得分
Performance	count_average	go_ont_check	ont_tx_sinal_level	columns=ont_tx_sinal_level	unknown&unsupport&invalid&nodata	Positive	[2.5,3.5]	5	1.40
Performance	count_percentage	go_ont_check	temperature_check	Ont-temp-high&Ont0temp-low	NA	Negative	[20,70]	7	2.33
Performance	count_average	go_ont_check	ont_temperature	Columns=ont_temperature	unknown&unsupport&invalid&nodata	Positive	—	5	1.67
Performance	count_percentage	go_ont_check	ont_tx_seg	NOK&warning	NA	Negative	—	8	2.60
Inventory	count_percentage	go_cpe_inventory	checkrs	NOK-Wrong	NA	Negative	—	3	1.00
Inventory	count_percentage	go_cpe_inventory	checkrs	NOK-Wrong	NA	Positive	—	3	0.00
Inventory	distinct_type	go_cpe_inventory	s_vendor_name	Columns=s_vendor_name	NA	Negative	[1,5]	5	0.00



Areas	Key Outcomes	Additional Comments	Impacted KPI
Quality	Overall FTTH network quality score is almost best in class, DSL is lower (7363 micro node is the exception for copper)	Optical and copper quality from SFP to ONT would benefit from a deeper analysis and a continuation of remediation programs	RSP SLA, End user QoE
Configuration	Very high control for copper micro-nodes (best in class overall) and FTTH networks (2nd best in class) but signs of degradation on 7330 copper	Most of the impact on the configuration score comes from the benchmarking of network scalability check points. If the traffic behavior and its associated customer expectation are changing through out the network, it might become a pressure point, which could be evaluated against the current design/engineering rules	Operational efficiency
Overall	The NHI scores of the NBN access networks are among the best as currently observed by Nokia Health Check Assessment program	The difference in config score between FTTH/micro nodes and 7330 nodes could also indicate that in terms of NHI score the potential exists for the 7330 platform to reach best in class level.	Network operations cost, Quality

Key Actions	What
Review Meetings	5 reviews planned: 7302, 7330 and 7363/67 at operational level. 7302/7330 and 7363/67 at executive level
Annual Health Check	Suggested as part of the journey
Short term improvement	Validate a program based on Health Check results
Long term improvement	4 steps approach on quality and network cost optimization use case covering: <ul style="list-style-type: none"> Capacity engineering OSP optimization Change management/Configuration optimization KPI framework optimization Network cost of operations optimization Implementing predictiveness from a reactive to a proactive system

图 5 分析评估模型应用实例

4 结论

本文介绍了一种系统日志的通用性评估方案,该

方案采用单控制节点/多工作节点模式,扩展灵活,支持 TB 级的系统日志数据分析和评估;相对于当前流行

的 ELK 等方案, 设计了配置驱动的评估方法, 具有很好的通用性, 同时该方案方便定制性的二次开发. 基于该方案而实现的 PON 网络健康检查评估系统在实际中也得到了很好的应用, 极大地提升了服务的效率. 该方案对于今后需要进行系统日志类信息分析和评估的工作人员具有非常好的参考和借鉴作用.

参考文献

- 1 刘百祥, 周超. 高校环境下日志管理的思考与设计. 网络空间安全, 2019, 10(10): 44–48. [doi: 10.3969/j.issn.1674-9456.2019.10.008]
- 2 樊建永. 日志综合审计系统在高校中的应用研究. 卫星电视与宽带多媒体, 2019, (12): 35–37.
- 3 康金辉. 一种数字校园日志获取模型及实现方法. 计算机与数字工程, 2009, 37(4): 178–181. [doi: 10.3969/j.issn.1672-9722.2009.04.054]
- 4 张鹏. 利用 Elastic Stack 构建集中运维日志分析平台. 电子技术与软件工程, 2018, (15): 133–134.
- 5 李剑, 陈俊杰, 郑金帆. 应用 ELK 构建图书馆日志统一管理分析平台. 情报探索, 2020, (3): 71–76.
- 6 姚姜虹. 基于 ELK 企业级日志采集分析平台搭建方案研究. 信息系统工程, 2020, (10): 119–120. [doi: 10.3969/j.issn.1001-2362.2020.10.053]
- 7 Lu CJ, Zeng H, Liu JY, *et al.* Network security log analysis system based on ELK. Proceedings of 2017 International Conference on Computer, Electronics and Communication Engineering. Lancaster: DEStech Publications, 2017. 554–559.
- 8 Andre JM, Behrens U, Branson J, *et al.* A scalable online monitoring system based on elasticsearch for distributed data acquisition in CMS. Proceedings of the 23rd International Conference on Computing in High Energy and Nuclear Physics. Les Ulis, 2019.
- 9 黄强文, 曾丹. 基于 Spring Cloud 和 Docker 的分布式微服务架构设计. 微型电脑应用, 2019, 35(6): 98–101. [doi: 10.3969/j.issn.1007-757X.2019.06.030]
- 10 Lewis J, Fowler M. Microservices a definition of this new architectural term. <https://martinfowler.com/articles/microservices.html>. [2014-05-25].
- 11 张晶, 黄小锋, 李春阳. 微服务框架的设计与实现. 计算机系统应用, 2017, 26(6): 259–262. [doi: 10.15888/j.cnki.csa.005796]