

分布式资源监控系统设计与实现

刘春雷

中兴通讯股份有限公司 江苏 南京 210012

【摘要】：本文针对本企业内部资源分布广，不具备主动上报特点和现状，设计了一种分布式资源监控架构，并实现系统。实践证明，有效的解决了现实中资产监控和管理的难题。通过对系统架构和原理的分析，可以为其他企业解决同类问题，提供有效的参考。

【关键词】：分布式资源监控；资产管理；消息中间件

1 前言

大多数企业现实中存在的问题：机房资源分布广，设备变动频繁；种类多；仪表昂贵，使用效率难以统计；资源分布难以数字化，难以提供评估和参考；资产位置盘点困难，工作量大，且不及时，查找困难。

为了解决准确、及时盘点资源位置，了解资源的状态和使用情况，数量分布等。可以借助软件工具来帮忙。现有的资源监控工具：Zabbix, Saltstack 等，这些资源监控工具主要采用终端上报方式。Zabbix 适用于终端设备硬件监控，系统监控，应用/网络监控，流量分析，日志监控，API 监控，电脑性能监控，业务监控等。Saltstack 是轻量级的实时通信工具，适用于分布式网络环境，便于扩展，规范有序。但它们都需要在终端被监控设备上安装终端工具，不适合不具备上报特性终端或不适合采用上报方式的终端使用。所以根据企业实际情况，设计和开发了该分布式资源监控系统。

2 系统架构

根据以上存在的实际问题，该系统的设计需要达到可扩展、高可用、高吞吐量、故障容忍、不要求实时但数据不能丢失等目标。所以该系统的架构主要分为：1) 数据收集部分，扫描代理 (ScanAgent) 对控制范围内的资源信息采取定期主动探索的方式进行收集返回信息，同时上报消息。一般都是有多个扫描代理。2) 数据传递部分，主要有消息中间件存储转发数据，对扫描代理上报来的数据进行有序存储，提供缓存，防止丢失，消费删除等。消息中间件主要是解决分布式系统之间的消息传递问题。一个消息中间件对应一个或多个扫描代理。3) 数据处理和存储部分，提供数据消费，计算，分类和保存等功能，同步和交互相关系统的数据等，同时对前端提供各种查询等业务接口。4) 数据输出部分，对用户提供的 web 展示或各种外接服务接口。

系统结构如下图 1 所示：

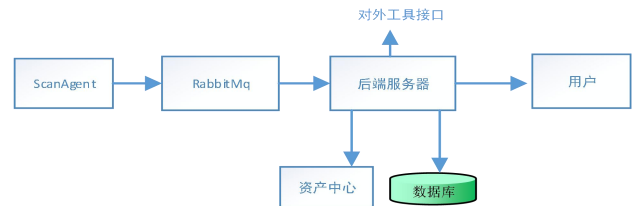


图 1 系统架构图

系统的架构各模块分工协作。除了依赖外部存储如数据库 MySQL 外，不依赖其他系统，各模块全面采用组件化设计和实现。系统每天消息量大，必须多进程或分片处理，均衡负载。

该系统采用 python 语言进行扫描代理和后端开发；前端采用 Angular 的架构开发；数据库使用 MySQL；消息中间件采用的 RabbitMQ，也可以采用其他消息中间件，如 Kafka；后端服务可以采用微服务的方式进行开发，这个能够保证各子功能的独立性，耦合行小。单个子功能微服务出现故障不会影响到整个系统。且便于维护、部署和扩展。整个系统也可以使用其他的语言和工具，但选择时要注意便于部署和维护。

系统模式可以通过消息中间件来拓展，可以使用分布式消息中间件，即多个消息中间件组成的集群。通过集群消费，广播消费等方式来完成广义分布式系统，使系统覆盖的范围更广。

3 工作原理

整个系统采用多个扫描代理分布式部署，信息集中式管理，扩展性和可用性就能够满足，其中采用数据缓存和消费机制，能够满足高吞吐量和防丢失。最终所有的数据都保存在本地数据库，方便做进一步处理。整个系统的主要工作是消息数据的处理，其流程如下图 2 所示：



图 2 数据处理流程图

数据的流程包括：数据建立、数据收集、数据传递和缓存、数据保存、数据分析对比和整理、以及数据对外输出。主要的经历以下几个阶段：

(1) 对每个资产进行编码，外部贴纸质编码，内部存储芯片中烧入新编码，使其一一对应。对内部没有编码或内外不一致的需要重新烧入。同时在其他参数中写入物理位置信息。形成资产编号、物理位置加 IP 和设备结构信息，建立基础有效数据等。

(2) 读取设备信息。为了保证被监控设备及组件的独立性、高性能和便捷性，不能采取安装终端软件的主动上报方式，或者有些设备不具备智能特性，需采用主动向下扫描方式来发现设备及组件。扫描代理是按照 IP 来进行扫描遍历终端。采用 SSH 或 Telnet 远程连接登录设备，通过内部命令显示各种信息，提取其中的有效信息，包括资产编号、物理位置、和设备组件结构信息等，将信息按照一定的格式序列化处理。

对于不同物理位置，设置不同的扫描代理。多个扫描代理进行集中配置和管理，节省人力且有效的保证其正常运行。合理分配扫描的 IP 地址量和区域，保证每个代理的任务适中。

为了保证不影响日常业务测试，采用空闲时间定时扫描，所以收集的设备信息具有一定的延迟，但在合理范围内。将收集的信息序列化处理。

(3) 上传和保存扫描信息。由于多个代理同时工作，数据信息大，为了防止数据丢失和覆盖，需要增加缓存功能，增加了消息中间件 RabbitMq 来接收数据，每个代理在 RabbitMq 上建立一个消息队列，利用 RabbitMq 的存储功能和消费消失机制，起到很好的消息缓存、防溢出和防丢失作用。

(4) 后端微服务及时消费数据，并将数据进行反序列化，进行本地保存到数据库中。采用多进程及时消费中间件 RabbitMq 中多个队列中的数据，可以防止数据大量堆积和溢出队列。对数据按照数据关联规则进行分类分表保存。

(5) 后端和资产中心同步和并盘点数据。后端定期同步资产中心上的数据，并按照资产的编码进行比对，向资产中心开放的接口上传扫描的设备信息，对在线的设备和组件进行位置盘点，节省盘点时间，便于后续查找资产。同时也可以了解资产的使用率等，为后面衍生出的更多的高级应用储备基础数据。

(6) 对数据的应用和展示。后端一方面从消息中间件消费数据，同时和资产中心同步数据，结合两方面数据进行

统计、分析、对比形成多种丰富高级应用，如团队资产分布、各种图形图表等。在 WEB 系统上聚合后输出 HTML 展示，给用户直接提供参考消息；历史日志报表则直接从数据库中获取并展示；提供 XML 数据接口方便外围工具调用。

消息数据在向上传递和消费时使用 AMQP 协议，它工作在应用层，是一套标准协议，该协议基于 TCP 协议，似 HTTP 协议，保证了网络连接的可靠性。同时保证不同平台消息的交互，有很强的兼容性。

4 重点和难点

该系统顺利实施，有几个重点需要妥善解决：

(1) 保证所有的资产都有编码，且唯一，是资产设备标识的最基础的一步。需要设计一套有效的机制来保证现有的资产编码的唯一性，及以后加入的资产的唯一性。

(2) 对于有些资产内部存储芯片没有编码或错误编码，需要重新烧入，这项工作需要所有的资产所有人能统一认识，严格执行烧入策略，严把资产准入机制。资产量越大，分布的范围越广，实施的就越难。

(3) 扫描代理管理和维护。特别是有多个代理时，最好指定专人，集中配置和维护，有问题能及时解决，保证基础数据顺利收集。

(4) 数据消费要及时，中间件虽然有缓存功能，但并不是无限量的，所以后端还是要及时消费，快速处理和存储数据。本文采用多进程消费，对于多进程要妥善的管理和销毁，防止内存泄漏。

(5) 日常工作中设备和组件变动比较频繁，需要及时的更新设备和组件的物理位置信息，保证数据的正确和有效。需要配套的机制或规范来配合实施。

5 系统部署

该系统采用分布式部署。后端服务器和消息中间件 RabbitMq 一般放置在中心机房；扫描代理是按照指定的 IP 或 IP 段进行扫描，可以分机房放置也可以集中存放，有的机房为了网络安全采取防火墙策略或访问控制策略，这样扫描代理最好设置在机房；其他的情况，最好将代理集中存放，便于物理维护，合理分配好扫描的 IP 范围。分配扫描 IP 范围时要关注有效 IP 的量，保证分配的任务量都比较均衡，或按照物理位置来分配，便于后续识别和查找具体物理位置和分割边界。后端服务器集中接收来自所有设备的数据进行保存、分析整理处理，形成各种应用，通过接口对外提供服务。整体部署如下图 3 所示。

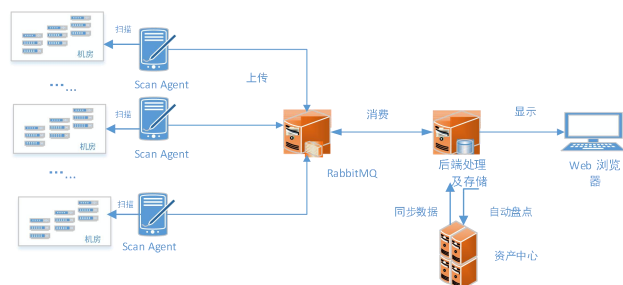


图3 整理部署

6 总结

该系统主要是针对资产监控管理，其优势是不增加设备负荷，能够监控不能上报信息的终端，部署灵活，便于扩展和维护；其劣势是信息不能实时反馈，有一定的延迟性。很适合实时性要求不是很强的需求。该系统的架构简单，对于很多有特殊需求的定制化企业可以进行参考，并在此基础上新增或开发出一些适合企业的个性化的功能。部署该系统，不会影响原网络物理环境，不会影响设备的运行性能和功能。在该系统的架构模式下，可以采用不同开发语言和工具来开发扫描代理和后端服务。经过该系统的实际使用和验证，该模式是一个高效的、可以借鉴的模式。

参考文献：

- [1] 郑伟.分布式云资源监控[Z].第六届“科研信息化联盟”暨“前沿与交叉学科科研信息化发展战略”讨论会,2018.07.26.
- [2] 彭弘思.面向分布式平台的资源监控系统研究与实现[D].华中科技大学,2019.
- [3] 倪炜.分布式消息中间件实践[M].电子工业出版社,2018-9.
- [4] 李永杰.跨平台主机资源监控系统的设计与实现[D].中山大学,2007.05.