虎符网管系统设计概述

修改历史

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 | 2014/07/29 |  | 初始版本 |

目录

[目录 2](#_Toc394528441)

[Overview 4](#_Toc394528442)

[1. 系统架构概述 4](#_Toc394528443)

[1.1. All-In-One网管 4](#_Toc394528444)

[1.1.1. 网管软件流程图 4](#_Toc394528445)

[1.1.2. 软件构成 4](#_Toc394528446)

[1.1.3. 运行环境 5](#_Toc394528447)

[1.1.4. 产品发布 5](#_Toc394528448)

[1.2. 分布式网管 5](#_Toc394528449)

[1.2.1. 网管软件流程图 5](#_Toc394528450)

[2. 系统功能模块概述 6](#_Toc394528451)

[2.1. WEB 服务 6](#_Toc394528452)

[2.1.1. MVC框架 6](#_Toc394528453)

[2.1.2. 用户和域管理 6](#_Toc394528454)

[2.1.3. 配置管理 7](#_Toc394528455)

[2.1.4. 设备监控 7](#_Toc394528456)

[2.1.5. 网管全局设置和监控 7](#_Toc394528457)

[2.1.6. 设备远程调试工具 7](#_Toc394528458)

[2.2. 后台模块 7](#_Toc394528459)

[2.2.1. 消息调度模块 8](#_Toc394528460)

[2.2.2. 系统缓存模块 8](#_Toc394528461)

[2.2.3. 后台定时任务模块 8](#_Toc394528462)

[2.2.4. Capwap模块 8](#_Toc394528463)

[2.2.5. 设备的配置下载和软件升级模块 8](#_Toc394528464)

[2.2.6. 系统日志模块 9](#_Toc394528465)

[2.2.7. Monitor后台模块 9](#_Toc394528466)

[2.2.8. 告警模块 9](#_Toc394528467)

[2.2.9. 系统备份升级 9](#_Toc394528468)

[2.3. CentOS平台 9](#_Toc394528469)

[2.3.1. 安装部署及系统升级 9](#_Toc394528470)

[2.3.2. Shell脚本 10](#_Toc394528471)

# Overview

网管理系统是一个软硬件结合以软件为主的分布式网络应用系统，其目的是管理网络，使网络高效正常运行。一个好的网管系统可以快速管理网络设备、简化用户配置、灵活的定制用户个性化服务、监控设备状态和告警、快速排除设备故障。

# 系统架构概述

## All-In-One网管

### 网管软件流程图



图表 1

### 软件构成

本设计包含4部分:

* Capwap Server: 负责与AP等嵌入式设备通讯，他只负责Capwap协议的传输，对AP的数据无任何业务处理，直接将Capwap Event送往BE，主要起到传输的作用。
* BE(Backend-Engine)：负责后台业务的处理。包含的模块有通信框架，配置模块，监控模块，告警模块，AP版本控制模块，Cache模块，Capwap模块，周期任务(scheduler) 模块，日志模块, 数据库访问模块, troubleshoot模块等。
* FE(Front-Engine)：负责前台业务的处理，实现MVC框架。包含的模块有通信模块，配置模块，AP监控模块，AP版本控制模块，告警模块，用户管理模块，系统模块，数据库访问层和troubleshoot模块。
* UI(User -Interface)：用户的UI设计.

### 运行环境

* 网管运行在CentOS系统上。
* 数据库用Mysql/Postgresql，数据库中间件采用Hibernate。
* Web Server 可以用Tomcat。
* FE和BE运行在Java 环境上。

### 产品发布

1. 网管系统要实现自动化部署，可以将软件运行在CentOS上。
2. 可以将网管系统和CentOS一起打包成一个VMware, 以实现网管系统方便灵活的部署。

## 分布式网管

靠单纯All-In-One系统架构的处理能力是有限的。分布式网管系统可以通过增加物力节点的方式来达到管理更多设备的目的。把系统的FE, BE, DB, Capwap Server 从单台系统分离开来部署, 每一项都可以添加或移出节点，随着设备数量和用户群体的不断增加，当某项遇到性能瓶颈的时候就可以通过增加节点来提升性能。

### 网管软件流程图



图表 2

将FE 和BE分离部署，同时引入Message Queue, Memory Cache 和Scheduler节点，这样FE和BE就可以实现无状态运行。在实际应用中可以通过扩展FE和BE来提升性能。

前台通过Load balance 将请求发送到任意一个FE，FE通过Message Queue 和任意BE通讯。 BE 如果需要和Capwap通讯，通过Capwap Routing路由到相应的Capwap Server。

Memory Cache把Cache模块从BE分离出来，其中记录了一些需要被共享的状态。 Schedule 模块也需要单独部署（整个分布式系统中只能部署一台Schedule Server，因为定时任务是需要单线程运行的）。

# 系统功能模块概述

## WEB 服务

### MVC框架



图表 3

1. **JSP**：用户看到的HTML页面，把页面数据提交给WEB服务器。
2. **Controller**：接收JSP提交过来的数据，但不做业务逻辑处理，所有的业务逻辑处理有后面的Service来完成，他只负责把JSP 传递过来的离散的数据封装成数据模型。
3. **Service**：所有的业务校验和逻辑处理都在这里完成，他可以间接地操作数据库但不能直接访问，必须通过后面到DAO来访问；同时Service也可以访问后台模块，但不能直接调用，必须通过同步或异步事件来访问。
4. **BE**：管理着后台所有的服务模块。
5. **Dao**：Service提供访问数据库的接口（比如Profile的创建、修改和删除），他直接操作Hibernate，封装了所有数据库操作的细节。
6. **Hibernate**：是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装，使得Java程序员可以随心所欲的使用对象编程思维来操纵数据库。他的引入可以轻松实现不同数据库的切换。
7. **DB**：Mysql/Postgresql数据库。

### 用户和域管理

1. **域管理：**一个公司底下的所有设备，我们把它们虚拟为一个管理域，虽然一台网管系统同时管理着不同公司的不同设备，但是公司的网管通过域账户登录网管系统后只能看到和管理自己的设备。
2. **用户：**用户可以分为系统级用户和域级用户。
3. **系统级用户：**作为超级用户一般只发放给系统维护人员，可以看到并管理网管系统上的所有设备。
4. **域用户：**看到和管理的设备只局限于他所在的域。

### 配置管理

设备的配置信息可以分为两类：共性配置和个性配置。

* **共性配置**：以profile的方式存在，不同设备间无差异或者差异极小，如：SSID, IP-POLICY, DNS 这些对大部分设备来说他们都采用相同的配置策略。
* **个性配置**：由于硬件的差异或业务的限制导致每个设备的配置都不尽相同，如：天线的数量，天线的模式，PPPOE的用户名密码等。

基于上面的共性和个性两类配置需要抽取其中的共性部分，让它以profile的形式存在以减少用户的配置量，根据抽取方式的不同可以分成以下三类：

1. **所有设备的共性：**如SSID, IP-POLICY, DNS
2. **同一类设备的共性：**可以按照设备来分组，让同一类设备选择同一个profile。
3. **纯粹的个性：**必须每个AP单独配置如PPPOE.

### 设备监控

网管可以实时监控设备的当前状态或之前一段时间的历史记录，根据监控数据的来源方式可以分为三类：

1. 数据通过实时查询的方式获取，当需要查看某个设备的某个状态时，可以通过命令行或报文的形式向设备发出查询请求。这种方式对网络的要求较高，默认情况下不建议采用这种方式。
2. 设备的某些状态发生变化时主动上报、设备定时上报、网管发起定时查询，这里三种方式虽然不能百分百的反应设备的当前状态但也是极为接近的。这种方式的优点也很明显系统消耗低、页面响应快，所以我们建议采用这种方式。
3. 有些时候我们需要的不是一个时间点的数据而是一些统计数据，这就要系统对一段时间内的数据做汇总分析。

### 网管全局设置和监控

* **全局设置：**这里的全局设置可以分为两类

1. 系统参数：系统级别的参数，这些参数只有系统管理账号登陆后才能够修改。
2. 功能开关：更多的是和一些业务相关的，域账号就有修改的权限。比如有一个较冷门的功能，如果直接暴露给客户会增加其配置的复杂度，这时通过功能开关控制其在页面上的可见度是一个较好的解决方案。

* **网管系统监控：**这里的网管监控有别于之前对设备的监控，这里只指针对系统本身，比如监控当前网管的CPU、内存的使用情况，监控网管后台线程的状态和队列的饱和程度。

### 设备远程调试工具

当远在千里之外的设备出现故障时，维护人员往往鞭长莫及，这时开发一个人和设备交互的中间平台就势在必行了。调试工具相当于一个命令行窗口，维护人员以命令行的方式向设备发送指令，网管接收到设备的反馈后把结果显示在页面上。

## 后台模块

### 消息调度模块

消息模块负责消息的接受和发送，消息模块的引入主要是为了避免模块间的直接调用，让模块彼此之间解耦，有利于将来分布式系统的横向扩展，为将来像分布式迁移做准备。

按照对消息结果处理的不同，可以把消息分成两类：

**异步消息：**不需要等待消息的处理结果，比如Capwap报文的处理，只需要把报文的内容放到DB里，如果有错误发生只需要记录日志就可以，而不需要主动通知消息的来源方。

**同步消息：**和异步消息相反，消息发送方迫切的想得到消息的处理结果。比如用户从网管前台发起对某个设备的状态查询，用户在前台同步等待响应。

按照对消息发送方式的不同，可以把消息分成两类：

**广播消息**：一个消息对应多个消息处理模块，比如设备上报一个Discovery事件，很多处理模块都需要对这个事件做出相应的处理。

**定向消息**：消息对应唯一的处理模块，比如用户发出一个配置下载事件，只有配置模块会对此做出响应，这种定向消息可以在消息中指定目的地，调度模块会做点对点的调度。

消息调度模块的引入对开发规范有一定的要求，最重要的是禁止模块间的直接调用，模块间的所有调用全部通过消息的方式实现。

### 系统缓存模块

系统缓存模块也是为之后的分布式框架迁移做准备的，分布式框架不允许个别服务记录一些全局的状态，那些全局的状态要对所有的服务可见，而不是自己对自己可见。

系统级别的缓存要集中记录到缓存模块中，而不是分散在各处。集中缓存会大大降低向分布式迁移的工作量，这点也是开发规范之一。

### 后台定时任务模块

网管系统除了消息模块里的线程外，还有很多定时任务。所有的定时任务统一放到定时模块里实现统一启动，统一维护。

### Capwap模块

此模块负责与Capwap server 通信，负责处理来自设备的报文。

* **接收发送：**Capwap模块，接收Capwap Server 转发过来的AP事件，同样也把网管需要发送给AP的事件发送给Capwap Server。同时包括事件的分发，根据事件的类型将事件送往相应的模块进行业务处理。
* **Capwap事件的解包和组包：**该模块还会完成事件的解包和组包。事件由AP上来的时候，模块会负责解包，每种包根据类型和包结构，解析成相应的模型。同时也会将要发送给AP的数据，拼装组成数据包，发送给AP。

### 设备的配置下载和软件升级模块

* **配置的下发**:读取DB 中有关设备的配置封装成XML或JSON 格式，通过Capwap发送到设备端。
* **设备的升级**:所有设备版本的维护，当设备连接网管自动检测是否需要升级软件，弹出告警提示用户，经用户确认后自动选择相应的软件包发送给设备。
* **文件的下发**：以文件的形式下发那些用户定制的广告页面。

### 系统日志模块

日志模块负责记录软件打印出来的日志，这个模块支持日志格式的定制，日志的级别，日志的分文件和日志的rotate。该模块属于基础框架模块，有了它我们可以查看我们代码中记录的信息，一旦软件有什么问题，会从日志文件中找到线索。支持动态的开启模式，软件运行过程中动态的设置log的级别。

### Monitor后台模块

Monitor模块主要负责对设备的监控，AP将各种状态信息发送出来，经由Capwap到达Monitor的模块。模块负责业务逻辑的处理，将设备信息和设备状态处理完后存入数据库，以备前端显示作用。设备上有什么异常，可以通过该模块直接的展示给用户，如端口down了，或者AP和网管失去联系。此模块在以往的网管中，比较类似网元信息的部分。

### 告警模块

此模块专门处理告警信息，如果产生新的告警除了业务逻辑处理，还可以通过各种方式通知指定的人，比如说手机或者邮件等方式。当收到clean告警时，进入业务逻辑处理，将软件正常话。这部分是用户比较关心的，告警可以定制。

### 系统备份升级

软件本身需要升级，数据备份和数据恢复。当admin对系统进行升级的时候，软件处于维护模式，客户暂时不能访问，当升级结束再开放给用户，比如说从1.0升级到2.0版本，同时用户的数据不能丢失。

备份和恢复都包括2部分。一个是admin的行为，一个是用户的行为。Admin可以对网管整体数据进行备份和恢复。用户可以只针对自己的数据进行备份和恢复。

## CentOS平台

### 安装部署及系统升级

网管系统需要运行在CentOS平台，除了系统必要的软件，网管软件也需要应用软件的支撑。比如说数据库，JRE等，有时候为了更高的性能也需要修改一些，系统和应用软件的配置文件。这样如果每次都手工操作这些，很容易出错也很枯燥。

系统自动化部署可以自动将运行所需要的软件和配置，安装部署到目标机。如果想省略CentOS安装，自动部署可以将网管产品和CentOS打包成一个VMware, 里面包含CentOS和网管所需的软件。

系统升级，根据网管的需求，基础软件有时候需要版本的更新。如JRE和DB的更新。这时系统就必须利用升级功能，如果操作系统本身有什么重大升级也需要该模块。

### Shell脚本

因为我们运行在CentOS的上面，有些工具可以用shell脚本完成。例如数据库的建库和环境准备，还有数据库层的数据备份和恢复。这对线上产品来说非常重要。Shell脚本通常完成一些系统的功能，这样不仅实现迅速，效率也很高。