# 第一章前言

本章首先介绍了项目的背景和意义，然后提出了本文主要的研究方向，并对目前国内外的研究现状进行了分析，同时确定了本文的研究目标和范围。最后简单描述了本文的结构。

## 项目的背景和意义

随着计算机网络的快速发展，计算机网络已经渗透到了社会的各个领域，这导致企业服务器系统数量和应用服务大量增加。管理员需要面对越来越多的应用，不仅需要监控服务器本身是否正常启动运行，还要监控物理服务器上运行的web、数据库、DNS等应用服务是否正常运行。面对大量的上千台服务器的数量，通过管理员手工检测将不能完成这项工作，需要使用网络监控系统，网络监控系统可以使管理员在用户之前发现和解决问题，自动执行例行任务，检查可用资源，从而提高管理员工作效率。

现在主流的开源网络和服务器监控系统有Zabbix、Nagios、Cacti、Ganglia。Nagios采用插件方式进行数据采集，需要使用种类繁多的插件，这就造成Nagios使用困难，Cacti不支持agent数据采集方式，这使Cacti不能全面地检测服务器的状态。随着服务器数量的不断增加，保证服务器和应用服务器的正常运行变得越来越复杂，相比Nagios,Cacti监控系统，Zabbix具有更高的性能和可扩展性，更加适应网络中心机房监控环境，Zabbix集成了SNMP、agent、IPMI等多种数据采集方式，可以方便在不同环境下使用，相比其他2种监控系统更加完善地监控和图形化显示功能。利用Zabbix可以构建分布式的网络监控系统，实现对Windows和Linux平台上关键应用服务的监控。重点研究对Web、数据库和DNS应用进行监控的原理和配置，并给出了在网络中心机房的使用实例。

## 国内外研究现状分析

基于以上背景，国内外有一些关于服务器监控系统的出现，主要有Zabbix、Nagios、Ganglia等，简要介绍如下：

Zabbix提供了先进的检测、报警和可视化功能、可以检测众多的网络参数和服务器的正常运行和完整性，包括服务器、路由器、交换机、打印机等设备。

它具有自动发现功能，可以自动发现被监控网络新加入的服务器和其他网络设备。Zabbix系统采用分布式监控和通过web前端进行集中式管理，通过C/S模式进行数据采集、支持轮询和陷阱2种数据采集机制。除此之外，它能够部署在lunix、solaris、AIX、HP-UX、FreeBSD等Unix操作系统上。

Nagios是一款运行与Lunix系统上的开源网络监控软件，能够按照系统管理员的要求，对多种系统以及其服务项进行实时监控。它是一款开源免费的网络监控软件，运行于linux或unix操作系统，由一个Nagios主程序，一个Nagios-plugins插件和几个可选的ADDON组成，nagios系统通过自身的插件来实现对多系统、服务的监控，它的常用功能：监控各种网络服务，如http、smtp、ping等，监控主机资源如磁盘利用率、cpu负荷等，可以通过插件来扩展自己监控方式；允许多种监控方式运行；当服务设备或网络服务出现异常时，可通过自定义的短信或邮箱方式来通知系统管理员；用parent主机定义来表达网络主机之间的关系，具有网络分层功能，可通过web界面查看监控主机间的关系，避免了Lunix命令字符的复杂性。

由于nagios有着自由可选的各种插件，还可以配合上自定义的shell脚本进行工作，这种灵活的配置方式，全面地监控能力，适合用于各种大小规模的企业，同时被越来越多的数据中心系统管理员所重视。

[Ganglia](http://ganglia.info/)是伯克利开发的一个集群监控软件。可以监视和显示集群中的节点的各种状态信息，如：cpu 、mem、硬盘利用率、I/O负载、网络流量情况等，同时可以将历史数据以曲线方式通过php页面呈现，通过曲线很容易见到每个节点的工作状态，对合理调整、分配系统资源，提高系统整体性能起到重要作用。同时它支持浏览器方式访问，但不能监控节点硬件技术指标。它是能在一个Linux系统下能图形化监控系统运行性能的软件，其界面美观、展示丰富，功能强大。

数据中心监控系统对当今云计算发展的重要性不言而喻。尽管目前对在这种环境下的监控工作，工业界和学术界进行了大量的研究和探索，但是在实际操作中，发现这些工具仍然有大量的不足，还有许多可提升完善的空间

目前商业监控系统存在许多突出问题，使商业系统只能在目前的数据中心运维中处于配角的地位，仅仅作为主要监控系统的补充。主要表现在：

1、特定的硬件要求。以iDRAC和iLO为代表的集群监控系统具有良好的监控效果，作为硬件的增值服务，不对其他硬件平台提供支持，这就使该系统的可用性大大降低，不能满足大规模数据中心的实际部署要求。

2、高额的软件授权费用，难以实现功能扩展。即便是需要特定硬件支持的商业系统，仍然面临着软件授权费用高昂的问题，同时，不断变化的机房规模对系统提出了很高的扩展性要求，闭源的商业软件难以适应用户的这种需求。

除上提到的商业监控系统特有的问题外，开源系统和商业系统还共同面临着诸多其他方面的挑战。

1、未能实现一体化监控。目前的这些系统准确地说是服务器集群监控或者是ICT设备监控，而实际的数据中心中，包含ICT基础设施、空调冷却系统和电力供应系统，这三大系统都对云平台运行具有重要的意义。实现对机房各大系统整体监控、统一调度无疑更具有现实意义。

2、现有的工具各自独立，数据挖掘价值低。实际的数据中心监控系统是一系列互相协作的工具集。但是从目前的运营情况来看，这些系统处于独立运行、互不关联的状态，日志数据也单独保存。而一般认为综合ICT基础设施、空调冷却系统和电力供应系统的数据更易于抽象出有效信息，有利于保障系统运行稳定。

3、数据持久化系统效率低下，结构化数据存储方案难以实现监控项目扩展。系统日志的持久化方案需要实现数据索引、快速响应、易于扩展的特性。同时云数据中心的可扩展性需要数据库具有高效处理非模式化数据的能力。也就是既要保持结构化数据的存储，又能够随时扩充或者减少存储的数据类型。目前尚难以实现这方面的功能需求。

4、缺乏创新高效率的交互界面。当今主流的系统主要采用的是命令行用户界面和web用户界面，这些方案的成熟性和可靠性得到了多年的检验，但是也面临现实效率低下、交互性差、数据可视化程度差的问题，需要新的高效UI界面突破现有瓶颈。

5、实时数据利用率低。故障诊断，实时报警以及系统故障自动排除都是未来的方向，目前系统数据中心日志数据几乎没有利用，只是在出现故障时才会查询，随着数据挖掘等技术的应用，实现对日志的高效利用将是提高数据利用率的重要手段。

## 项目的目标和范围

本文的目标是设计并实现一个开源的服务器运维系统中的服务器监控模块的设计与实现，开源的服务器运维系统具有五大模块，包括服务器部署、服务器管理、应用管理、服务器监控、日志管理。针对问题领域是为服务器简化管理和运维提供方案以及解决实际的openstack的部署和管理问题。能够根据实际需求对批量服务器进行监控，简化服务器，从而推动教育平台的普及和发展。本系统的主要目标有简化服务器管理工作、提供图形化服务器全局视图、持续监控服务器状态、服务器图像化配置、服务器图像化持续管理各项应用服务。

本文是在该系统的基础上参与到其中的两个模块--应用管理与服务器监控，主要包括以下部分：

1. 服务器管理模块：将服务器加入到管理系统中来进行集中部署操作系统，控制服务器的启动与关闭，完成对服务器的注册、电源控制以及操作系统的远程部署。
2. 监控模块：该模块是整个服务器监控系统的核心模块。执行对服务器状态进行持续监测，并针对监测结果提供一定的系统控制操作。
3. 应用模块：主要针对docker，同时对容器进行监控和错误报警的配置。

## 论文结构简介

本文以基于服务器运维系统中的服务器监控模块的设计与实现为主要内容，分为以下部分：

第一章：前言。首先介绍了项目的背景和意义，提出了本文的研究方向，同时分析国内外对集群监控系统的研究现状，并确定了本文的研究目标和范围。最后介绍了整篇文章的整体结构。

第二章：技术与原理。首先介绍了监控模块的核心技术：监控原理介绍，然后介绍了界面语言Django框架与监控对象Docker，最后对本项目的开发语Python语言进行了较详细的介绍和讲解。

第三章：需求建模。对业务进行建模，并指出了项目的利益相关者，同时解释了相关的业务概念。根据对需求的详细分析，设计了系统的用例图，并对关键用例进行了分析，同时确定了系统的领域模型。

第四章：架构设计。介绍了整个系统的整体架构以及主要组件，并对关键用例的实现进行了分析。最后，介绍了系统的数据库设计。

第五章：详细设计。介绍了系统关键模块的详细设计，并指出了相应的关键类和关键方法。

第六章：部署与测试。介绍了整个系统的部署，同时展示了系统使用时的相关核心业务的界面。最后，对本文所提出的系统的功能进行了测试分析。

第七章：总结与展望。对本文的工作进行了总结分析，指出了本文的研究存在的不足之处并对所研究的服务器监控系统提出了展望。