

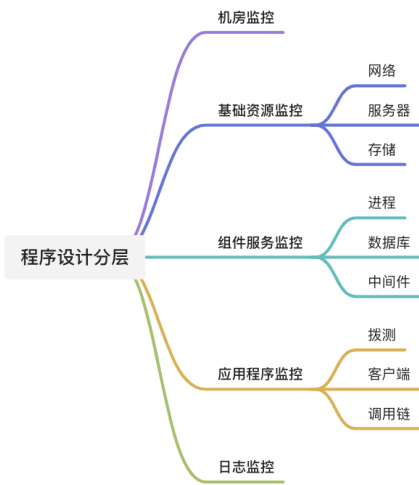
监控产品测试

监控类型产品的业务形态与商业模式

-	-
概念	监控系统一般特指对数据中心的监控，主要针对数据中心内的硬件和软件进行监控和告警。企业的IT架构逐步从传统的物理服务器，迁移到以虚拟机为主导的 IaaS 云，抑或当前流行的容器云PaaS 平台。无论基础架构如何调整，都离不开监控系统的支撑。不仅如此，越来越复杂的数据中心环境，对监控系统提出了越来越高的要求：需要监控不同的对象，例如容器、分布式存储、SDN网络、分布式系统、各种应用程序等，种类繁多；还需要采集和存储大量的监控数据，例如每天数TB数据的采集汇总，以及基于这些监控数据的智能分析、告警及预警等。
商业模式	监控系统其实相当于一个中台系统，供各个云计算产品的部分调用相关服务，然后有一个监控部门单独维护监控服务，监控系统的商业模式通常并不直接面向客户进行计费，而是作为一种关键的技术支撑服务，为内部的业务部门和产品团队提供支持
核心作用	<ul style="list-style-type: none">采集并且展示监控数据，提高其他产品竞争力及时告警，提高产品质量提供数据，辅助进行性能调优，定位故障，提高运维效率

监控产品的监控对象

- 可以将监控分为**机房监控**、**基础资源监控**、**组件服务监控**、**应用程序监控**和**日志监控**。



网络监控

网络监控主要分为以下几个方向。

(1) **网络性能监控 (Network Performance Monitor, NPM)**
主要涉及网络监测、网络实时流量监控 (网络延迟、访问量、成功率等) 和历史数据统计、汇总和历史数据分析等功能。目前，有很多公司提供了 NPM 解决方案，比如天玑、nCompass、SolarWinds、Nagios (已开源) 等。

(2) **网络攻击检查**
主要针对内网或外网的网络攻击如 DDoS 攻击等，通过分析异常流量来确定网络攻击行为。

(3) **设备监控**
主要针对数据中心的多种网络设备进行监控，包括路由器、防火墙和交换机等硬件设备，可以通过 SNMP 等协议收集数据。。

服务器监控

每个程序最终都会在对应的服务器上运行，对服务器的监控一方面是提供服务进程的运行环境信息，另一方面是通过汇总服务器上的监控数据来了解整个数据中心内部的服务器资源使用情况。

服务器监控包括物理服务器主机监控、虚拟机监控和容器监控，需要做到对多种环境的兼容，如下所述。

- 对服务器硬件的兼容。数据中心内的服务器通常来自多个厂商如Dell、华为或者联想等，服务器监控需要获取不同厂商的服务器硬件信息。
- 对操作系统的兼容。为了适应不同软件的需求，在服务器上会安装不同的操作系统如 Windows、Linux，采集软件需要做到跨平台运行以获取对应的指标。
- 对虚拟化环境的兼容。当前，虚拟化已经成为数据中心的标配，可以更加高效便捷地获取计算和存储服务。服务器监控需要兼容各种虚拟化环境如虚拟机 (KVM、VMware、Xen) 及容器 (Docker、rkt) 。

采集监控数据的方式通常分为两种：一种是内置客户端，即在每台机器上都安装采集客户端；另一种是在外部采集，例如在虚拟化环境中可以通过Xen API、VMware Vcenter API 或者 Libvirt 的接口分别获取监控数据。

从操作系统层次来看，采集指标通常有如下几种。

- **CPU**：涉及整体的 CPU 使用量、用户态百分比、内核态百分比、每个 CPU 的使用量、等待队列长度、I/O 等待百分比、CPU 消耗最多的进程、上下文切换次数、缓存命中率等。
- **内存**：涉及内存使用量、剩余量、内存占用最高的进程、交换分区大小、缺页异常数等。
- **网络I/O**：涉及每个网卡的上行流量、下行流量、网络延迟、丢包率等。
- **磁盘I/O**：涉及磁盘的读写速率、IOPS（每秒读写次数）、磁盘用量、读写延迟等。

伴随着云计算的兴起，容器和虚拟机监控成为最近几年所有监控系统的必备功能，我们通常可以通过虚拟化软件提供的监控接口获取其监控数据。由Google 开源的 cAdvisor 可以获取主机上所有容器的性能指标。

开源的服务器监控项目主要有 Prometheus、Zabbix 及 Open-Falcon 等。

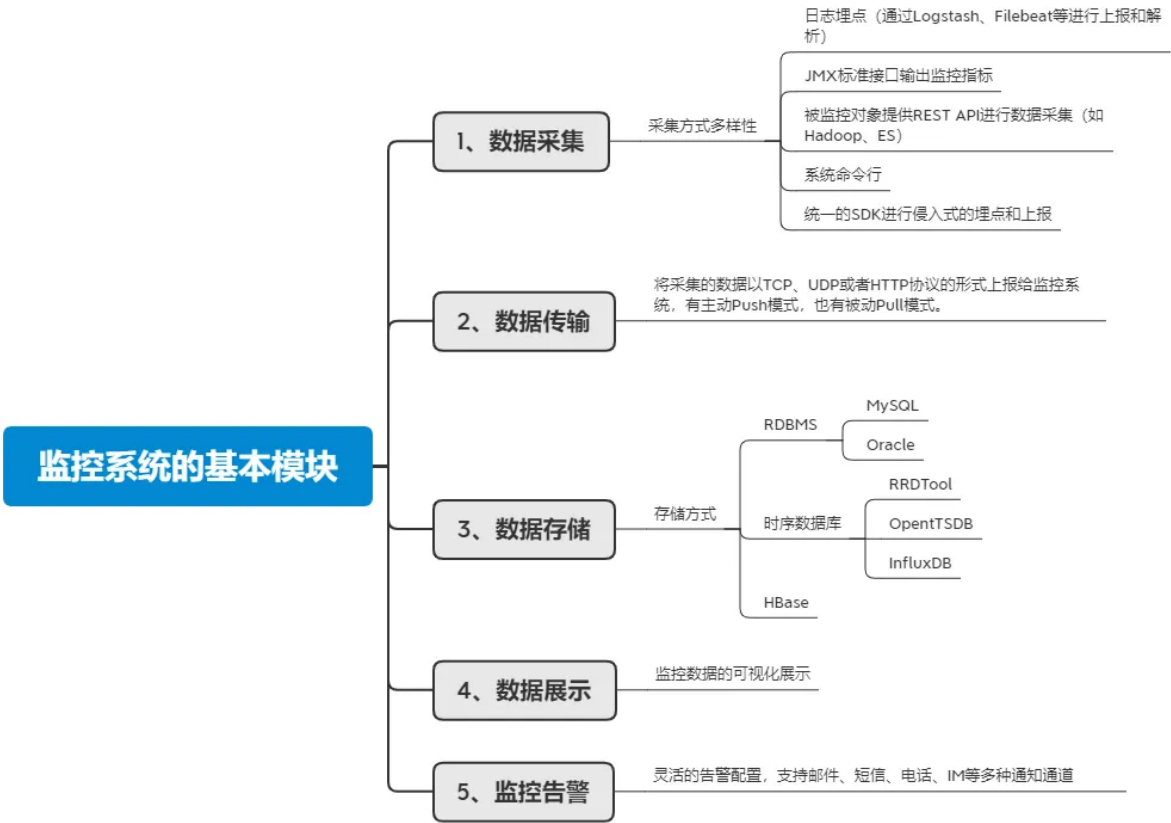
中间件监控

中间件是指系统软件和应用软件之间的连接软件，无论是在分布式系统中还是在单体系统中，中间件都发挥着重要的作用。

常用的中间件主要有以下几类。

- 消息中间件，例如 RabbitMQ、Kafka。
- Web 服务中间件，例如 Tomcat、Jetty、Nginx。
- 缓存中间件，例如 Redis、Memcached。
- 数据库中间件，例如 MySQL、PostgreSQL。

监控系统架构与核心功能



- 所有监控系统的核心都是采集和处理数据。

监控系统通常由指标采集子系统和数据处理子系统这两大子系统组成，对这两大子系统说明如下。

- 指标采集子系统主要负责信息采集、过滤、汇总和存储。
- 数据处理子系统主要负责数据分析、展现、预警、告警动作触发和告警等。

UMon基础功能演示

- 主要功能有五个模块
- 资源监控：能够提供按照产品纬度，快速查询各实例的各个指标的当前数值、历史数值、以及多资源单指标的对比视图，并能够通过设置告警对各个资源快速绑定告警模板，实现阈值触发时的告警通知，以便快速定位故障原因。
- 告警模版：以项目和地域的维度展示所有的告警模版，包含各产品的默认告警模版和用户自建的自定义告警模版
- 告警记录：以项目的维度展示所有的告警记录（同一个资源的同一个指标的告警记录只展示一条，这里做了一个去重的逻辑），提供搜索和自动刷新功能
- 告警屏蔽：提供对告警通知进行屏蔽的功能，支持选择多个指标和全部指标进行屏蔽，生效方式支持持续性生效和周期性生效（PS：告警的通知和告警的产生也就是告警记录是两个功能，告警屏蔽只会屏蔽告警通知并不会组织告警也就是不会影响告警记录的产生）
- 通知人管理：通知组和通知人模块，通知组提供多个通知人划分到一个组并且确定通知方式和语言的能力

UMon指标接入逻辑

指标分为业务方上报指标和采集指标，业务方上报的指标和监控自采集指标，作为UMon的测试，理论上不需要关注业务方上报指标的准确性，这是业务方自己保证的，我们只需要关注指标能够正常显示

- 业务方指标接入UMon流程参考：[【标准】UMON接入](#)
- 监控自身采集指标：比如云主机的监控代理（采集内存，磁盘使用率，进程等指标），对于这种监控侧采集的指标，在指标指标上线或者新增采集指标的时候，需要测试agent采集数据的准确性（<https://docs.ucloud.cn/umon/agent>）

UMon业务测试

需求分类	测试方法	补充与更正
老产品回归（如回归测试云主机，云数据库，共享带宽等）	<div data-bbox="326 879 1427 1003"> <p>1. 创建资源，如云主机（云主机比较特殊，这个产品部分数据是云主机产品方主动上报的，还有一部分数据是监控侧的agent采集的）</p> <p>2. 打开UMon资源监控，选择云主机，显示对应资源以及数据，比如基础资源名称和基础网络、告警模版字段等</p> </div> <div data-bbox="326 1003 1427 1199"> </div> <div data-bbox="326 1199 1427 1247"> <p>3. 点击数据视图，查看详细的指标数据（agent采集的指标，在没有安装agent的时候是没有数据的）</p> </div> <div data-bbox="326 1247 1427 1774"> </div> <div data-bbox="326 1774 1427 1822"> <p>4. 查看对比图标数据显示完全且正常</p> </div>	

UMon平台变更测试	<p>测试平台各功能模块</p> <ul style="list-style-type: none">• 资源监控• 告警模板• 告警记录• 通知人管理• 消息订阅 <p>各个功能模块正常使用，数据正常显示，UI正确</p>	
产品特性测试	<ul style="list-style-type: none">• 数据指标准确性：登录到机器上，确保物理机/虚拟机上的命令指标和图标显示的监控指标误差在一定范围内• 告警场景模拟：模拟各种异常情况，能够正常告警；模拟多个相似告警情况能够正常聚合	
性能测试	<ul style="list-style-type: none">• 数据采集的性能，如数据延迟，数据吞吐量• 系统响应的性能，如告警响应时间• 稳定性测试：故障恢复能力• 数据存储测试：存储性能，查询历史数据• 告警性能：秒级告警	
自动化设计（覆盖使用场景）	举一个最近遇到的需求的例子，其实就可以考虑做自动化提高测试效率：云主机新增几个镜像，需要uma同步做适配支持采集，这个需求的测试中有一个部分是需要新的uma安装在全部uma支持的镜像上面，测试能否采集的数据，这种就可以使用自动化测试的方法，批量创建云主机，并且自动安装agnet，然后校验指标是否有数据。	

参考文档

<https://www.yuque.com/teresa-qrbzf/ch9xlg/rzi894?singleDoc#>

【标准】UMON接入

<https://docs.ucloud.cn/umon/README>

<https://console.ucloud.cn/umon/overview>

Umon-监控