**一.系统简介**

我们将在不改变OpenStack核心程序的前提下，外加我们的系统。该系统监测云端计算机（包括物理机和虚拟机实例）的运行情况（包括cpu、内存、硬盘等的基本使用情况）等。

（1）云端弹性资源监控系统模块包括：

|  |  |  |  |
| --- | --- | --- | --- |
| **模块** | **运行位置** | **用途** | **简称** |
| **节点程序** | 在每台物理机上 | 1. 采集物理节点的运行情况； | NCMonitorSvr |
| **虚拟机监控程序** | 在每台计算节点上 | 1. 采集各虚拟实例的运行情况； 2. 管理各虚拟实例的关、启。 | VMMonitorSvr |
| **总管理程序** | 在任意某台计算机上 | 1. 宏观管理节点程序和虚拟机监控程序； 2. 作为网页的触发服务程序 | CCMonitorSvr |
| **用户网页服务程序** | 在任意某台计算机上 | 网页服务 | WebSvr |
| **数据库** | 在任意某台计算机上 | 存储各计算机的运行情况 | DB |

（2）云端弹性资源监控系统各模块各运作关系如图：

**二．实现详解**

目前主要完成了红圈部分，实现的技术框图如下：

事实上，本项目用不着这么庞大的集群架构。我当时搭建网站，顺便把负载均衡搭上去了。

**1.后台服务**

用户在web端进行交互，从web发出请求，在（proxy+loadbalance）处进行分发，而且是按照类型分发：将页面的请求分发到httpsvr，将用户动作的请求分发到cgi，这样的好处是：将页面请求与用户请求的后台完全隔离，隔离之后，以后的弹性伸缩就很方便了。

Httpsvr依然是用nginx来搭建。Nginx可以做负载均衡，也可以做web服务，而且其性能比apache、djago等高很多。Cgi则是自己用python编写，它是怎个系统的核心所在，它的设计复杂一些，等完成之后再详细介绍。

**2.页面设计**

我请刘邓帮我写了一个简单的网页（主要是一些html、javascript、css文件）来给我测试。之所以请他帮忙，因为我之前没做过网页，写一个测试网页我可能要花好几天，而他只要几个小时。我设计的网页界面如下：

主要分四个部分：

A：我们实验室的信息，包括实验室logo、实验室云logo。以后可以在背景设一张gif图片，使其显示动画。

B：小标题，服务器监控、虚拟机监控、集群工具等，B中选中某一项，C和D就相应地切换页面。

C：此图是服务器监控的界面，图片显示云端各计算机的运作情况，用各种颜色（蓝、红、黑白）表示计算机的运行状态。

D：选中某台计算机时，对应该计算机的信息、操作等。显示存储、内存等使用情况的实时、历史条状图和曲线图。

网页的动态可以通过编写javascript实现，网页的难点是整合通信协议。网页的美化工作不是主要的，留到后面优化。

**3.需改进之处**

（1）Cgi的功能可能需要分离。上图设计cgi=web后台服务+集群后台管理，应该将web后台服务与集群后台管理隔离，前者仍命名为cgi，后者则是GetNodeInfo-CC。

（2）后台的通信可能需要改为http。但是http可能仅仅方便了代码的编写与维护，通信数据的冗余可能会影响后台的