单位代码 10006

学 号 10231007

分类号 TP311

****

毕业设计(论文)

国家高性能计算环境运行支持平台前端的设计和实现

|  |  |
| --- | --- |
| 院（系）名称 | 高等工程学院 |
| 专业名称 | 计算机科学与技术 |
| 学生姓名 | 黄绍晗 |
| 指导教师 | 钱德沛 |

2014年6月

国家高性能计算环境运行支持平台前端的设计和实现析 黄绍晗 北京航空航天大学

北京航空航天大学

**本科毕业设计（论文）任务书**

Ⅰ、毕业设计（论文）题目：

国家高性能计算环境运行支持平台前端的设计和实现

Ⅱ、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

使用原始数据：

国家高性能计算任务书，RESTful API手册-用户管理API接口

设计技术要求：

用户界面使用Bootstrap和jQuery开发；后台数据采用RESTful接口调用；编码格式规范，代码可读性好，有适当的注释

Ⅲ、毕业设计（论文）工作内容：

完成国家高性能计算环境运行支持平台前端的设计与实现，其中包括用户登录，平台权限控制，平台用户管理，环境运行展示等。用户界面使用Bootstrap和jQuery开发，实现样式丰富的展示前端。服务器端采用Java，使用Apache HttpClient进行RESTful接口的调用。

Ⅳ、主要参考资料：

[1] 国家高性能计算任务书

[2] RESTful API手册-用户管理API接口

[3] Nagios CGI接口（集群实时、运维服务、集群统计）使用说明

[4] jQuery技术内幕：深入解析jQuery架构设计与实现原理 机械工业出版社

[5] jQuery Javascript与CSS开发入门经典 清华大学出版社

高等工程 学院（系） 计算机科学与技术 专业类 102311 班

学生 黄绍晗

毕业设计（论文）时间： 2014 年 3 月 \* 日至 2014 年 6月 \* 日

答辩时间： 2014 年 6 月 \* 日

成 绩：

指导教师： 钱德沛

兼职教师或答疑教师（并指出所负责部分）：

系（教研室） 主任（签字）：

本人声明

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者：黄绍晗

签字：

时间：2014年6月

国家高性能计算环境运行支持平台前端的设计和实现

学 生：黄绍晗

指导教师：钱德沛

摘 要

随着我国高性能计算机研制水平和高性能计算中心规模的不断发展，建立国家级高性能计算基础设施并投入良性运行已成为迫切需要解决的问题。

运行支持平台作为高性能计算基础设施的一部分，将对整个高性能计算环境的计算资源、基础设施服务、用户进行监控和管理提供帮助与服务。运行支持平台是高性能计算环境在实际生产性运维期间的环境监控和管理工具，提供对环境运维决策制定的数据支持。

本课题完成了运行支持平台前端的设计和实现，其中包括用户登录，平台权限控制，平台用户管理，环境运行展示等。整体架构采用分层的结构，各层之间相互独立、保持接口一致更加易于维护和实现。用户界面使用Bootstrap 2.0和jQuery进行开发，实现了样式丰富的用户界面和简单快捷的用户交互。用户管理权限控制采用与运算的方式使得权限计算简单、快速。设计完成了用户审批流程，使用Ajax技术遵循用户审批流程完成了用户管理模块，浏览器与 Web 服务器之间使用异步数据传输，页面只需要部分刷新，减少网页从服务器请求的信息。后台数据的获取使用了RESTful接口和Nagios/CGI接口。

关键词： Bootstrap、jQuery、RESTful、Ajax

Frontend design and implementation of national high performance computing environments running support platform

Author: HUANG Shaohan

Tutor: QIAN Depei

Abstract

With the rapid development of China's high-performance computers and high- performance computing scale, it’s an urgent problem that how to improve the establishment of high-performance computing infrastructure.

Running support platform as part of the infrastructure of high performance computing, it supports help and service for computing resources, infrastructure services and user manager. Running support platform is a very efficient environmental monitoring and management tools during the actual operation and maintenance of productive.

Frontend design and implementation of running support platform includes user login, platform access control, user management platform, operating environment and exhibition. Overall architecture uses a hierarchical structure that is independent between the layers. User interface development Bootstrap 2.0 and jQuery to achieve a style rich user interface and simple user interaction. Design complete user approval process, the use of Ajax technology to complete the approval process for the user to follow the user management module, asynchronous data transfer between the browser and the Web using the server, the page refreshes only partially reduce the requested information from the web server. Background data acquisition using the RESTful interface and Nagios / CGI interface.

**Key words**：Bootstrap、jQuery、RESTful、Ajax

**目 录**

[1 绪论](#_Toc389587575)

[1.1 课题背景及目的 1](#_Toc389587576)

[1.2 国内外研究状况 1](#_Toc389587577)

[1.3 课题研究内容 2](#_Toc389587578)

[1.4 论文构成 3](#_Toc389587579)

[2 相关技术 4](#_Toc389587580)

[2.1 Bootstrap 4](#_Toc389587581)

[2.2 JQuery 5](#_Toc389587582)

[2.3 RESTful 6](#_Toc389587583)

[2.4 SSL/HTTPS 7](#_Toc389587584)

[3 运行支持平台前端的整体结构 8](#_Toc389587585)

[4 支持平台权限管理 10](#_Toc389587586)

[4.1 权限管理设计 10](#_Toc389587587)

[4.2 用户管理权限控制的算法实现 12](#_Toc389587588)

[5 环境运行 25](#_Toc389587589)

[5.1 Nagios/CGI 25](#_Toc389587590)

[5.2 环境运行中的用户交互 27](#_Toc389587591)

[6 用户管理 15](#_Toc389587592)

[6.1 用户审批流程 15](#_Toc389587593)

[6.2 用户管理模块系统框架 17](#_Toc389587594)

[6.3 用户管理模块的页面设计 19](#_Toc389587595)

[6.3.1 用户管理首页设计 19](#_Toc389587596)

[6.3.2 用户管理审批页面设计 22](#_Toc389587597)

[结论…… ……………………………………………………………………………………...32](#_Toc389587598)

[致谢…… ……………………………………………………………………………………...34](#_Toc389587599)

[参考文献 36](#_Toc389587600)

# 绪论

## 课题背景及目的

高性能计算能力是国家综合实力和创新能力的重要体现。随着我国高性能计算机研制水平和高性能计算中心规模的不断发展，建立国家级高性能计算基础设施并投入良性运行已成为迫切需要解决的问题。国家级高性能计算基础设施的建立有利于充分有效的组织国内优秀的计算资源，提升资源整体利用情况，避免了不必要的浪费；同时有利于为科研人员提供优质、不间断的计算服务，提高科研人员的科研效率，加速其成果产生。

运行支持平台将联合环境中所有超级计算中心的技术支持力量，面向用户形成一个统一的、优质的、可快速响应的专业技术服务队伍。运行支持平台将对用户的问题作出快速响应，同时要将问题准确的转发给相关负责人员，保证问题能够得到真实有效的解决。

本课题为国家高性能计算环境运行支持平台前端的设计与实现，运行支持平台将对整个高性能计算环境的计算资源、基础设施服务、用户进行监控和管理提供帮助与服务，是高性能计算环境在实际生产性运维期间的环境监控和管理工具，提供对环境运维决策制定的数据支持。本课题作为环境运行支持平台前端的设计与实现，主要是使用Nagios/CGI和SCE/RESTful接口作为数据调用接口，设计实现用户界面的展示和web应用服务。

运行支持平台的目的是主要研究资源监控和运行管理技术、统一认证与系统安全技术，构建环境的运行支持平台，为国家高性能计算基础服务的稳定运行和长期发展提供支持，最大程度的降低运维成本。而本课题作为运行支持平台的前端，可以完成更好的运行支持平台与用户的交互，更加完善、更加人性化的数据展示和数据可视化。

## 国内外研究状况

中国国家高性能计算基础设施的建立要追溯到20世纪九十年代后期。1998年，国家863计划“智能计算机系统”主题将研究重点从研制单台高性能计算机转向从技术上支持我国高性能计算基础设施的构建。这个项目支持建立了由5个高性能计算中心构成的国家高性能计算环境，形成了我国高性能计算环境的雏形。

2002年到2005年，863计划实施了“高性能计算机及其核心软件”重大专项，在网格环境、网格软件和网格应用等方面完成了一批重要的研究课题；基于自主研制的网格系统软件，将分布的8个高性能计算中心连接起来，向全国的科学研究用户和行业用户提供开放共享的高性能计算服务，为我国的科学研究和信息化建设提供了新型环境和平台[1]。

2006年到2011年，国家863计划实施了“高效能计算机及网格服务环境”重大项目，其战略目标是研制千万亿次高效能计算机系统，构建中国国家网格服务环境。在这个重大项目支持下，建成了14个结点组成的国家级高性能计算服务环境，聚合计算能力超过3000万亿次以上，存储能力超过15PB，部署了450多个软件与服务，支持了1100多项国家与地方科技项目[2]。

经过十余年的不懈努力，国家高性能计算环境积累了丰富的基础软件建设经验和环境运行管理经验，在实践中发挥了重要的作用，但仍存在着亟需解决的问题，比如环境建设缺乏规划、实际应用推广还有待普及、既懂计算又具有应用领域知识的跨学科人才严重不足等等[3]。

美国是世界上最重视高性能计算机的国家，同时美国也是在高性能计算基础环境上投入最多的国家。在美国有很多高性能计算的重大计划和项目，比如：美国能源部的加速战略计算ASCI计划，ASCI计划的目标是构造具有100万亿次计算能力的超级计算机；白宫直属的HECC（High-End Computing and Computations）计划，对高性能计算的关键技术进行研发，并构建高性能基础设施；Petaflops计划开发构造千万亿次级系统的技术。

欧洲的强项则主要体现在高性能计算机的应用方面。欧盟投入巨资建设欧洲数据网格Europena DataGrid(EDG)，面向卫星观测和数字地球、生物和气象、高能物理等应用联合开展研究并构建计算和数据网格。欧盟还资助了EuroGrid的研究和技术开发计划，目标是使用高性能计算为科学和工业群体提供一个广域分布的信息处理环境。

## 课题研究内容

本课题主要完成了运行支持平台前端的设计与实现，其中包括用户登录，平台权限控制，平台用户管理，环境运行展示等。用户界面使用Bootstrap和jQuery进行开发，实现样式丰富的用户界面和简单快捷的用户交互。服务器端使用Java Servlet处理Http请求，与数据的交互主要使用Apache HttpClient进行RESTful接口的调用以及CGI接口的调用。

运行支持平台前端从用户界面到数据调用都采用分层结构，每一层之间相互独立，更加易于维护和实现。

## 论文构成

本文的内容组织如下：

第一章 绪论，介绍了课题背景及其目的，概述了运行支持平台对于高性能计算环境的重要性，同时简单介绍了国内外对于高性能计算环境基础的研究，最后介绍了本课题的主要内容。

第二章 相关技术，对在运行支持平台前端设计和实现过程中涉及到的这些关键技术的背景、特点以及在课题中使用情况进行了简单的介绍。其中有Bootstrap、 jQuery这样前端开发，也有RESTful和SSL/HTTPS数据接口调用技术。

第三章 运行支持平台前端的整体结构，对运行支持平台前端的整体结构、层次结构进行了分析与介绍，阐明了采用分层结构设计的好处，同时对每一层的作用与功能进行了介绍。

第四章 支持平台权限管理，介绍了运行支持平台权限管理的设计以及用户管理权限控制的核心算法。

第五章 环境运行，简单介绍了Nagios和CGI的基本情况和数据图表生成方式，详细介绍了环境运行模块的页面设计和用户交互。

第六章 用户管理，介绍了用户审批流程、用户管理模块的系统架构以及用户管理模块的页面设计。

第七章 结论，对本文工作进行了总结，并讨论了之后的工作目标。

# 相关技术

在这一章中主要介绍了本课题运行支持平台在实现过程中使用到的一些关键技术。运行支持平台的实现不仅有用户界面的绘制、布局，同时也包含底层数据的调用和传输。其中相关技术不仅有Bootstrap、 jQuery这样前端开发，也有RESTful和SSL/HTTPS这样数据接口介绍，在这一章节中对这些关键技术的背景、特点以及在课题中使用情况进行了简单的介绍。

## Bootstrap

Bootstrap是由著名的社交网站Twitter在2011年8月推出的开源WEB前端框架，用于前端开发的工具包。它由Twitter公司的设计师Mark Otto和Jacob Thornton合作开发，集合CSS和HTML，使用了最新的浏览器技术[4]，为快速WEB开发提供了一套前端工具包。其中包括布局、网格、表格、按钮、表单、导航、提示等等。使用Bootstrap可以构建出非常优雅的前端界面，而且占用资源非常小。

Bootstrap是基于jQuery框架开发的，它在jQuery框架的基础上进行了更为个性化和人性化的完善，形成一套自己独有的网站风格，并兼容大部分jQuery插件。Bootstrap的主要特点是：

* 丰富的组件，其中下拉菜单、按钮组、按钮下拉菜单、导航、导航条、面包屑、分页、排版、缩略图、警告对话框、进度条、媒体对象。
* 人性化的Javascript插件，Bootstrap自带了13个jQuery插件，包括模式对话框、标签页、滚动条、弹出框。
* 可制定的框架代码，可以对Bootstrap中所有的CSS变量进行修改，依据自己的需求裁剪代码。

在本课题中，运行在浏览器中的html文件使用的就是Bootstrap2.0，Bootstrap2.0把现有框架进行了清晰的功能划分，主要分为框架（Scaffolding），基础CSS，构件库和jQuery插件库。在平台前端开发过程中框架（Scaffolding）、基础CSS，构件库和jQuery插件库都有所使用。

Scaffolding主要提供基于网格的各种布局，包括普通网格系统、嵌入式网格，固定布局、自适应布局，同时可以对网格和布局进行自定义。Bootstrap2.0提供了响应式设计，可以通过单个文件支持各种手持设备，自适应不同的设备和屏幕变化。

基础CSS包括各种排版样式（标题、段落、引用块、列表、内联标签等），代码展示方面提供了基于code标签的内嵌代码，基于pre的块代码和基于Google Prettify的代码样式。同时提供各种表格、表单、按钮、图标的展示方式。

构件库提供了基于按钮、导航、标签、排版、警告、进度栏、图像网格等控件。

jQuery插件库则提供了十几种插件实现动态效果，例如Modal、Dropdown、Tab、Tooltip、Popover、Carousel等，开发者可以根据自己的业务需求使用不同的插件实现各种动态效果。

## jQuery

jQuery是一个优秀的Javascript框架。它是轻量级的js库，jQuery 是一个优秀的 Javascript 框架，它能使用户更方便地处理 HTML 文档、events 事件、动画效果和 Ajax 交互[5]。

本课题中静态页面和CSS效果通过使用Bootstrap框架完成，而前台的绘制、事件与服务器端的数据交互都是通过jQuery完成。在本课题中主要使用了jQuery以下的几个功能：

1. 访问页面框架的局部。这是 DOM 模型所完成的主要工作之一，DOM获取页面中某个结点或者某一类结点有固定的方法，而jQuery则大大地简化了其操作的步骤。 使用jQuery的元素选择器和属性选择器可以通过标签名、属性名或内容对 HTML 元素进行选择，要比Javascript简单得多[6]。
2. 修改页面的表现。CSS 可以通过样式风格来修改页面的表现。然而由于各个浏览器对 CSS3 标准的支持程度不同，使得很多CSS 的特性没能很好地体现。可以通过jQuery获取到元素的位置，对元素的CSS属性进行修改，完成对页面的表现工作。
3. 更改页面的内容。同修改页面的表现原理一致，jQuery获取到元素的位置，可以对元素的属性进行修改，同样可以任意添加、删除某些元素。通过强大而方便的 API括文本的内容、插入新的图片、表单的选项，甚至整个页面的框架。
4. 响应事件。Javascript 有处理事件的相关方法，而引入 jQuery 之后，可以更加轻松地处理事件，而且开发人员不再需要考虑讨厌的浏览器兼容性问题。 在本课题中的用户管理模块，有许多按钮需要添加点击事件，通过jQuery的on方法，监听click事件，可以简化处理事件的相关方法，提高开发效率。
5. 与服务器异步交互。Ajax框架可以简化代码的编写，jQuery也提供了一整套Ajax 相关的操作，大大方便了异步交互的开发和使用。jQuery与服务器异步交互功能同上面的功能一样重要，在平台前端实现过程中，需要与后台进行大量的数据交互，Ajax框架简化了代码的编写，而jQuery使得后台的操作简单化。

通过jQuery的DOM操作、事件监听和动画、表单操作、AJAX以及插件等操作特性，可以完成所有的用户界面开发的需求，展示出简单、灵活的用户交互。同时jQuery学习曲线平缓，适合新手学习，这也是采用jQuery的一个重要原因。

## RESTful

REST[7] (REpresentation State Transfer)描述了一个架构样式的网络系统，比如 web 应用程序。它首次出现在 2000 年 Roy Fielding 的博士论文中，Roy Fielding同时也是 HTTP 规范的主要编写者之一。REST 指的是一组架构约束条件和原则。满足这些约束条件和原则的应用程序或设计就是 RESTful。RESTful更准确的描述应该是一种风格，满足架构约束条件和原则。

Web 应用程序最重要的 REST 原则[8]是，客户端和服务器之间的交互在请求之间是无状态的。从客户端到服务器的每个请求都必须包含理解请求所必需的信息。如果服务器在请求之间的任何时间点重启，客户端不会得到通知。此外，无状态请求可以由任何可用服务器回答，这种特性十分适合云计算之类的环境。用户管理API接口为集成SCE环境的用户管理功能，RESTful的HTTP和HTTPS的服务端口，支持账号申请的工作流程；支持账号的管理功能；对操作进行记录；允许工作人员在操作过程中添加备注信息[9]。

在服务器端，应用程序状态和功能可以分为各种资源。资源是一个有趣的概念实体，它向客户端公开。资源的例子有：应用程序对象、数据库记录、算法等等。每个资源都使用 URI (Universal Resource Identifier) 得到一个惟一的地址。所有资源都共享统一的界面，以便在客户端和服务器之间传输状态。使用的是标准的 HTTP 方法，比如 GET、PUT、POST 和 DELETE。在支持平台中使用的RESTful接口，其URL为

http[s]:// api.scgrid.cn/ v1/{function}/{resource}?{query\_string}

URL里面的function表示不同类型的资源集合，如users和account等类别，URL页面的resouce表示不同的资源，如username 、profile，ldap和username/status等，这些都是符合RESTful资源在URI中的要求。

同时REST框架可以扩展到大量客户端的应用程序，比如用户管理接口可以不止支持平台使用，同样支持大量客户端的应用程序。它还降低了客户端和服务器之间的交互延迟。统一界面简化了整个系统架构，改进了子系统之间交互的可见性。REST 简化了客户端和服务器的实现。

## SSL/HTTPS

HTTPS（Secure Hypertext Transfer Protocol）安全超文本传输协议，HTTPS是以安全为目标的HTTP通道[10]。HTTP下加入SSL层，HTTPS的安全基础是SSL，提供了身份验证与加密通讯方法。它是一个URI scheme(抽象标识符体系)，句法类同http:体系。用于安全的HTTP数据传输。https:URL表明它使用了HTTP，但HTTPS存在不同 于HTTP的默认端口及一个加密/身份验证层（在HTTP与TCP之间）。这个系统的最初研发由网景（Netscape）公司进行，提供了身份验证与加密通讯方法，现在它被广 泛用于万维网上安全敏感的通讯，例如交易支付方面。

网景公司开发HTTPS并将其内置于其浏览器中，用于对数据进行压缩和解压操作，并返回网络上传送回的结果。HTTPS实际上应用了网景的完全套接字层（SSL）作为HTTP应用层的子层。（HTTPS使用端口443，而不是象HTTP那样使用端口80来和TCP/IP进行通信。）SSL 使用40 位关键字作为RC4流加密算法，这对于商业信息的加密是合适的。HTTPS和SSL支持使用X.509数字认证，如果需要的话用户可以确认发送者是谁。

在平台首页用户登录模块以及登陆RESTful接口中，都会使用HTTPS（全称：Hypertext Transfer Protocol over Secure Socket Layer）。用户登录需要提交用户名和密码，如果使用HTTP连接，使用明文传输安全性较差，因此采用HTTPS协议，由SSL+HTTP协议构建的可进行加密传输。

# 运行支持平台前端的整体架构

运行支持平台的目的是主要研究资源监控和运行管理技术、统一认证与系统安全技术，构建环境的运行支持平台，为国家高性能计算基础服务的稳定运行和长期发展提供支持。

运行支持平台的整体框架如图3.1：

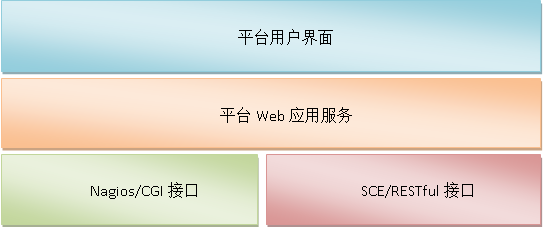


图3.1 运行支持平台的整体框架

运行支持平台自上而下分为三个层次，平台用户界面；平台web应用服务；平台数据接口（Nagios/CGI接口和SCE/RESTful接口）。平台用户界面运行在浏览器端，主要是通过html和Javascript完成，完成与用户的直接交互、数据信息的展示，需要向下调用平台web应用服务。平台web应用服务为网站的服务器端，使用java实现，通过http请求向上提供平台用户界面所需要的数据，平台web应用服务需要调用底层的平台数据接口。平台数据接口（Nagios/CGI接口和SCE/RESTful接口）主要提供平台web应用服务所需数据。

将运行支持平台划分为三个层次，符合网站开发的设计流程，可以借鉴计算机网络[11]的层次结构对运行支持平台层次结构进行理解，各层之间是独立的，某一层不需要知道它的下一层是如何实现的，只需要知道该层通过层间的接口所提供的服务，将一个难以处理的问题分解为若干个比较容易处理的小问题，平台的开发难度也会降低。同时，分层具有良好的灵活性，当任何一层发生变化时，只需要关心如何保持接口不变，它的上层和下层都不受到影响。而且分层的结构易于维护和实现。在平台前端开发的过程中同样使用了分层的思想，将层次进行进一步划分。

本课题完成的部分为平台用户界面和平台web应用服务的设计与实现，平台数据接口（Nagios/CGI接口和SCE/RESTful接口）是由中国科学院网络信息中心提供。

平台用户界面和平台web应用服务的设计与实现过程中同样采用了分层的思想，对整体的层次结构进行更为详细的划分，如下图3.2所示。



图3.2 平台用户界面和平台web应用服务分层结构

平台web应用服务的逻辑代码部分可以见图3.3，其中有就有servlet、service和dataservice，四层结构主要体现在首页登录、用户管理和网站管理，在用户管理模块中数据后台使用的是RESTful接口，在环境运行模块主要使用了CGI接口。

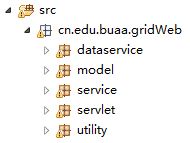


图3.3 平台web应用服务的逻辑代码分层结构

将平台用户界面和平台web应用服务又细分为四层，同整体的分层设计一直，每一层之间都是独立的，只需要关心如何保持接口不变，它的上层和下层都不受到影响。

最上面一层为运行在浏览器端的用户界面，通过html和Javascript实现，主要负责页面的绘制、用户的交互以及和后台的交互。用户界面共有5个一级标题，13个二级标题。其中完成了首页、环境运行、管理维护和网站管理中一部分功能。主要完成的二级界面有集群实时、集群统计、运维服务以及用户管理等二级页面。用户界面使用Ajax的技术，在与用户交互时，某些事件的触发比如加载集群实时数据不会刷新全部页面，而是加载改变的页面。用户界面与后台数据的交互使用jQuery的get、getJson和post方法实现。

Servlet一层主要是http servlet，主要是由doGet和doPost方法构成，用来处理来自客户端的http请求，doGet方法处理Get请求，doPost方法处理Post请求。Servlet一层中的类继承父类BaseServlet，主要完成两个任务：1、接受用户界面的数据，调用 service接口进行处理，直接将service一层的数据交给用户界面；2、对用户的权限进行验证，判断用户是否有权限进行Get或Post请求。

Service提供相应的接口处理servlet的各种请求，完成请求的各种功能，其中包括数据的查询、增加、修改和删除，在Service一层中将Servlet传入的参数格式（主要是JSON格式）进行反序列化，转换为RESTful接口需要的形式，调用data service的接口，得到结果数据后，转换为客户端需要的格式和内容，把数据传输给Servlet。

Data Service作为框架的最底层主要负责调用RESTful接口，其重要的作用是用来调试RESTful接口，因为RESTful接口是由第三方提供，在调用过程中可能出现问题，包括接口本身的错误，调用方法的错误，网络的问题等，Data Service可以看做在平台web应用中把RESTful接口又封装了一层，因为在Service一层已经将数据格式转换为RESTful接口调用的参数形式，Data Service只需要使用Service传入的参数调用RESTful接口，返回RESTful接口的数据。

# 运行支持平台权限管理

## 权限管理设计

权限管理是指根据系统设置的安全规则或者安全策略[12]，用户可以访问而且只能访问自己被授权的资源，用户只能进行系统已经授权的权限进行操作[13]。权限管理技术，一般使用基于角色访问控制技术RBAC（Role Based Access Control）[14]，系统提供的功能：角色管理界面，由用户定义角色，给角色赋权限；用户角色管理界面，由用户给系统用户赋予角色。

在权限粒度上，权限管理一般有两种方式实现：分组和细分到个体。组策略即：将系统用户进行分组，每一组的权限是一致的，将权限一致的人员编入同一组，然后对该组进行权限分配。这样的好处的是，用户利用管理，新增用户可以直接确定所在就可以完成对用户的权限分配，在系统管理过程中直接对组进行操作更加简单、方便。2、如果系统中每个用户的权限很分散，不同用户权限不一致，同时用户个数不是很多，如果进行分组可能导致组数很多反而更加复杂。如果维护一个用户权限表，不同的用户赋予不同的权限，可以更加方便。细化到个体的好处是用户的权限更加灵活，但管理和维护相对复杂，适用于用户较少的系统。

在支持平台开发过程中，不同角色有不同的权限和访问限制。支持平台将权限管理分为两部分页面访问权限管理和用户管理权限控制：1、页面访问权限管理采用的是分组的策略，一共分为三组：未登录、登录普通用户、管理员。支持平台共有5个一级子页面。其页面访问的权限设置如下：未登录的用户只可以进入首页；已经登录的普通用户组可以查看首页、环境运行、管理维护和技术支持页面；管理员可以查看首页、环境运行、管理维护和技术支持，同时管理员用户可以进入网站管理进行操作。其分组页面权限情况如下表所示：

表4.1 页面权限分组情况表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 首页 | 环境运行 | 管理维护 | 技术支持 | 网站管理 |
| 未登录组 | √ |  |  |  |  |
| 普通用户组 | √ | √ | √ | √ |  |
| 管理员组 | √ | √ | √ | √ | √ |

2、用户管理模块中，需要不同角色的管理员对不同状态的用户进行审核、修改和批准，管理员的数量相对较少，同时用户角色较多，如果采用分组的策略不够灵活，因此，将权限细化为用户个体，根据平台业务需要，一共定义了9种权限内容，系统管理员可以对用户进行权限的增加、删除、修改，其中有：添加备注权限、用户信息编辑权限、审批权限、创建账户权限、创建映射权限、创建VPN权限、创建试用账号权限、创建正式账号权限、重置密码权限。这些权限类型细化到Get或Post方法，用户管理的权限控制具体体现在Servlet接受到用户发送的Get或者Post请求，判断该用户是有有权限进行该操作。如图4.1所示，在用网站管理-用户权限管理中可以对账号进行用户管理权限的设置。

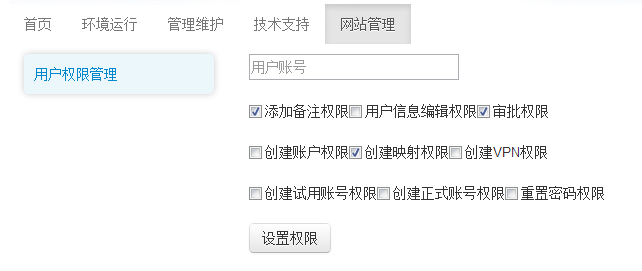


图4.1 用户权限管理页面

## 用户管理权限控制的算法实现

在用户管理模块中，不同的管理员有不同的角色和身份，有些管理员负责审核用户账户，有些管理员负责为用户创建VPN，有些管理员负责设置试用账户或者正式账号。由于不同管理员有不同的角色，也就是有不同的权限，因此需要进行权限控制。在这一章节中第一部分介绍管理员进行用户流程操作的流程，尤其是针对用户权限控制的流程，第二部分主要针对用户权限控制中涉及到的关键算法进行介绍。

以管理员批准试用用户转正为例介绍用户管理权限控制的流程，用户管理权限控制流程图为图4.2：

管理员进行用户转正操作，即向BuildOfficialStatusServlet发送Get请求，BuildOfficialStatusServlet通过doGet(HttpServletRequest, HttpServletResponse)方法接收到Get请求，通过HttpServlet- Request获得管理员信息，通过管理员信息检查管理员用户管理权限，如果管理员没有用户转正权限，向客户端发送权限验证失败的信息，如果管理员具有用户转正权限，继续完成用户转正操作。

用户管理权限控制的算法实现要满足两个条件，实现本地化存储和方便权限的增加和修改。用户管理权限的数据不在SCE/RESTful接口中存储，平台需要实现本地化存储。现在的需求只需要9个权限类型，但是这个可能会在以后的开发过程中发现变化，进行修改或者添加、删除。



图4.2 用户管理权限控制流程图

有两种方式可以实现用户和权限的储存，一种方式是一个用户对应一个权限列表，这样的好处是权限个数不受限制，但后期维护相对较为复杂，并且储存的信息较多，为了简化平台，平台会把用户管理信息用文件的形式永久化存储而不是使用数据库。第二种方式是，使用位运算中的与运算[15]，一位代表一种权限，一个用户对应一个整型，如果与运算为0即该用户有此操作的权限，如果为1其用户没有操作权限，储存操作相对于第一种方法更加简单、快速。经过讨论和考虑打算采用第二种方式实现用户管理权限控制。一个用户对应了个长整型（long），长整型一共就有64，这样一共可以有64种权限类型，在数量上是可以满足以后的需求变化。

首先构建MethodPrivacyMapping类，如图4.3，MethodPrivacyMapping会记录两类信息，权限类型所代表的位数和权限类型的描述。权限类型所代表的位数使用长整数的数据结构，权限类型的描述采用的是字符串。

图4.3 MethodPrivacyMapping类的主要属性

class MethodPrivacyMapping {

public long AccountServlet\_POST = 1l;

public long EditAccountInfoServlet\_POST = 2l;

public long ApprovalStatusServlet\_GET=4l;

…

public String AccountServlet\_POST\_CString = "添加备注权限";

public String EditAccountInfoServlet\_POST\_CString = "用户信息编辑权限";

public String ApprovalStatusServlet\_GET\_CString="审批权限";

…

}

记录权限类型的描述的好处是在进行权限验证时，可以将传入的参数设置为class，利用java反射机制得到类名，通过类名作为MethodPrivacyMapping类的参数名得到权限位数，使用位运算进行判断。

public static boolean checkPrivacy(String name,

String action, Class<? extends BaseServlet> class1) {

String keyString = class1.getSimpleName() + "\_" + action.toUpperCase();

Class clazz = MethodPrivacyMapping.class;

MethodPrivacyMapping instance = new MethodPrivacyMapping();

For ( Method m:clazz.getMethods()){

If (m.getName().equalsIgnoreCase("get" + keyString)){

Long value = (Long)m.invoke(instance,null);

Long p = *getPrivacyFromMap*(name);

If ((p & value)!=0)

return true;

else

return false;

}

}

return false;

}

图4.4 checkPrivacy方法的核心代码

PrivacyManager类用于权限管理，PrivacyManger有一个属性privacyMap，privacyMap为HashMap，其key值为用户名，其Value值为长整型，即用户权限信息。在平台运行之前加载privacyMap，如果用户管理权限发生改变，privacyMap也会进行相应改变。PrivacyManager类有一个重要的方法checkPrivacy()，用于检查权限，其核心代码如图4.4。权限检查的方法需要传入三个参数，用户名、方法类型和需要进行权限验证的类，利用java反射的机制可以得到类名即servlet的类名，类名和方法类型可以在MethodPrivacyMapping类中查询得到其位数，即上述代码中的Long value = (Long) m.invoke (instance, null);通过PrivacyManager类中的privacyMap，查询得到用户名的权限值，两者进行与运算，如果与运算为0即该用户有此操作的权限，如果与运算的结果为1其用户没有操作权限。

采用以上的流程和算法完成用户管理权限控制，可以将每一个权限细化到每一个用户，更加灵活、方便，同时采用与运算的方式可以保证本地化存储用户权限信息时只需要保存用户名和权限值两个字段，同时计算简单、快速。利用java反射的机制可以应对需求的变化，当权限种类发生变化不需要对代码做大的调整。

# 用户管理

用户管理模板是运行支持平台的重要组成部分，用户管理的主要目的是在国家高性能计算环境中管理人员可以对用户信息、用户状态进行查询申请，对用户申请进行审核、批准。用户管理模块主要涉及三个功能：用户申请流程管理；用户申请信息管理；用户审批权限管理。在这一章节中主要介绍的是用户申请信息管理和用户申请流程管理，用户审批权限管理在支持平台权限管理章节中已经进行了详细的介绍。

用户申请信息管理主要是统计不同状态的用户个数、通过用户名、申请人等条件进行用户的查询以及不同状态下用户信息的展示。用户申请流程管理主要是针对多个部门对不同状态的用户申请进行审批，完成用户流程状态的跳转。

## 用户审批流程

用户申请审批是用户管理的重要环节，而用户状态较多、用户审批流程较为复杂，就需要对用户审批流程详细分析。用户审批流程的难点在于在用户申请流程的过程中需要多个部门对不同状态的用户进行审批，而且用户的状态较多，有用户申请状态、开通平台账号状态、VPN账号状态、集群映射状态、试用账号状态、正式账号状态等多个状态。同时还有映射失败，超时等多种额外的状态。

由于用户状态种类较多，用户状态跳转较为复杂，需要设计用户状态转移图，在系统操作的过程中。除了超级管理员具有用户状态随意跳转的权限外，在其他管理员操作用户状态跳转过程中必须遵循用户状态转移图。

根据业务需求和用户状态逻辑，设计了用户申请流程图和用户状态转移图。



图5.1 用户申请流程图

 图5.2 用户状态转移图

从用户申请流程图和用户状态转移图综合分析来看，首先用户提交申请，用户状态处于“用户申请”状态，当用户提交完申请材料并对申请材料确认后，用户状态变为“用户确认”状态，对于已经完成用户确认状态的用户，管理员对申请材料进行初审，如果申请材料审核不通过，进行退修或者修回的状态，期间可以对用户进行无效处理。通过初审之后，用户进入创建账户状态，相关管理员需要为该用户建立账户，账户建立完成之后，需要为用户建立集群映射，用户状态变成了“账户完成建立”的状态，在为用户建立集群映射过程中，可以出现三种情况，全部映射成功、部分映射成功、映射失败，不管用户映射为何种结果，当管理员为用户建立过集群映射后、用户状态进入“待创建VNP”状态，同时如果发生用户集群映射失败的情况后会进入映射失败机制。用户状态进入“待创建VNP”状态，需用相关管理员为其用户创建VPN。创建成功后，用户已经变为试用用户，考核期过后，可以将用户批准为正式用户。

合理的用户申请流程和用户状态转移不仅仅是满足系统需求设计的必要条件，也是保证系统正常运行的首要条件。

## 用户管理模块系统框架

在这一章节中将详细讲述用户管理模块的实现过程，在整体结构中已经介绍了运行支持平台采用分层的结构，从用户界面到后台数据调用每一层都是相对独立，只要保证层次之间的接口不变可以对每一层内部的详细实现方案进行修改。通过这一章以用户管理为例，详细介绍分层结构的实现。如图5.3所示，四层结构主要体现在首页登录、用户管理和网站管理，在用户管理模块中数据后台使用的是RESTful接口，在环境运行模块主要使用了CGI接口。

最上面一层为运行在浏览器端的用户界面，通过html和Javascript实现，在这次层中使用jQuery的get、getJson和post等方法与HTTP Servlet的交互，其交互的数据格式使用的是JSON格式。JSON (Javascript Object Notation) 是一种轻量级的数据交换格式，JSON采用完全独立于语言的文本格式，JSON 是 Javascript 原生格式，在 Javascript 中处理 JSON 数据不需要任何特殊的 API 或工具包，而在Java中也提供很多处理JSON数据格式第三方Jar包。因此jQuery和后台交互的数据格式选用了JSON格式。



图5.3 用户状态转移图

jQuery可以直接解析JSON数据，以下面图5.4的JSON数据为例，json.name就可以得到“广东”，json.cities.city[1]就可以得到字符串“深圳”。在使用Post方法给后台传输JSON字符串时，使用JSON.stringify(json)就可以得到JSON字符串。

{

"name":"广东",

"cities":{

"city":["广州","深圳","珠海"]

}

图5.4 JSON实例数据

Java后台使用第三方jar包org.json用来解析和生成JSON数据。使用org.json的jar包，可以直接构造JSONObject和JSONArray，在对JSONObject和JSONArray对象初始化时，可以直接传入JSON字符串。org.json提供大量的方法对JSON对象进行解析[16]，比如getString( string )，getJSONArray( string )，getJSONArray( string )等，通过put()方法可以对一个JSONObject添加内容[17]。

HTTP Servlet一层主要是由doGet和doPost方法构成，用来处理来自客户端的http请求，doGet方法处理Get请求，doPost方法处理Post请求。Servlet一层中的类继承父类BaseServlet，HTTP Servlet一层首先通过checkPrivacy()对用户的权限进行验证，判断用户是否有权限进行Get或Post请求。完成对用户权限的验证后，接收用户界面的数据，调用 service接口进行处理，直接将service一层的数据交给用户界面。

Service一层主要完成用户管理的逻辑和计算，接收到从HTTP Servlet传入的JSON字符串在Service一层进行解析，将JSON参数转换为RESTful需要的参数，调用Date Service接口，将Data Service返回的结果转换为JSON数据格式向上传递给Servlet。

Data Service作为框架的最底层主要负责调用RESTful接口，Data Service可以看做在平台web应用中把RESTful接口又封装了一层，用来解决在调用过程中可能出现问题，包括接口本身的错误，调用方法的错误，网络的问题等。Data Service一层中有一个RESTMaster，通过静态方法维持一个单例，同时有一个HashMap的connectionMap属性，用来存放用户名和ApiHttpClient4Account类，通过下面的静态方法维持单例。

图5.5 RESTMaster静态方法维持单例方法

public class RESTMaster {

private static RESTMaster *instance*;

private RESTMaster(){

}

public static RESTMaster getInstance(){

if (*instance* == null)

*instance* = new RESTMaster();

return *instance*;

}

private HashMap<String,RESTConnnectionImpl> connectionMap = new HashMap <String,RESTConnnectionImpl>();

}

当用户登录后会实例化ApiHttpClient4Account，将用户名和RESTConnnectionImpl添加到connectMap。当调用Data Service接口时，需要传入用户名，实例化一个RESTMaster

RESTMaster instance = RESTMaster.getInstance();

当调用Data Service接口时通过传入的用户名，在instance中的connectMap查找到实例化的ApiHttpClient4Account，进行RESTful接口调用。

## 用户管理模块的页面设计

### 5.3.1 用户管理首页设计

用户管理首页是用户管理的第一个页面，用于展示统计不同状态待办用户的个数，用户查询机制和映射失败的处理。利用jQuery的Ajax技术[18]，使网页实现异步更新，可以在不重新加载整个网页的情况下，对网页的某部分进行更新。不同状态待办用户的个数，用户查询机制和映射失败的处理三部分可以不需要加载整个网页，用户在用户管理首页就可以进行部分更新完成全部操作。AJAX 在浏览器与 Web 服务器之间使用异步数据传输（HTTP 请求），这样就可使网页从服务器请求少量的信息[19]。



图5.6 用户管理首页页面

如图5,6所示，用户管理首页由三部分组成，第一部分为不同状态待办用户的个数统计，其中有待确认、待审批、待建立、带映射、待创建VPN、待设置试用用户、待设置正式用户，7种状态的待办用户，不同状态待办用户的个数统计可以方便不同角色的管理员对不同状态的用户进行审核。在个数统计栏中还有映射失败的选项，用来统计用户和集群映射失败的条目会在映射失败机制上进行详细介绍。第二部分为用户信息查询，其中有4种选项，用户名、邮箱、申请人和状态，单击查询按钮就可以按照输入条件进行查询，如图5.7，输入“hsh\_9”用户会在下方显示该用户的信息。这些查询条件可以进行组合查询，条件为空时表示任意，当多个条件不为空时，如图5.8，需要满足所有的条件，当不为空的条件有一项不满足将返回没有找到的提示，全部满足条件会得到相应用户如图5.9。

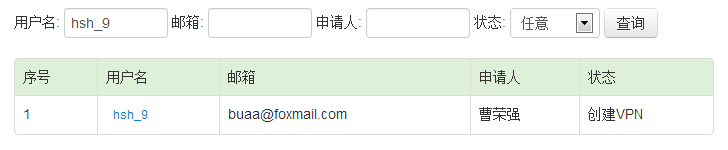


图5.7 用户管理首页用户查询页面

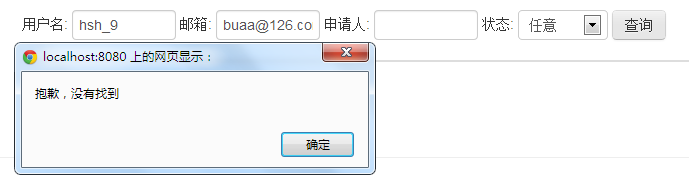


图5.8 用户管理首页用户查询失败页面

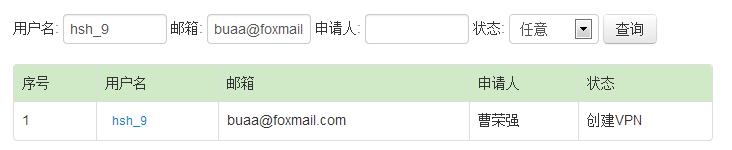


图5.9 用户管理首页用户组合查询页面

第三部分用来展示用户的查询结果，通过表格的形式进行展示，共分为5列，序号、用户名、邮箱、申请人和状态。对于失败机制，同样是在第三部分的位置使用表格进行展示，共有用户名、集群和映射账号3列。

映射失败机制是用户管理首页重要的组成部分，用来处理用户创建集群映射时出现的失败异常情况。映射失败机制要完成对映射失败的用户名、集群名和映射账号的本地存储，查询统计，条目显示和一键重试等相关功能。

映射失败机制产生的原因是当管理员对用户和集群建立映射时由于种种原因出现映射失败的情况，但是RESTful接口在完成映射失败时是调用了系统的其他接口，双方都没有进行映射失败信息的存储，为了保证支持平台的完整和正确以及良好的用户体验，运行支持平台需要对失败条目进行本地化存储，这也是支持平台中为数不多的需要进行数据存储的模块。



图5.10 用户管理首页映射失败页面

映射失败机制的过程：管理员在进行创建用户与集群映射时，可能因为集群自身故障、网络环境不好等原因导致用户与集群映射失败，支持平台需要记录失败条目，当RESTful接口返回失败信息时，后台会自动记录映射失败的用户、集群名和集群账号，完成失败条目的本地化存储，储存的方式使用ObjectOutputStream可以直接将对象序列化为二进制文件，在读取失败信息时，采用ObjectInputStream直接读取文件获得Java对象。Java对象使用的是HashSet可以保证失败条目不会重复。

当用户单击映射失败的条目时如图5.10，使用用户名、集群名和映射账号三列表格进行展示。在表格下方有一个重试的按钮，当用户点击重试按钮后会对以上所有的失败条目进行一键重试，如果用户在集群中的映射建立成功将删除失败条目，如果仍然没有成功，将保留失败条目。

用户管理首页是进入用户管理的入口，提供了查询、统计和映射失败的功能，方便管理员进行查看、查询、审批。同时，用户管理首页采用Ajax的技术，在用户进行操作时不需要加载全部页面，只需要部分更新，主要是表格内容的更新。通过AJAX 使得浏览器与 Web 服务器之间进行异步数据传输，减少网页从服务器中的请求信息。

### 5.3.2 用户管理审批页面设计

用户管理审批是按照用户申请流程图，从用户申请开始，经过初审、退修、创建账户、建立映射、创建VPN、创建试用账号、创建正式账号这些步骤，不同部门的管理人员针对不同的账号进行审批。



图5.11 用户流程管理主页面

管理人员单击某一用户名后，就可以进入用户的个人页面，如图5.10。不同状态的用户界面有所不同，下面的操作按钮也不同。用户审批界面同样使用了Ajax技术，当用户操作时不需要对整个界面进行刷新，只需要对页面的部分内容进行更新。

用户信息表由三部分组成，第一部分的第一栏为用户名和用户状态，第二栏为用户流程图，用来提示管理员和用户现在已经完成的状态和整个申请流程。第三栏为用户操作跟踪，使用了Bootstrap的Javascript插件，由于用户操作过多默认用户操作为折叠形式，当管理员单击全部申请表跟踪，如图5.12会显示全部的每一个操作条目，其中包括操作时间、操作类型和操作人。用户信息表的第三部分为申请表信息，用来显示用户申请表的详细信息，其中有用户名、通信地址、邮箱、电话、申请人等用户申请表的各项信息。同时还有备注添加功能，可以在此界面添加备注。对于已经完成集群映射的用户来说，在用户信息一栏中可以查看集群映射情况，点击进去之后，如图5.13.可以查看用户名、集群、映射账号和映射状态。同时在这一界面也可以继续添加其他集群映射。

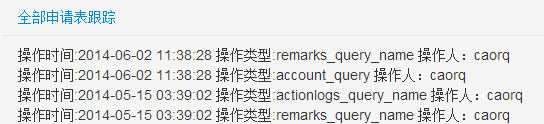


图5.12 展开申请表跟踪图

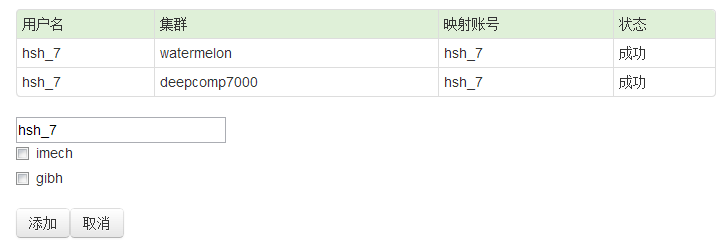


图5.13 展开申请表跟踪图

用户信息表下方为操作按钮，根据用户状态不同操作按钮不同，以图5.11为例，用户为退修状态，管理员可以对用户进行通过或者继续退修，也可以将该用户设置为无效。这些操作都可以改变用户状态，同时管理员还可以对用户信息进行编辑和修改。当管理员点击编辑按钮，页面接收到点击事件后，使用Ajax技术，在用户信息的表的位置进行刷和修改，出现如图5.15的页面，其中有通信地址、邮箱、电话、申请人等8项信息可以编辑。当管理员为超级管理员时，用户编辑页面会出现用户状态一栏，超级管理员可以完成对用户状态的跳转。

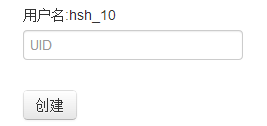


图5.14 建立账号页面



图5.14 用户信息编辑图

管理人员单击用户信息的按钮可以对用户状态进行操作，其中用户初审通过、退修、创建VPN、设置试用用户和设置正式用户这些操作管理人员直接单击按钮就可以完成，但对于建立账户和集群映射需要管理人员添加相关信息，比如在建立账户时需要管理员添加Uid，如图5.14，在建立集群映射时需要管理人员选择需要映射的集群。

# 环境运行

环境运行模块用来展示高性能计算环境的计算资源的运行情况，完成计算资源的可视化，其中包括CPU占用率、磁盘占用率等计算资源指标的可视化。环境运行可以在高性能计算环境实际生产运维期间提供可靠的环境数据监控，对环境运维决策制定的提供有效、准确的数据支持。环境运行模块主要是使用Nagios/CGI接口作为数据调用接口。

环境运行模块的结构如图6.1，主要由集群实时、集群统计、运维服务三部分组成，在环境运行的用户界面使用Html和jQuery完成，jQuery通过Ajax与Servlet进行异步调用，获取界面展示的图表和数据，Service负责调用Nagios/CGI接口获取后台数据，将数据向上传输给Servlet，Servlet只负责从http请求中获取用户传入的参数，将参数传给Service，在Servlet一层不会进行逻辑判断和运算。环境运行模块的层次结构与支持平台整体的层次结构一致，也同样具备分层结构的优点与好处。

图6.1 环境运行模块整体架构

环境运行模块以下三部分组成：集群实时、集群统计、运维服务。集群实时主要是以天单位显示大连、合肥、昆明、青岛等集群结点的CPU利用率/结点利用率和磁盘利用率。集群统计主要是以月为单位，统计显示CPU历史数据，其中集群结点与集群实时子界面的结点内容一致。运维服务主要是以天单位，显示各个运维服务器的服务器负载和磁盘利用率。

## Nagios/CGI

环境运行模块中的图表和数据是通过调用Nagios/CGI接口完成的。

Nagios是一个监视系统运行状态和网络信息的监视系统。Nagios能监视所指定的本地或远程主机以及服务，同时提供异常通知功能等。Nagios可运行在Linux/Unix平台之上，同时提供一个可选的基于浏览器的WEB界面以方便系统管理人员查看网络状态，各种系统问题以及日志等等。Nagios主程序根据预先定义好的所有需要监控的项目的配置文件，在定义的时间点对主机或服务进行检测。这一检测工作主要是通过插件来完成并返回一个明确的状态给Nagios守护进程。Nagios根据这一信息给出相应操作（发警告、记录日志、执行命令等）。

Nagios可以提供很多监控网络功能，其中包括SMTP、POP3、HTTP、NNTP、PING，同时Nagios可以提供监控主机资源的功能，在支持平台中需要的数据就来自于Nagios对处理器负荷、磁盘利用率等主机资源的监控，也可以实现对主机的冗余监控。除了网络信息和主机资源信息的监控，Nagios可以指定简单的插件用来扩展Nagios的监视功能，方便地扩展自己服务的检测方法。除了Nagios提供的监视功能，Nagios另一个优点是对于错误的处理机制，可定义一些处理程序，使之能够在服务或主机发生故障时起到预防作用，同时可以通过Email、Pager或者用户自定义的方式进行错误通知，当服务或主机问题产生与解决时将警报发送给联系人。Nagios另一个便利之处在于基于浏览器的WEB界面以方便系统管理人员查看网络状态，各种系统问题以及日志等等。

CGI是外部应用程序（CGI程序）与Web服务器之间的接口标准，是在CGI程序和Web服务器之间传递信息的规程。CGI规范允许Web服务器执行外部程序，并将它们的输出发送给Web浏览器，CGI将Web的一组简单的静态超媒体文档变成一个完整的新的交互式媒体。

CGI脚本程序可以用C、C＋＋等语言在多种平台上进行开发。无需太大修改就可以从一个平台移植到另一个平台上运行，具有很好的兼容性。

Nagios可以利用Nagios Graph对数据进行图表化，得到各项指标的监控信息图表。首先，Nagios中自定了多个可显示监控信息的CGI脚本，可以根据自己的需要方便地改写它们来获取所需要的信息。比如：利用check\_disk可获取集群的磁盘空间，运维服务器的磁盘空间，check\_load可获取运维服务器的负载等信息。针对集群的CPU/Node利用率，需要根据集群的作业系统的使用方式自己编写相应的插件来获取信息。这里的CPU利用率是指有作业运行的CPU核数与在线的总CPU核数的比值。一般作业系统都采用CPU利用率这个指标来衡量整个集群的运行情况。

有了获取这些信息的插件后，通过Nagios的配置文件来定义这些监控信息，指定获取这些信息的时间间隔。在Nagios平台中每五分钟获取一次数据，数据信息存储在status.dat的文件中。有了实时更新的本地数据之后，可以借助Nagios Graph将数据图表化。Nagios Graph可以将Nagios监控得到的数据存入到RRD文件中，基于这些数据生成图表。根据Nagios Graph中的map配置文件的定义，提取监控得到的数据按照定义的格式存入到RRD文件中。利用RDDtool读取相关文件即可得到我们的所画的图表。

## 环境运行中的用户交互

环境运行模块的后台逻辑与数据接口调用相对简单，用户界面使用jQuery完成逻辑操作，jQuery通过Ajax与Servlet进行异步调用，直接得到图表和数据进行界面展示， Servlet只负责解析http请求，获取用户传入的参数，将参数传给Service；Service负责调用Nagios/CGI接口获取后台数据，将数据向上传输给Servlet。环境运行模块以下三部分组成：集群实时、集群统计、运维服务。集群实时主要是以天单位显示集群结点的CPU利用率/结点利用率和磁盘利用率。集群统计主要是以月为单位统计显示CPU历史数据，其中集群结点与集群实时子界面的结点内容一致。运维服务主要是以天单位，显示各个运维服务器的服务器负载和磁盘利用率。

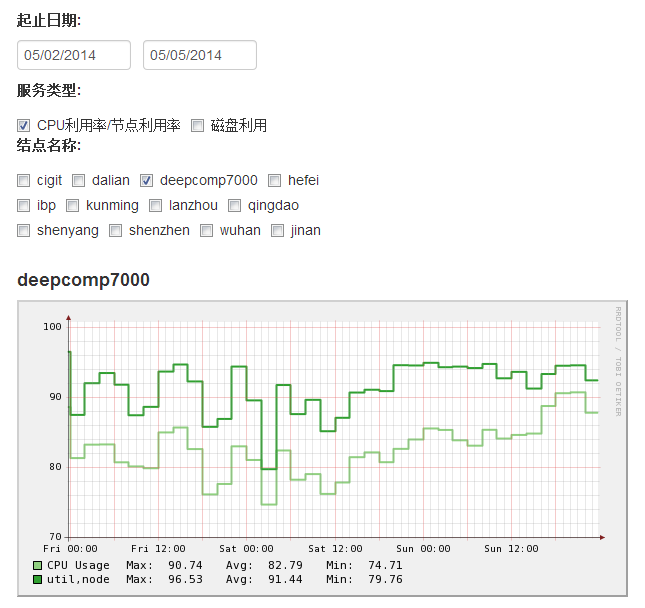


图6.2 集群实时图

共有17个集群结点，根据不同集群结点的名称可以选择想要查看的集群信息。集群结点的取值范围见表6.1：

表6.1 集群结点名称

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cigit | dalian | deepcomp7000 | hefei | ibp |
| kunming | lanzhou | qingdao | shenyang | shenzhen |
| wuhan | jinan |  |  |  |

共有11个运维服务器，运维服务器的取值范围见表6.2：

表6.2 运维服务器名称

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| fs-jinan | fs-ibp | fs40 | fs41 | cs30 |
| cs31 | cs32 | cs33 | client50 | client51 |
| portal81 |  |  |  |  |

环境运行模块用来展示高性能计算环境的运行情况，其中包括CPU占用率、磁盘占用率等计算资源指标。环境运行用来提供可视化的环境数据监控方案，因此环境运行模块的用户界面和用户交互更加重要。

集群实时和运维服务两项子菜单需要用户选择起止时间、服务类型和结点或者运维服务器类型三种参数，集群统计只需要起止时间和结点名称两个参数。

集群实时的参数选择如图6.3：



图6.3 集群实时的参数选择

起止日期选用了Bootstrap的日期选择器Javascript插件如图6.4，如同在相关技术中讲到的，Bootstrap提供了人性化的Javascript插件，Bootstrap自带了13个jQuery插件，日期选择器不是Bootstrap自带官方插件，而是由第三方提供的Javascript插件。通过日期选择器用户可以自由选择年、月、日，一方面简化了用户的操作，只需要鼠标点击就可以完成，另一方面不会因为日期的格式而产生一些不必要的错误。

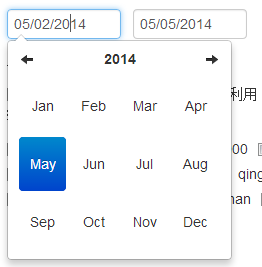


图6.4 Bootstrap的日期选择器

服务类型与结点名称选用的是选择框（check box），单击勾选/取消。这些简单的条件选择方式更加有利于用户使用、查询，更加快速、简单的得到所需要的内容。

除了简单的条件选择方式，环境运行还提供了另一项更加简单的用户交互，当用户选择完查询条件，无需点击确定按钮，下面直接出现所需要的图表。以集群实时为例，当用户选择一项服务类型时，只在页面中摆放所选结点、所选服务的图表，如图6.2，当用户选择两项服务类型时，之前的设计方案是同一结点或者运维服务器，不同服务会平行摆放，如图6.5。但受限于页面宽度，如果进行平行摆放会减小图标的宽度导致图表上的文字很难识别，因此采用了依次罗列的方式，如图6.6，在以后的版本中会针对这个问题进行讨论、分析，得到更好的展示形式。

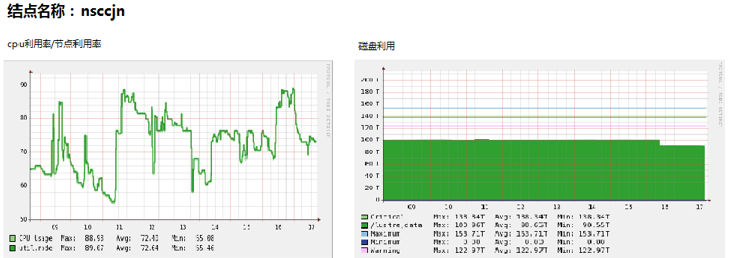


图6.5 平行摆放同一结点不同服务内容

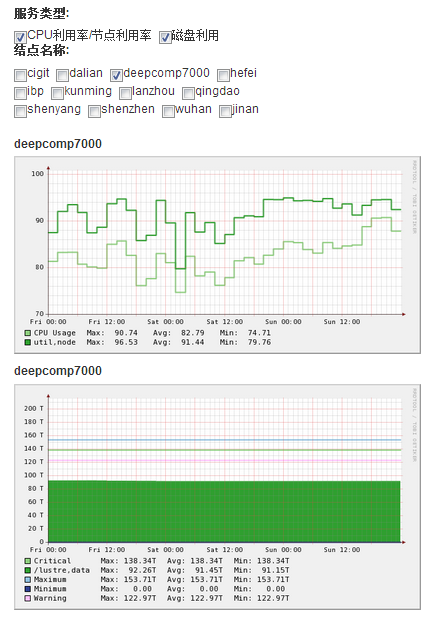


图6.6 罗列摆放同一结点不同服务内容

除了上面介绍的集群实时图表，还有集群统计和运维服务的图表，集群统计只有起止日期和结点选项，主要显示统计每个月所选集群的数据，如图6.7。运维服务的图表同集群实时一样有三种选项，页面逻辑和集群实时一致，只是服务类型和结点名称的内容不同，运维服务的页面见图6.8.



图6.7 集群统计页面

通过在环境运行模块用来展示高性能计算环境的实时集群信息、集群统计信息和运维服务器的信息，完成了高性能计算环境中资源信息的可视化，可以对环境运维决策制定的提供有效、可靠的数据支持。环境运行模块不仅提供了便利的数据可视化方案，同时通过使用Bootstrap的日期选择器、选择框等插件，这些简单的条件选择方式更加有利于用户使用、查询，更加快速、简单的得到所需要的内容。

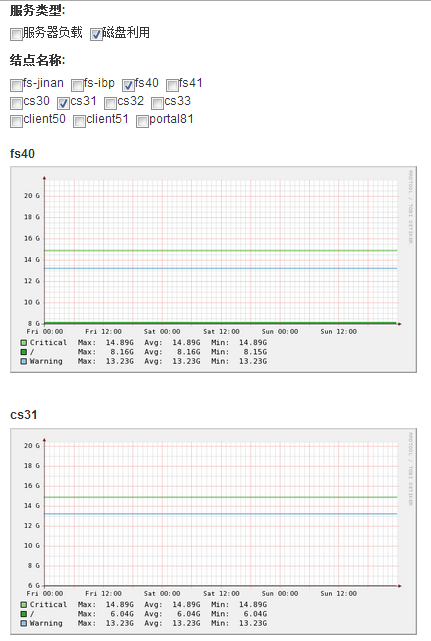


图6.8 运维服务的页面

# 结论

本课题主要完成了运行支持平台前端的设计与实现，其中包括用户登录，平台权限控制，平台用户管理，环境运行展示等。

运行支持平台前端采用分层结构，符合网站开发的设计流程，各层之间是独立的，某一层不需要关心它的下一层是如何实现的，只需要知道该层通过层间的接口所提供的服务，平台的开发难度也会降低。同时分层具有良好的灵活性，当任何一层发生变化时，只需要关心如何保持接口不变，它的上层和下层都不受到影响。分层的结构易于维护和实现。

用户界面使用使用的就是Bootstrap 2.0和jQuery进行开发，使用了Bootstrap 2.0框架（Scaffolding）、基础CSS、构件库和jQuery插件库，使得用户界面样式丰富、用户交互简单直观。使用jQuery的Ajax技术与后台进行数据交换， AJAX 在浏览器与 Web 服务器之间使用异步数据传输（HTTP 请求），这样就可使网页从服务器请求少量的信息。

用户管理权限控制可以将每一个权限细化到每一个用户，更加灵活、方便，同时采用与运算的方式使得权限计算简单、快速，可以保证本地化存储用户权限信息时只需要保存用户名和权限值两个字段。利用java反射的机制可以应对需求的变化，当权限种类发生变化不需要对代码做大的调整。

由于时间的限制，在本课题中有一些不足将在接下来的工作中进行完善，在环境运行模块，受限于Nagios图表的大小很难做到同一结点的不同服务类型平行摆放，因此采用了罗列的方式，不够直观。会在以后的工作中想办法解决这个问题，让图表的展示更加直观、简洁。

致谢

论文完成之际，我想把心里最诚挚的感谢送给所有关心我帮助我的人们，正是他们的帮助和支持使我能够顺利地完成论文。

首先，我要感谢我毕业设计的指导老师钱德沛老师。钱老师是一位学识渊博、治学严谨、和蔼可亲的老师，对学科前沿敏锐的洞察力和平易近人的态度也给我留下了深刻印象。他的学识广博、治学严谨、思维敏锐，在他的指导与督促下，我逐渐学会如何思考系统的设计，代码编写的规范、也让我逐渐掌握科学的研究方法，这对我论文的顺利完成起了不可替代的作用。在毕业设计期间，钱老师为我们营造了良好的学习氛围，让我们在完成毕业设计的同时也可以从中学到很多专业知识。正是在钱老师的督促和指引下，我顺利完成了毕业设计。在此再次感谢钱老师。

同样也要感谢栾钟治老师和孟由师兄。栾老师和孟由师兄在我做毕设的整个过程中，从选题、系统设计与开发、论文撰写与完成，他们都给了我很大的帮助。我的开题答辩不是很顺利，在栾老师的细心讲解和指导下，我对毕设题目有了更深的了解，题目的方向也有了更深认识，在接下来的工作中变得更加顺利。没有栾老师的帮助和指导，不可能在毕设中积累这么多知识和经验。在整个毕业设计过程中，孟由师兄也同样给予了很大的帮助，使我能够在很多关键技术和难点的地方得到突破，毕业设计是使用Java语言编写，是第一次使用Java进行Web编程，刚开始的时候遇到一些困难，在孟由师兄的细心讲解下，程序的编写按照原先的计划一步一步顺利开展，对Java的Web编程有了初步的认识和理解。在孟由师兄的帮助下，我的毕业设计顺利地进行，编码能力和解决问题的能力得到了很好的锻炼，收获颇丰。没有栾老师和孟由师兄的帮助，就很有可能没有这篇论文的顺利完成，在此向他们表示最真挚的谢意。

还要感谢中科院网络信息中心的王小宁师姐、曹荣强师兄、何荣师姐等，没有他们的合作与帮助，我的毕业设计也不会顺利完成。在平台页面设计与数据接口调用过程中遇到一些困难时，他们都会及时给予帮助，一起解决遇到的问题。

面对着电脑奋战毕业设计的日子很快就要结束了，通过这次毕业设计，我不仅仅学到了专业知识，锻炼了编码能力和解决问题的能力，老师们严谨治学、一丝不苟的作风同样是我一生宝贵的财富，从老师身上学到的精神与品质会一直激励、影响着我。

感谢计算机学院的各位老师，是他们辛勤的付出，才换来了我们文化知识的不断提高。感谢王星歌、徐墨客两位辅导员，无论是在学习还是在生活上，你们都给了我很多帮助，你们默默的付出，让我们感到班集体的温暖。

在这里，我还要把最真诚的感谢献给我的父母。从小到大，父母对于我的关心总是无微不至的。上大学是第一次离开父母，独自一人在外漂泊，他们不仅在生活上给了我莫大的关怀，在精神上也给予了很多支持和鼓励。正是带着他们的期盼和鼓励，我不断努力，不断地提高自己。

最后，衷心感谢为评阅本文而付出辛勤劳动的各位老师和评委！

参考文献

[1] 赵毅, 朱鹏, 迟学斌, 等. 浅析高性能计算应用的需求与发展[J]. 计算机研究与发展, 2007, 44(10): 1640-1646.

[2] 李根国, 桂亚东, 刘欣. 浅谈高性能计算的地位及应用[J]. 计算机应用与软件, 2006, 23(9): 3-4.

[3] 张军华, 臧胜涛, 单联瑜, 等. 高性能计算的发展现状及趋势[J]. 石油地球物理勘探, 2010, 45(6): 918-925.

[4] Gould W, Pitblado J. Guidelines for bootstrap samples [html document][J]. Stata FAQ Statistics, 2005..

[5] Chaffer J. Learning jQuery[M]. Packt Publishing Ltd, 2013.

[6] Chaffer J, Swedberg K. Jquery reference guide: a comprehensive exploration of the popular Javascript library[M]. Packt Publishing Ltd, 2010.

[7] Richardson L, Ruby S. RESTful web services[M]. O'Reilly Media, Inc., 2008.

[8] 潘冰. 面向资源的 RESTful Web 应用研究[J]. 微计算机应用, 2010 (7): 38-43.

[9] 袁赟. Java 与 Restful Web Service[J]. 电脑知识与技术: 学术交流, 2008 (11): 780-782.

[10] 邓晓军, 沈浩. HTTPS 中客户端认证问题的分析[J]. 计算机与数字工程, 2008, 36(5): 118-120.

[11] 计算机网络基础及应用[M]. 哈尔滨工业大学出版社, 2000.

[12]暴志刚, 胡艳军, 顾新建. 基于 Web 的系统权限管理实现方法[J]. 计算机工程, 2006, 32(1): 169-170.

[13] 覃章荣, 王强, 欧镔进, 等. 基于角色的权限管理方法的改进与应用[J]. 计算机工程与设计, 2007, 28(6): 1282-1284.

[14] 周文峰, 尤军考, 何基香. 基于 RBAC 模型的权限管理系统设计与实现[J]. 微计算机信息, 2006 (05X): 35-36.

[15] 周敏, 张卫东. 基于二进制位运算的权限控制模型[J]. 常州信息职业技术学院学报, 2009, 8(6): 12-15.

[16] 屈展, 李婵. JSON 在 Ajax 数据交换中的应用研究[J]. 西安石油大学学报: 自然科学版, 2011, 26(1): 95-98.

[17] 张涛, 黄强, 毛磊雅, 等. 一个基于 JSON 的对象序列化算法[J]. 计算机工程与应用, 2007, 43(15): 98-100.

[18] 王星, 潘郁. 基于 AJAX 技术的 Web 模型在网站开发中的应用研究[J]. 微计算机信息, 2006 (09X): 206-207.

[19] Asleson R, Schutta N T. Ajax 基础教程[M]. 人民邮电出版社, 2006.