面向大规模集群的网元自动巡检系统设计与实现

朱云杰

（中博信息技术研究院有限公司，江苏 南京 210012）

摘　要：如何将新建与升级改造后的网元快速纳入网元巡检系统，以及控制网元或网元组上巡检任务数量和优先级，是现有网元巡检系统中迫切需要解决的问题。为此，基于微服务（SpringCloud）与分布式内存数据库（Redis）技术设计网元数据统一采集平台，在此基础上实现网元巡检任务的调度执行。系统运行测试结果表明，通过巡检任务调度配置以及在采集平台部署微服务，能够快速完成系统的平滑扩容，并解决网元巡检任务的限流与优先级问题。

关键词：网元巡检；任务调度；限流；逻辑模板；微服务；分布式内存数据库

DOI：10.11907/rjdk.211244　　**开放科学（资源服务）标识码（OSID）：**

中图分类号：TP319　　**文献标识码：**A　　**文章编号：**1672-7800（2022）001-0216-06

Design and Implementation of Network Element Automatic Inspection System for Large-Scale Cluster

ZHU Yun-jie

（*Zhongbo Information Technology Reserch Institute Co.，Ltd.*，*Nanjing* 210012，*China*）

Abstract：How to bring the newly built and upgraded network elements into the network element inspection system quickly，and how to control the number and priority of inspection tasks on the network element or network element group，is an urgent problem in the existing network element inspection system. Therefore，based on the technology of SpringCloud and Redis，a unified collection platform of network element data is designed. On the basis of the collection platform，the scheduling and execution of network element patrol task is realized. The system running test results show that，through the patrol task scheduling configuration and the deployment of micro services in the collection platform，the system can quickly complete the smooth expansion of the system，and solve the problem of current limitation and priority of network element patrol task.

Key Words：network element patrol；task scheduling；current limiting；logical template；microservice；distributed memory database

0　引言

网元巡检是电信操作维护部门的一项重要工作，也是保证网元正常运行的必要手段。网元巡检的基本过程为：连接登录网元，向网元发送若干巡检指令，网元接收指令后返回相应的指令执行报告，最后分析返回的报告是否存在异常。

由于网元类型多、数量多，且检查项目多，因此人们开发了相关的网元自动巡检系统，以代替人工完成繁重的巡检任务。随着5G技术的快速发展，现有网元巡检系统暴露出一些突出问题，主要包括：新接入网元周期较长，新型网元接入时需要对现有系统进行二次开发，然后重新进行版本升级，无法做到快速支撑、快速响应；现有系统框架在数据采集与数据应用上采用紧耦合方式，数据采集能力不易开放给其他应用，数据采集需求存在大量重复开发且开发周期较长等问题；采集任务缺乏优先级控制，重点任务得不到保障，由于数据采集缺乏统一规划，同一网元上可能对接多种采集机与采集接口，存在大量重复采集、资源抢占、采集数据无法及时同步等问题，尤其对于重点采集任务无法第一时间执行；采集数据丢失，由于采集过程缺乏有效的监控预警，没有审计监控日志，因此采集数据的完整性、准确性无法得到保障。

本文设计与实现的系统采用微服务技术架构构建统

收稿日期：2021-03-12

作者简介：朱云杰（1970-），男，硕士，中博信息技术研究院有限公司高级工程师，研究方向为智能网管。

一的网元数据采集平台,实现网元自动巡检作业调度执行,采集平台可同时开放给其他系统应用,从而能较好地解决上述问题。

目前江苏电信公司已由不同厂家建设了若干网元巡检系统,如交换网元例行测试系统[1]。本文采用先进的集群技术,对原有网元巡检系统进行的改进与优化如下:

(1)基于高可用、可伸缩微服务架构设计,较原有系统的Monolith单体架构具有更强的服务能力[2-4]。随着巡检网元数量的不断增加,只须增加服务结点即可,而原有系统则需要更换性能更好的服务器。

(2)业务逻辑做到完全解耦,将业务逻辑内聚性高的业务置于单个微服务中(采用Spring Boot[5-6]开发),使得新增与修改业务逻辑变得更为方便。例如:原有交换网元例行测试系统[1]的网元数据采集接口具有数据采集、报告解析与异常分析功能,每次开发新的网元接口时,报告解析与异常分析都需要重新写一遍,可重用性差。

(3)建设统一的网元数据采集平台,采用分布式内存数据库Redis[7-9]对各个应用系统的数据采集任务进行统一调度,实现采集任务的限流与优先级调度。例如:IMS网元数据采集需使用数据总线,但该总线除交换网元例行测试系统使用外,其他厂家的巡检系统也会使用,因此经常会出现总线拥堵的情况。

(4)采用采集数据缓存方法,某些巡检任务在一定时间间隔内变化微小,可将采集的数据存放于共享内存数据库并设置存放时长,在此时间内若再有该采集任务,则用缓存内采集的数据作为本次采集任务的结果数据,从而减少网元上数据采集产生的负荷。

(5)采用集群技术部署逻辑服务,使系统更加稳定、可靠。对原有系统采用主程序+看门狗程序进行加固,当某个逻辑服务发生故障时,不影响系统正常运行。

(6)采用开源的XXL-JOB任务调度平台[10]进行开发,能够对巡检任务执行情况进行有效监控,通过查看日志跟踪错误。

1　系统整体架构设计

网元自动巡检系统整体分为3部分:巡检任务调度执行系统、统一网元数据采集平台和网元。统一网元数据采集平台采用SpringCloud微服务框架实现,也主要包括3部分:数据采集任务路由服务、网元数据采集任务网关和网元数据采集服务,该平台可作为通用服务开放给其他系统使用。系统整体架构如图1所示。

1.1　巡检任务调度执行

采用开源的分布式任务调度平台XXL-JOB作为基础架构进行开发,主要分为3个模块:网元巡检任务调度控制中心模块、巡检任务执行器模块和采集报告告警分析模块。各模块完全解耦,以提高系统整体的稳定性和扩展性。

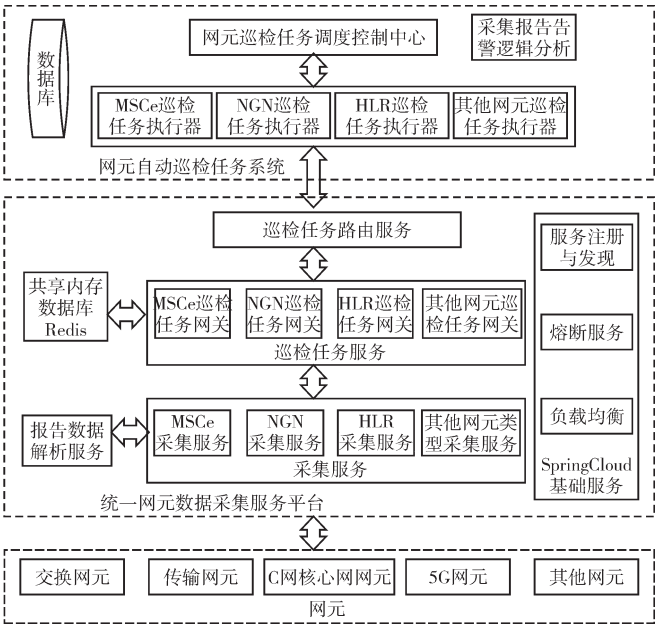

Fig. 1　Overall system architecture

图1　系统整体架构

调度中心采用中心式设计,执行器支持集群部署,通过在注册中心周期性地自动注册执行任务,调度中心会自动发现执行器,并使用Cron策略调度执行。

1.1.1　调度控制中心

调度中心进行任务调度时,会解析不同任务参数发起远程调用,以调用各自的远程执行器服务。这种调用模型类似RPC调用,调度中心提供调用代理功能,而执行器提供远程服务功能。调度中心自身并不承担业务逻辑,只负责发起调度请求,将任务抽象成分散的JobHandler,交由执行器统一管理。执行器专注于任务执行等操作,负责接收调度请求并执行对应JobHandler中的业务逻辑。

为了实现网元自动巡检,控制中心设计以下功能模块完成巡检任务的配置、调度与监控。

(1)网元接口管理。对网元连接进行管理,包括:网元ID、网元名称、网元类型、网元组ID、接口IP、账号口令、接口版本、网元限流及网元组限流,网元限流小于等于网元组限流。

网元组是指多个网元使用同一个接口,例如:登录专业网管,从该网管上发送巡检采集数据指令,此时接口IP使用网管IP,若该网管上管理了数十个网元,为避免同一时刻因巡检采集数据在该网管上建立了数十个连接,而影响网管正常运行,必须配置网元组限流。在网元类型中定义一个特殊的“虚网元”作为网元组,“虚网元”作为网元组的父网元,并不生成巡检任务。当网元组中只有一个实体网元,此时网元ID等于网元组ID,且网元限流等于网元组限流;当网元组中有多个实体网元,此时网元ID不等于网元组ID,且网元限流小于等于网元组限流。

接口版本指定使用哪一个网元数据采集微服务,当网元的接口升级后,重新开发一个数据采集微服务并部署到统一的网元数据采集平台,此时修改接口版本后即可实现

动态、平滑地升级网元接口。

(2)作业项配置。在网元类型上配置作业项,作业项配置内容包括执行指令、作业项执行结果缓存时长与指令逻辑模板。一个网元类型配置有一个或多个作业项,一个作业项配置有一个或多个指令。如华为软交换(HWSS)配置的巡检作业项有5个:检查媒体网关设备状态、检查MTP目的信令点状态、处理器负荷检查、检查SCCP目的信令点状态、检查M3UA路由状态。检查MTP目的信令点状态配置的执行命令有2个:“LST N7DSP:;”和“DSP N7DSP:DPX=$D1$;”。

作业项执行结果缓存时长配置:某些作业项检查结果在短时间内不会出现较大变化,若此时再有采集任务,则用缓存内上次的采集报告作为本次采集结果。其优点是减少频繁地连接网元,减轻网元运行压力。

作业项执行超时时长配置:防止因执行巡检作业项时网元繁忙导致作业项执行超时,影响网元正常运行。

指令逻辑模板配置:逻辑模板分为数据逻辑与告警逻辑,数据逻辑用于作业项中某条指令参数依赖于上条指令执行结果。如华为软交换(HWSS)的“检查MTP目的信令点状态”作业项,第一条指令“LST N7DSP:;”列出所有目的信令点,第二条指令“DSP N7DSP: DPX=$D1$;”为指令模板,其中的参数$D1$通过执行数据逻辑得到所有目的信令点具体值,并实例化为多条可执行指令。告警逻辑用于分析指令执行返回报告中是否存在异常。

(3)执行器JobHandler管理。具体包括:执行器名称(AppName)、说明、任务网关、发布者、发布时间。执行器集群的唯一标识为AppName,执行器会周期性地以AppName为对象进行自动注册,供任务调度时使用。配置任务网关用于指明该执行器任务所使用的任务网关。

(4)任务调度管理。针对某一网元类型,将与巡检任务周期相同的若干作业项制定为一个巡检任务。配置项有:任务ID、执行器、任务执行计划、网元类型、网元列表、作业项列表、并发巡检数、任务等级、租户与任务描述。

执行器是该任务执行的载体,任务执行计划采用Cron表达式,调度中心根据该Cron表达式周期性地启动执行器,并将参数任务ID传递给执行器。

(5)调度监控。监控巡检任务调度执行状态,包括:任务ID、任务名称、执行器名称、调度时间、启动时间、调度结果(成功或失败)、执行状态(运行中或结束)。

调度结果若失败,能查看调度失败的具体原因。对正在执行的任务可人工操作中断或暂停,暂停后可重新启动。

(6)网元巡检报告,查看巡检作业项执行情况,包括:任务ID、任务名称、执行器名称、网元名称、网元类型、作业项名称、巡检时间、巡检结果(正常或告警)、报告详情。

1.1.2 任务执行器

执行器实际上是一个内嵌的服务,每个执行器都有机器集群的唯一标识ID、服务IP与端口。执行器启动后,通过心跳进行执行器注册,调度中心收到心跳实现自动任务发现。执行器任务是一个Spring的Bean类实例,在Spring容器中进行维护,在这个类中实现任务逻辑。

执行器的任务类通过注解定义唯一的任务ID,查询业务数据库得到具体巡检任务,包括:任务ID、任务优先级、网元类型、网元列表、作业项列表、任务网关URL、巡检网元并发数以及每个作业项对应的指令列表与指令逻辑。

任务被调度启动时根据巡检网元并发数N启动N个线程,线程执行步骤如下:①查询网元列表,若网元列表空则线程结束,否则从网元列表中取出1个网元,并将该网元从网元列表中移出;②依次从作业项列表中取出一个作业项以及该作业项对应的执行指令序列和指令逻辑;③生成系统唯一的巡检作业序列号,发送巡检任务参数(JSON格式)到统一网元数据采集平台中的巡检任务路由服务;④接收巡检返回报告时,通过消息中间件MQ发送消息给报告告警分析模块;⑤当该网元所有作业项执行完毕后,重复第一步。

待N个线程全部运行结束,执行器完成对某一网元类型的巡检任务。

1.1.3 巡检报告告警逻辑分析

报告逻辑分析模块作为消息中间件MQ的消息消费者,接收来自任务执行器发送的作业项,并执行返回报告分析请求。根据执行指令配置的告警分析逻辑模板对该指令执行报告进行分析,分析结果连同指令、原始报告、作业项信息、网元信息、任务信息一起存入业务数据库。

1.2 统一网元数据采集平台

平台接收来自上层应用系统的网元数据采集请求,请求参数包括:任务流水号、任务ID、任务优先级、作业项ID、网元ID、网元组ID、网元限流、网元组限流、缓存时长、采集超时时间、指令列表、指令分析逻辑、任务网关服务名、采集服务名。采集任务完成后,平台返回任务流水号、任务ID、作业项ID、网元ID、指令、指令逻辑、指令执行报告与采集时间。

1.2.1 巡检任务路由服务

该服务直接接收来自巡检任务调度中心任务执行器发送的巡检任务请求,从请求参数中解析出任务网关服务名,调用对应的服务网关并转发请求参数,同步等待返回采集数据。

根据网元数量,采集平台部署了多个该任务网关服务名的微服务,因此路由服务还具有负载均衡功能。负载均衡采用Netflix提供的框架Feign,Feign整合了Ribbon与Hystrix。Feign是一个声明式的Web Service客户端,完全代理HTTP请求,通过编写简单的接口和插入注解,即可定义好HTTP请求的参数、格式、地址等信息,使用极其方便。

1.2.2 任务网关

任务网关是采集平台的核心模块,任务网关微服务名按网元类型命名,具有同一服务名的任务网关通常有多个,以负载均衡方式接收来自路由服务的巡检任务请求。主要实现以下功能:①采集任务分发功能。根据巡检任务请求参数中的采集服务名,将采集任务分发到相应的采集

服务;②采集任务优先级控制。优先分发优先级高的采集任务;③限流功能。用于解决以往因采集任务频繁导致的资源抢占等问题,限流内容包括:单个网元限流和网元组限流;④采集数据缓存功能。将采集的数据按照请求参数中的缓存时长保存在采集平台中,若在缓存时长内有同样的采集任务,则使用缓存内采集的数据,以减轻网元压力。

1.2.3　采集服务

接收来自任务网关的采集任务请求,一个采集任务包括若干条有顺序的指令以及指令数据逻辑。根据请求参数中的网元连接参数,连接并登录网元,按顺序发送采集指令,获取返回报告。若指令配置有数据逻辑,则调用采集平台中的报告逻辑分析服务,得到报告中的关键数据作为后续指令参数。所有指令执行完毕后断开网元连接,将指令及指令报告返回给任务网关。

1.3　网元

网元层中的网元按照专业可划分为数据专业、传输专业、交换专业、无线专业等。按照类型可划分为:PSTN、NGN、MSCE、MGW、LSTP、AG、TG、HLR、MGCF、PCSCF、HSTP、OLT、BAS、ONU、SDH、MSAP、RRU、BSC等。

2　关键模块设计实现

2.1　任务网关设计

任务网关基于Spring Boot设计,分布式内存数据库Redis为任务网关集群所使用,并在任务网关进程内存上开辟作业项执行排队区和作业项执行报告区,作业调度模块根据作业优先级、限流、超时参数对作业项进行调度,作业执行模块对调度成功的作业项调用对应采集服务,取得作业项执行报告。任务网关结构设计如图2所示。

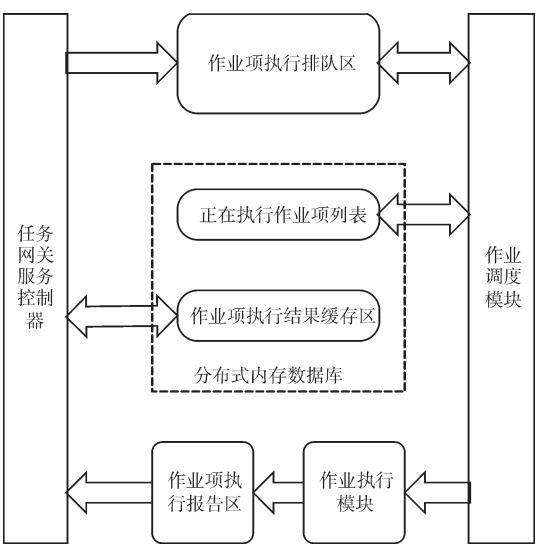

Fig. 2　Task gateway software module structure

图2　任务网关软件模块结构

任务网关接收来自任务路由的作业项请求,进行任务限流与优先级调度后,分发给相应的采集服务并接收返回报告。

2.1.1　任务网关服务控制器

采用注解@RequestMapping将任务路由HTTP请求映射到REST控制器的作业项执行方法上,该方法处理过程如下:

(1)解析执行的作业项参数(缓存时长)确定是否缓存采集报告。若缓存时长大于零,则从Redis中获取;若超时获取不到或缓存时长等于零,则将作业项放入作业项执行排队区。

(2)等待作业项调度结果,定时查询作业项执行区,若超过作业执行超时时间,则返回结果失败。获取到作业后,根据作业项参数中的采集服务名,将作业项转发到对应的采集服务。

(3)等待采集结果,若采集成功且缓存时长大于零,则将采集报告存放到Redis中,最后返回作业项执行结果。

2.1.2　任务网关内存缓存区

在任务网关进程内部创建作业项缓存区作为进程内部线程共享,因此在缓存区上必须加互斥锁进行访问。作业项缓存区分为作业项执行排队等待区和作业项执行报告区。

作业项执行排队等待区采用队列方式,队列中的每一项按照网元组ID进行区分,并存放对应该网元组ID内所有网元的作业项。在队列上实现放入与取出作业项两项操作。

每次放入一个作业项,根据该作业项的网元组ID扫描排队等待区,若队列中存在该网元组ID,则将作业项放入队列中的对应项;若不存在,则在队列中创建新项。

从排队等待区中取出作业项的调度策略如下:扫描排队等待区,删除等待超时的作业项,同一网元上作业项优先级高的先执行,相同优先级的作业项则等待时间长的先执行。通过网元组ID检索正在执行作业项列表,统计所在网元组正在执行的网元数量。对比所在网元组限流,只有在未超出设定的网元组限流时,方可取出并执行该网元的作业项。

作业项执行结果报告区同样实现放入和取出结果报告两个操作,作业项调度成功执行后放入执行结果报告,并采用任务网关服务控制器取出执行结果报告。

2.1.3　分布式内存设计

分布式内存数据库采用Redis,其是一个高性能的Key Value数据库,被任务网关集群共享,支持多种类型的数据结构,可实现数据缓存。

基于Spring与Redis的RedisTemplate工具类实现列表List上的以下操作:获取list缓存长度、通过索引获取list中的值、将list放入缓存、根据索引修改list中的某条数据,以及从list中移除某条数据。采用网元组ID为每个list创建唯一的Key,list中存放对应网元组中的网元作业项。

集群上多个任务网关存在同时操作共享内存中list的情况,同样采用Redis的RedisTemplate工具类实现分布式全局锁类,提供获取锁和释放锁两种方法。采用网元组ID为每个list创建唯一锁名称,当要操作该列表时,先要获取

该列表的锁，获取成功方可进行操作，操作结束则释放该锁。

共享内存上定义了两种类型的列表：正在执行作业项列表和作业项执行结果列表，以网元组ID作为列表键值。正在执行作业项列表的作用是实现集群上的网元组限流，通过加入与移出作业项两个操作实现。

加入正在执行作业项列表的过程如下：①根据加入的作业项网元组ID获取需要操作的列表锁；②获取列表长度，即为正在执行的网元组流量，若大于等于作业项网元组限流值，则加入失败；③计算列表中与加入作业项中与网元ID相同的作业项个数，即为网元流量，若大于等于作业项网元限流值，则加入失败；④加入作业项时设置缓存时长等于作业项执行超时时长，释放列表上的锁。

正在执行作业项列表中的数据删除有两种方式，一种是Redis自动按照缓存时长进行删除，另一种是调用移出方法。

作业项执行结果缓存区数据以列表方式存放，并设置存放时长阈值，当缓存数据存放时间超过该阈值，Redis则自动清除。列表上实现加入与获取两种操作，操作前取得对应列表的锁，操作结束后释放。操作前根据作业项ID、网元ID搜索列表中已存在的执行结果，若搜索到则将缓存中的数据作为当前作业项执行结果，否则再次执行作业项并加入新的作业项执行结果到缓存区。

2.1.4 作业项调度执行

作业项调度执行模块作为独立线程在任务网关启动时创建并运行，执行过程如图3所示。

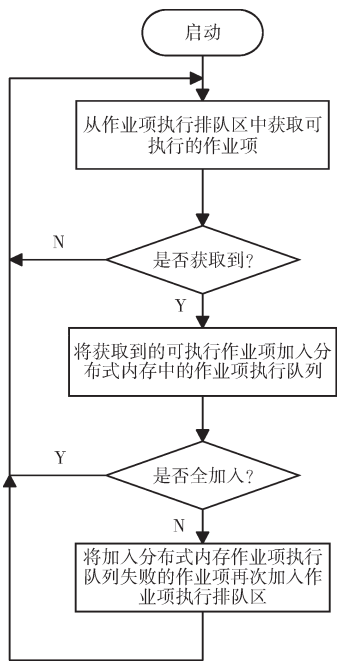

Fig. 3 Task item scheduling execution process

图3 作业项调度执行流程

由图3可见，调度流程主要分为3步：

步骤1：按照作业优先级、作业项等待执行先后顺序以及在单个任务网关内的限流，初步获取可执行的作业项，同时删除等待超时的作业项。

步骤2：将获取到的初步可执行作业项加入分布式共享内存作业项列表执行过程中，完成在任务网关集群上的限流，此时将有部分作业项加入失败。对于被成功加入的作业项，由作业执行模块调用对应的采集服务并取回采集结果报告。

步骤3：将调度执行失败的作业项重新加入作业项执行排队等待区，等待再次被调度执行或超时删除。

2.2 报告逻辑分析

网元巡检数据采集报告是具有特定格式的文本型数据，并且各厂家不同网元类型的报告格式千差万别。通常使用文本处理语言（如：Awk、Perl）进行处理，本文采用自己的专利方法：基于逻辑模板的文本逻辑分析[11]，其优点是不再需要编写复杂的文本分析脚本，只需通过简单的逻辑模板进行配置即可，从而调整分析逻辑或部署新的分析逻辑，更加灵活、快捷。

该专利方法的思想是：将文本数据格式化为行列式数据，创建数据库内存表存放格式化后的文本数据，在此数据库表上编写SQL。分析逻辑分为告警分析逻辑和数据分析逻辑，告警分析逻辑用于查看文本中是否存在某些关键数据，而数据分析逻辑用于获取文本中的若干数据，获取的数据作为作业项指令参数。

现以华为软交换“检查M3UA路由状态”作业项为例进行说明，作业项配置两条指令：“LST M3RT:;”和“DSP M3RT: DEX=$D1$, LSX=$D2$;”，参数“目的实体索引（DEX）”和“链路集索引（LSX）”由第一条指令执行报告通过数据逻辑分析得到，第二条指令应该是指令模板，参数值使用通配符，由该指令模板生成若干条可执行指令。报告的格式化：以换行符获取每一行，以空格获取行内字段。字段名为：F1，F2，F3…

第一条指令的返回报告如下：

```
+++   GL-SoftX      2020-10-10 16:24:56
O&M   #1911562
%%LST M3RT:;%%
RETCODE = 0 操作成功
M3UA路由信息
------------
路由名称 目的实体索引 链路集索引 路由优先级 主/从归属标志
NJ-YFXJ-MS01-M  20  0  0        从归属
NJ-YFXJ-MS01-M-1  20  1  0      从归属
……
(结果个数 = 30)
---   END
```

在这条指令上配置数据逻辑，获取“目的实体索引”和“链路集索引”。逻辑表达式“select F2, F3 from mytable where IsNumber(F2) and IsNumber(F3) IsNumber(F4)”的含义是：获取字段2、3、4为数字所有行中字段2和字段3的数据，其中“IsNumber”是数据库中自定义的函数，采用正值

表达式判断函数的参数是否为数字。

第二条指令的返回报告如下:

+++　GL-SoftX　　2020-10-10 16:55:39

O&M　#1912218

%%DSP M3RT: DEX=20, LSX=0;%%

RETCODE = 44650 由于软交换工作处于去激活模式,且所操作的对象为从归属,所以该对象未安装,操作不能进行

---　END

指令参数:DEX=20,LSX=0,由第一条指令通过数据逻辑分析得到。报告配置告警逻辑分析表达式为“select * from mytable where F1='RETCODE' and F3<>'0'”,含义为:查询RETCODE不等于零的行,若存在则报告异常。

3　结语

针对现有网元巡检系统运行过程中暴露出的问题,由于多个系统使用了如数据总线之类的公共资源,对原有系统进行改造显然无法解决该问题。本文借助微服务架构与分布式共享内存数据库设计统一的网元数据采集平台,对多个系统使用的公共资源进行统一调配,解决了诸如任务量限流等问题。该方式既保证了存量系统可继续使用,从而避免重复建设,节省投资,又新建了基于集群技术的网元自动巡检调度系统,可为新建的5G网元巡检服务。

参考文献:

[1] WU Q, DING M, ZHU Y J. Design of routine test system for switching network element[J]. Jiangsu Communication, 2010(4): 61-63.
吴倩,丁鸣,朱云杰. 交换网元例行测试系统设计[J]. 江苏通信,2010(4): 61-63.

[2] CHENG C, LIANG G Z, QIN J W, et al. Highly available and scalable microservice architecture[M]. Beijing: Publishing House of Electronics Industry, 2019.
程超,梁桂钊,秦金卫,等. 高可用可伸缩微服务架构[M]. 北京:电子工业出版社,2019.

[3] [UK] SAM N. Microservice design[M]. Translated by CUI L Q, ZHANG J. Beijing: People's Posts and Telecommunications Press, 2016.
[英] SAM N. 微服务设计[M]. 崔力强,张骏,译. 北京:人民邮电出版社,2016.

[4] [US]CHRIS R. Microservice architecture design pattern[M]. Translated by YU Y. Beijing: Publishing House of Machinery Industry, 2019.
[美]克里斯·理查森. 微服务架构设计模式[M]. 喻勇,译. 北京:机械工业出版社,2019.

[5] [US]CRAIG W. Spring boot in action[M]. Translated by DING X F. Beijing: People's Posts and Telecommunications Press, 2016.
[美]克雷格·沃斯. Spring Boot实战[M]. 丁雪丰,译. 北京:人民邮电出版社,2016.

[6] LIU W W. Practice of enterprise application development for Spring Boot 2.0[M]. Beijing: Publishing House of Peking University, 2018.
柳伟卫. Spring Boot 2.0 企业级应用开发实战[M]. 北京:北京大学出版社,2018.

[7] [US]JOSIAH L C.Redis in action[M]. Translated by HUANG J H. Beijing: People's Posts and Telecommunications Press, 2015.
[美] JOSIAH L C.Redis 实战[M]. 黄健宏,译. 北京:人民邮电出版社,2015.

[8] HUANG J H. Design and implementation of Redis[M]. Beijing: Publishing House of Machinery Industry, 2014.
黄健宏. Redis设计与实现[M]. 北京:机械工业出版社,2014.

[9] HUANG J H. User manual for Redis[M]. Beijing: Publishing House of Machinery Industry, 2019.
黄健宏. Redis使用手册[M]. 北京:机械工业出版社,2019.

[10] XU X. Distributed task scheduling platform[EB/OL]. https://www.xuxueli.com/xxl-job/.
XU X. 分布式任务调度平台[EB/OL]. https://www.xuxueli.com/xxl-job/.

[11] ZHU Y J. A method and device for logic processing of complex string based on logic template[P]. China, CN201110254900, 2016-03-30.
朱云杰. 一种基于逻辑模板对复杂字符串逻辑处理的方法和装置[P]. 中国,CN201110254900,2016-03-30.

(责任编辑:黄　健)