**DMS-agent规格设计**

目录

[1 参考文档 1](#_Toc478542490)

[2 引言 2](#_Toc478542491)

[3 系统要求 2](#_Toc478542492)

[4 总体架构 2](#_Toc478542493)

[5 规格说明 2](#_Toc478542494)

[5.1 硬件信息监控 3](#_Toc478542495)

[5.2 运行时信息监控 3](#_Toc478542496)

[5.3 dms-agent掉线告警 3](#_Toc478542497)

[5.4 数据库 3](#_Toc478542498)

[5.4.1 设备状态码 5](#_Toc478542499)

[5.5 Web规格 5](#_Toc478542500)

[5.5.1 信息展示 5](#_Toc478542501)

[5.5.2 功能规格 5](#_Toc478542502)

[5.6 性能规格 6](#_Toc478542503)

[6 后台设计流程 7](#_Toc478542504)

[6.1 Dms-Agent整体流程 7](#_Toc478542505)

[6.2 Mu处理dms-Agent消息 8](#_Toc478542506)

[6.3 runinfo表的清理 8](#_Toc478542507)

[6.4 设备参数修改 9](#_Toc478542508)

[6.5 Dms-Agent超时判断 10](#_Toc478542509)

[7 web流程 11](#_Toc478542510)

[7.1 设备信息流程 11](#_Toc478542511)

|  |  |  |
| --- | --- | --- |
| **日期** | **作者** | **说明** |
| 2017-3-15 | 肖闯 | 初稿 |
| 2017-3-20 | 肖闯 | 增加设备状态码说明 |
| 2017-3-23 | 肖闯 | 增加阈值字段和设备参数修改流程 |

# 参考文档

Dms-agent主要是针对dms系统功能的一个扩充.故对应的设计方案还是采用原先的设计。设计方案主要参考**<<DMS 系统设计方案.docx>>**

# 引言

在之前的DMS系统中,我们会对恒扬自研的程序进行进程和设备信息的监控。但是在现场环境还有很多设备没有部署自研的程序，针对这类设备，我们目前就无法监控到对应的设备和运行时对应的信息。为了全量的监控所有设备。故需要在原有的DMS功能上进行扩充。同时兼容对第三方设备的信息监控。DMS-agent的目的就是为了支持对现场全量设备信息的一个监控功能。

# 系统要求

Centos 6.6 64位

Redhat6.3 64位

# 总体架构



从该架构可以看出，与之前dms 系统差异主要体现在增加了非自研系统设备的监控代理模块。该文档主要针对非自研系统监控代理模块的设计进行详细说明。

# 规格说明

在对设备监控的功能上主要包含对硬件、运行时等信息的监控，下面主要针对不同的功能列出具体需要监控的规格

## 硬件信息监控

1. Cpu的类型
2. Cpu的数量
3. 磁盘大小
4. 内存大小
5. 系统版本
6. 操作系统位数
7. 网卡名称
8. 网卡速率
9. 网卡状态
10. 网卡ip
11. 网卡MAC

## 运行时信息监控

1. Cpu使用率
2. 内存使用率
3. 磁盘使用率

注：运行时信息可以在配置文件中配置对应的发送间隔事件。目前默认是agent 60秒上报一次.

## dms-agent掉线告警

在dms-agent给mu上报运行时信息时，通常在网络原因导致agent没有及时给mu上报信息。我们不能因为短短的几次没有上报信息，对agent作掉线处理。此时我们需要设置一个超时策略.目前在mu端需要为每一个agent维护一个计数器信息。1分钟出发一次计数器累加操作。当计数器大于5（超过5分钟没有接收到客户端消息），此时我们需要关闭agent对应的连接。并将掉线事件写入告警数据库。

## 进程告警监控

告警监控功能主要是将fude系统环境下所监控的进程进行管理。当监控的进程状态发生变化时，即上报对应的告警信息。初次认为所有的进程为运行状态.

告警信息的内容为：设备名+ip+进程名+状态变化+发生时间(系统时间)

多个进程告警时需要写入多条记录到告警数据库中.

* 支持功能

1. 状态从运行到未运行，上报告警信息
2. 状态从未运行到运行，上报告警信息

* 不支持功能

1. 监控进程状态信息的上报
2. 监控进程的停止
3. 监控进程的启动
4. 监控进程的重启

### 后台规格

后台在程序启动会去读取对应的进程配置文件（进程监控的配置文件通过web前端进行配置下发）。每隔10秒轮询所有监控的进程。并且上报对应的进程状态到mu.mu将接收的进程状态信息保存至数据库。当web有新增的进程监控时，需要实时更新到当前监控的进程列表中。并且将新加入的进程信息状态出入到数据库中。

数据库中的存储形式采用2个字段ip+json数据.json数据用于保存所有的进程状态信息。需要web去单独解析。因为监控的进程数目是可变的，用进程名作为字段名数据不可控.

设计参考:6.8 6.9

## 数据库

在对硬件和运行时的信息进行存储时，该系统采用MYSQL数据库。以下主要对表结构进行说明

* 硬件信息:

|  |  |
| --- | --- |
| 数组库名 | agent |
| 表名 | hostinfo |
| 字段信息 | |
| 字段名称 | 字段说明 |
| ip(varchar(32)) | 设备ip |
| devname(varchar(64)) | 设备名称 |
| state(int) | 设备状态(具体参照状态表) |
| cputype(varchar(256)) | cpu型号信息 |
| cpucount(int) | cpu数量 |
| disksize(varchar(32)) | 磁盘大小 |
| memsize(varchar(32)) | 内存大小 |
| osinfo(varchar(256)) | 操作系统信息 |
| osbit(varchar(32)) | 操作系统位数 |
| ethname(varchar(512)) | 网卡名称 |
| ethkbps(varchar(512)) | 网卡速率 |
| ethstate(varchar(128)) | 网卡状态 |
| ethip(varchar(512)) | 网卡ip |
| ethmac(varchar(512)) | 网卡mac |
| uptime(datetime) | 数据上报时间 |
| cpu\_threshold(int (11)) | Cpu阈值 |
| mem\_threshold(int (11)) | 内存阈值 |
| disk\_threshold(int (11)) | 磁盘阈值 |

注：在一台设备上可能存在多块网卡，此时会存在多个网卡的信息。在数据库中作存储时，采用’,’分割。

* 运行时信息

|  |  |
| --- | --- |
| 数组库名 | agent |
| 表名 | runinfo |
| 字段信息 | |
| 字段名称 | 字段说明 |
| time(datetime) | 存储日期 |
| ip(varchar(32)) | 设备ip |
| cpuuse(int) | cpu使用率 |
| memuse(int) | 内存使用率 |
| diskuse(int) | 磁盘使用率 |
| Uptime(datetime) | 上报时间 |

注：对于该数据库内容的要求，我们需要保存7-30天的数据。在mu的配置文件可以进行天数的配置。另外dms-agent是默认60秒上报一次。在兼容1000台dms-agent的条件下，该数据库一天所产生的数据记录是:24小时\*60分钟\*1000台机器=1440000记录。在达到配置保存的天数后，之前的数据需要定时删除(默认是保留7天的数据)。保证该数据库内容不会无限增长。

* 告警信息

|  |  |
| --- | --- |
| 数组库名 | svmweb |
| 表名 | swwarn |
| 字段信息 | |
| 字段名称 | 字段说明 |
| id(int) | 序号，自动增长 |
| groupname(varchar(40)) | 组名(固定为mu) |
| devname(varchar(40)) | 设备名称 |
| softname(varchar(40)) | 软件名称(固定为dms-agent) |
| warnlevel(int(11)) | 警告级别 |
| occurtime(varchar(40)) | 警告发生时间 |
| warntext (varchar(200)) | 警告消息内容 |

### 设备状态码

设备状态码主要是针对硬件信息表的state字段进行说明.该字段表示当前主机的一个状态

|  |  |
| --- | --- |
| state | 说明 |
| 0 | 主机运行正常 |
| 1 | 设备掉线 |
| 2 | CPU负载 |
| 3 | 内存负载 |
| 4 | 磁盘负载 |
| 5 | Cpu和内存负载 |
| 6 | CPU磁盘负载 |
| 7 | 内存磁盘负载 |
| 8 | CPU、内存和磁盘负载 |

## Web规格

### 信息展示

1. 在对设备硬件信息和运行时的信息进行展示的时候，必须放在同一个页面中。且通过图表的方式进行展示。
2. 在对运行时数据进行展示时，如果超过当前设置的阈值，前端需要通过不同的颜色来区分是否达到告警。
3. 前端设备获取设备名接口,通过树状的形式来展现数据.

### 功能规格

#### 设备操作

设备展示的树结构，需要支持分角色来管理对应的设备。在树的结构中，父节点管理对应的角色名，子节点管理对应的设备名。目前树仅支持1级分类.（方案1：固定5级 方案2：灵活级别）

对于父节点的管理，需要支持对应的增加、修改和删除操作。且删除操作必须是当前的子节点为空才支持此操作。

对于子节点的管理，在设备展示这一块，目前方案有两种:

1. 设备支持新增，用户可以根据已知的设备进行添加操作。但是默认添加的设备为离线状态。且对应的主机信息都没有。因为设备的信息和展示统统来源于后台的上报。对于该方案。还需具体讨论.
2. 在dms系统增加发现菜单。发现的主要功能是主动获取后台上报的设备信息，实时监测是否有新设备上报。当监测到新设备后，需要提示用户并且可供用户选择或创建设备下挂的父节点名称。

发现菜单需要实时读取数据库中的设备数据，并且与设备树中已经展示的过的做对比，在发现页面仅仅展示新发现的数据。用户可以对新发现的设备进行对应角色的划分和添加。

另外对于子节点，已经掉线的设备子节点需要支持删除操作。对于正常在线设备不支持删除

设计流程参照6.6和6.7

#### 设备参数修改

Web前端必须支持对于设备名称的修改。且修改后的值需要同步到数据库和后台配置文件中.

目前系统支持的设备参数的修改项有:

1. 设备名称
2. Cpu告警阈值
3. 内存告警阈值
4. 磁盘告警阈值

#### 告警

1. 对于监控设备的告警，我们需要区别不同的级别，目前我们定义的报警类别有:内存使用率过高、磁盘使用率过高、cpu使用率过高和设备掉线。告警级别如下:

* 设备掉线:EMER
* Cpu使用率过高:warning
* 磁盘使用率过高:alert
* 内存使用率过高:warning

1. 对于设备的告警信息，我们需要单独的在告警页面做展示。并且展示的顺序是以最近的告警优先。告警属于通过mu模块写入对应的数据库（svmweb库的swwarn表）中

告警页面需要定时刷新最新的告警信息（局部刷新 时间间隔（5min））

页面敏感数据的隐藏。

告警取决于状态的变化（进程监控） 以前的告警保持不变

## 性能规格

目前对于单台mu所支持的最大连接数必须大于1000台设备。即一个mu可以最大支持对1000个设备的监控.

# 设计流程

## Dms-Agent整体流程



目前配置文件中主要可配置的项有:

1. 连接mu的ip、端口
2. 设备名称
3. 发送运行时信息的时间间隔
4. Cpu、内存、磁盘阈值

## Mu处理dms-Agent消息



目前mu在接收到dms-agent上报的消息后，通过解析对应的数据，写入到对应的数据库中.硬件信息存放在hostinfo表中.一台设备对应一条记录。多次接收硬件消息后，只对该表作更新操作.同样在接收到运行时信息时，解析后写入到runinfo表中。数据时增量更新。

## runinfo表的清理

由于runinfo存储的运行时信息是增量添加。如果没有删除机制，会导致runinfo表越来越大。所以在设计时，我们需要定时删除。当前我们的程序是删除3天前的数据。即默认保存3天的数据。以前流程是单独一个线程去执行



## 设备参数修改



## Dms-Agent超时判断



## 设备上报



## 设备发现



## 进程配置下发



## 后台进程监控

