**系统设计**

**王绍贤 2017-5-26**

# 产品定义

数据中心基础设施管理（DCIM）平台是对基础设施、资产、容量、日常运维、机房安全性作出统一管理的系统。

  通过监控、分析数据中心基础设施的运行信息（如状态、参数、配置等），帮助数据中心管理者掌握数据中心基础设施运行情况（当前与趋势），管理数据中心资产、基础设施资源（空间、电力、冷量等）、能耗，提高基础设施可用性、资源利用率、管理效率与能效。

# 项目目标

开发数据中心基础设施管理系统。实现以下目标：

1. 实现数据中心基础设施集中监控，全面直观展示数据中心基础设施运行状态，精准定位故障，分析预防故障。
2. 实现数据中心基础设施容量分析，全面分析机房、机架、动力、环境设施利用率。
3. 对机房进行能耗分析，全面分析机房能效，动力设备效率。
4. 实现数据中心资产全生命周期智能管理，给客户提供面向企业内部统一的资产管理平台。
5. 提供机房实物资产自动化盘点工具，并为机房容量管理、配置管理等IT服务管理提供数据接口，协助管理员高效运维机房业务。

# 应用场景

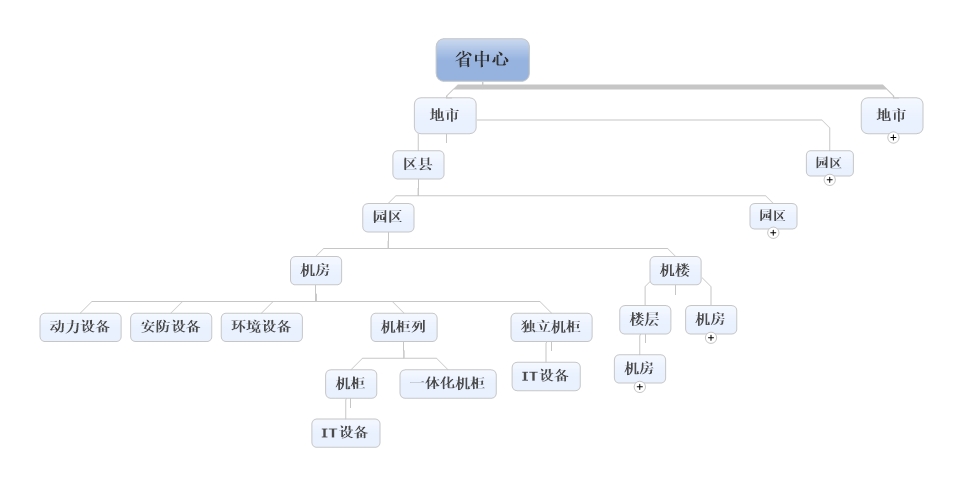
1. 能应用到各种规模IDC机房和运营商机房。
2. 能部署到linux和windows系统，首选linux系统。
3. B/S客户端可以通过不同浏览器在不同平台运行。
4. 移动客户端可以运行于不同移动设备。

# 业务层次约定

为了能更高效的对数据进行分析，需对系统业务数据进行规范化。按照分层结构方式管理数据。

业务层次分为：行政区域、园区、机楼、楼层、机房、机架列机架、各专业设备。

如下图：



# 逻辑架构

将系统分为3个层次：展示层、数据分析层、采集（接口）层。

如下图：



## 展示层

展示层包括B/S客户端和移动客户端。

### B/S客户端

使用nodejs开发应用服务，使用pm2作为nodejs进程管理工具。服务；使用nodejs开发应用服务；UI使用jquery，基础库选用easyui；曲线图等图形库选用Chart.js，仪表板选用easyui-portal。

具体扩展规则参考《**DCIM界面基础框架设计**》。

备份策略：根据需要配置多台服务器作为web服务器。具体冗余及负载均衡策略根据具体硬件资源及网络环境配置，如使用DNS或负载均衡设备。

### 移动客户端

根据现场需求开发不同版本的移动客户端。

采用原生app加网页外挂模式。对于告警、派单等消息数据采用原生app模块，对于告警查询、实时数据等交互式数据采用网页外挂模式。

## 数据分析层

采用osgi模块化分析方式，每个算法一个或多个模块。

**模块间通信策略：**模块间通信统一采用redis通信。

**分析内容包括：**事件、告警、历史记录、利用率、容量分析等。

## 采集（接口）层

DCIM平台非常庞大，其中所管理的业务功能有些已经在现有系统已经实现，完全没有必要重新开发，只需要开发接口与其对接。对于未实现的业务功能则需要开发。为了平台独立功能互不影响，需针对各种接口及业务功能单独开发，其所需的配置及参数都独立开发。

采集的内容分为瞬态信号（如状态、电压值等）、静态信号（如参数、设备型号等）、结构化数据（如从资源管理系统读取到的设备资源配置等）。

结构化数据根据结构自动交由指定分析模块进行分析记录。

# 数据存储策略

数据分为静态参数、业务配置、历史记录、分析结果、采集的数据瞬态值、登录session。

## 静态参数

如节点类型、设备类型、设备型号等在系统部署期已确定的内容，保存到配置文件。

每种参数单独建立文件或文件目录。

## 业务配置

包括基本配置、权限配置、其他具体业务配置等需具备完整性约束的业务内容，存储于配置数据库。

**基本配置：**schama为config；

**权限配置：**schama为portal；

**其他具体业务配置：**根据具体业务单独建立schama。

## 历史记录

包括周期性记录的原始记录值、告警、事件、日志等流水型记录，存储于历史数据库。

采集的历史数据记录模拟量，按照1分钟、10分钟、1小时、1天周期分级存储，系统记录数据时同时生成以上记录；

状态量、字符量、及静态信号根据变化时生成事件记录；

保存周期可以根据保存的服务器资源情况调整，默认设定为瞬态记录1分钟记录保存1天、10分钟记录保存1周、1小时记录保存1月、1天记录保存3年；事件保存1年；告警保存3年；分析结果保存3年。

## 数据分析结果

包括容量分析结果、能耗分析结果等经过预分析的数据，存储于历史数据库。

## 采集的数据瞬态值

存储于集中内存缓存。

## 登录session

存储于集中内存缓存。

# 业务模块划分及设计

具体业务具有具体的模块集合，设计者对具体模块单独建立设计文件，在本文件不做具体说明。

**模块设计文件包括：**

1. **程序模块划分：**界面模块，分析模块等。
2. **存储策略：**涉及的静态参数配置、业务配置、历史数据及记录
3. **内部通信接口：**模块间通信、与界面通信
4. **外部通信接口：**若有对外接口需说明对外接口定义文件。

# 系统部署方案

平台运行需部署以下部件：展示层、数据分析层、采集（接口）层、配置数据库、历史数据库、redis。

为了减少各部件互相干扰，采用各层逻辑上单独部署原则，但可以部署在同一台物理机也可以部署在不同的物理机。

**展示层：**包括应用模块和基础静态文件。使用pm2工具托管。

**数据分析层：**使用osgi模块开发，部署于tomcat。

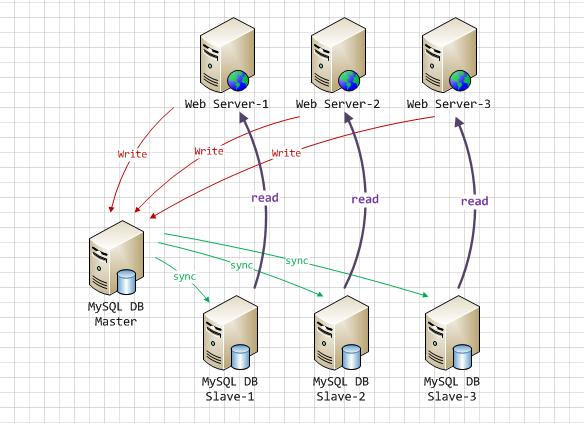
**采集（接口）层：**可以采用osgi模块开发也可以单独开发，各模块单独部署。

# 采用技术

## 配置数据存储

配置数据要求准确，但修改少，数据规模并不大。

选择mysql作为配置存储数据库，建立读写分离数据库集群作为配置数据库备份策略。



逻辑节点属性扩充原则：给逻辑对象建立基本属性表；每种类型对象的扩展属性则使用(key、value)对形式扩充属性，在对象类型定义可扩充的属性（key、type、desc、param[options、range、lenth]）。

## 历史数据和记录存储

历史数据：由于是自动生成的流水型数据，数据价值不大，但要求写入速度快、查询快；

事件：要求准确，数据量不能确定。

告警：要求准确，数据量不能确定。

分析结果及操作记录：要求准确，数据量不能确定。

选择cassandra作为历史数据和记录存储工具。Cassandra是一个混合型的非关系的数据库，是由一堆数据库节点共同构成的一个分布式网络服务，对Cassandra 的一个写操作，会被复制到其他节点上去，对Cassandra的读操作，也会被路由到某个节点上面去读取。对于一个Cassandra群集来说，扩展性能 是比较简单的事情，只管在群集里面添加节点就可以了。

cassandra和其他数据库比较，有几个突出特点：

**模式灵活** ：使用Cassandra，像文档存储，你不必提前解决记录中的字段。你可以在系统运行时随意的添加或移除字段。这是一个惊人的效率提升，特别是在大型部署上。

**真正的可扩展性** ：Cassandra是纯粹意义上的水平扩展。为给集群添加更多容量，可以指向另一台电脑。你不必重启任何进程，改变应用查询，或手动迁移任何数据。

**多数据中心识别** ：你可以调整你的节点布局来避免某一个数据中心起火，一个备用的数据中心将至少有每条记录的完全复制。

## 数据缓存

数据缓存采用redis作为缓存。

## 热力图生成算法及显示

# Todo

## 模块互操作

模块互操作采用redis的消息传输。

## Nodejs应用托管工具

采用pm2托管nodejs应用程序。

pm2 是一个带有负载均衡功能的Node应用的进程管理器，具有以下特性：

* 内建负载均衡（使用Node cluster 集群模块）
* 后台运行
* 具有Ubuntu和CentOS 的启动脚本
* 停止不稳定的进程（避免无限循环）
* 控制台检测
* 提供 HTTP API
* 远程控制和实时的接口API ( Nodejs 模块,允许和PM2进程管理器交互 )

## UI基础框架

Easyui，是一款使用简洁，模块化的UI控件库，拥有丰富的控件；方便简洁的模块化开发，文档<http://www.jeasyui.com/>。

## 图表框架

Echart是一款界面丰富，成熟的开源图表组件，由百度提供。

## 电子地图

选择谷歌离线地图，目前已经实现地图需要的基本功能，需根据项目下载对应的离线地图瓦片资源。

## 3D（2D）组态界面

需考虑控件使用要求，如硬件资源要求，软件平台限制等。

方案1、自己开发

方案2、采购控件（twaver，三源）

Twaver：可以运行于不同浏览器不同平台，纯B/S结构，但对内存及显卡有要求，否则运行较慢；只有控件，需二次开发，且开发工作量较大。

三源：仅能适应于windows，需安装插件，且对浏览器版本有限制，对硬件要求较高，否则不能运行。

# 系统内部组件监控策略

## UI Web服务

# Todo

## 配置数据库

# Todo

## 历史数据库

# Todo

## redis中间件

# Todo

## 各分析模块

# Todo

## 各采集模块

# Todo

# 需研究的技术

## Mysql读写分离技术部署配置及读写代码

## Cassandra部署配置及读写代码

## Redis集群部署配置方法及读写代码

## Osgi模块技术

## 数据分析策略

## B/S界面框架

## 3D组态控件

## 数据采集模块

## 动环智能设备采集

1. Snmp采集
2. Ipmi采集
3. 应用程序监控（java程序、oracle、sqlserver、各种web应用等）
4. 操作系统日志分析
5. 网络设备日志分析