

Integrated Management Platform for Enterprise Application System: Design and Implementation*

Yu Sheng¹, Zhongxin Jiang², Pingping Dong¹, Jia Yuan¹

¹School of Information Science and Engineering, Central South University, Changsha
²Hunan Police Academy, Changsha
Email: vlab@163.com

Received: May 2nd, 2013; revised: May 16th, 2013; accepted: May 28th, 2013

Copyright © 2013 Yu Sheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract: With the expansion of domestic enterprise network, more and more enterprises realize that, in addition to rely on the device itself and the reliability of the network architecture, to ensure the whole system run efficiently and reliably through monitoring the performance of network servers is important equally. In this paper, an enterprise-level application management platform based on the CPMM model was proposed. This platform integrates JMX (Java Management Extensions), SNMP (Simple Network Management Protocol) and WMI (Windows Management Instrumentation) to monitor the running state of all the servers in monitoring range, and judges whether the system has a poor performance or other problems according to the monitoring information and informs the fault to user. The platform provides a comprehensive solution for enterprise to ensure the whole system run efficiently and reliably, which makes administrators not need to prepare different monitoring tools for different IT resources.

Keywords: Performance Monitor; Server; Real-Time; Enterprise Application System

企业级应用系统综合管理平台的研究与实现*

盛羽¹, 蒋忠新², 董苹苹¹, 袁佳¹

¹中南大学信息科学与工程学院, 长沙
²湖南警察学院, 长沙
Email: vlab@163.com

收稿日期: 2013年5月2日; 修回日期: 2013年5月16日; 录用日期: 2013年5月28日

摘要: 随着国内企业网络规模的扩大, 越来越多的企业认识到, 除了要依靠设备本身和网络架构的可靠性之外, 通过对网络中服务器的性能监控来保证整个系统高效、可靠地运行一样重要。本文设计并实现了一种基于CPMM模型的企业级应用系统综合管理平台, 平台通过JMX(Java Management Extensions, Java扩展管理)、SNMP(Simple Network Management Protocol, 简单网络管理协议)和WMI(Windows Management Instrumentation, Windows管理规范)等技术对监控范围内所有服务器的详细运行情况进行监控, 并根据监控信息进行判断, 当性能变差或存在其他问题时及时预警, 告知可能会发生的故障。它为企业提供了一个综合的解决方案, 使管理员不必再为每个IT资源准备不同的监控工具来保证整个系统高效、可靠地运行。

关键词: 性能监控; 服务器; 实时; 企业应用系统

*基金项目: 湖南省科技计划重点项目“网络运行安全监控技术研究”(2011GK2008); 国家自然科学基金面上项目“难解问题的核心化技术及其应用研究”(61073036)。

1. 引言

随着企业应用需求复杂性的增加，企业网络规模的不断扩大，这导致网络故障率提高、可管理性下降。在中等规模以上的企业网络中，仅仅靠人工利用传统方式来管理网络是不切实际的^[1]。企业级应用系统综合管理平台为企业提供了一个综合的解决方案，实现包括实时查看整个系统的状态、检测系统可能出现的瓶颈、并通过邮件等方式向管理员发出告警等功能，以保证整个系统高效、可靠地运行。

在传统方式中，可以通过查看每个服务器的日志信息或运行相应的命令来获取服务器自身的一些参数，从而判断服务器的运行状况，这种方式是完全被动的依靠管理员人工操作来得到零散的数据，需要大量的人工操作。目前，一些网络管理方案，如惠普公司的 HP Open View、QUANTA 公司的 ESM^[2]、加州理工学院的 MonALISA^[3]、IBM 公司的 Tivoli^[4]，采用了代理的监控方式，即在被监控端安装代理，并由代理执行监测任务^[5]。这些监控方式往往存在一下问题：1) 这些监控系统一般是针对某一个层次的软件系统的运行状态进行监控，无法实现对整个应用系统从操作系统到服务器的全方位监控；2) 这些监控系统往往需要安装代理到每个被监控的服务器上，安装和维

护过程比较复杂；3) 监控数据呈现单一，缺少异常情况的全方位报警机制。

针对以上问题，本文设计并实现了一个企业级应用系统综合管理平台，包含对操作系统、数据库、应用服务器和 Web 服务器的全方位监控。该平台能自动的实现监控功能，并从监控范围内采集所需要的数据，这些数据通过网络集中到后台数据库中后再进行一系列的后续处理工作。对数据的采集实现的手段大致包括 JMX、SNMP 和 WMI 技术等。为企业提供了一个综合的解决方案，使管理员不必再为每个 IT 资源准备不同的监控工具。

2. 系统架构设计

企业级应用系统综合管理平台实现了对操作系统，应用服务器，Web 服务器和数据库的监控。提供对这几种系统及应用的性能数据的获取，例如，在应用服务器方面，可以获取其堆栈使用信息，线程池信息，JDBC 连接池信息等。支持对性能数据的存取，然后通过图表的形式表现出来，为管理员提供对所监控内容简单直观的理解。并且期望通过对历史数据的分析实现对其性能的评估。系统采用了基于 B/S 的无代理方式，整体架构如图 1 所示。

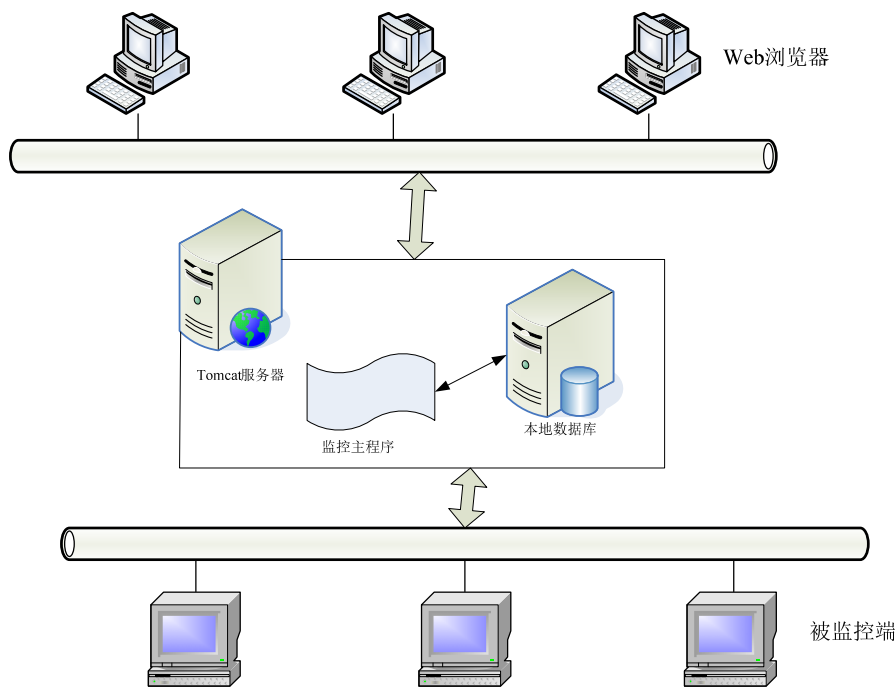


Figure 1. The system architecture
图 1. 系统整体架构图

系统设计采用了 CPMM(CERNET performance monitoring model)模型。CPMM 借鉴了 ISO 层次结构思想,是网络性能监控的一个整体解决方案,该模型已成功运用于 CERNET(China Education & Research Network, 中国教育科研网)全国主干网络的性能管理。如图 2 所示,系统分为 3 个层次,分别为数据采集层、数据管理层和数据表示层,各个层次核心线索是数据。数据采集层收集被监控服务器的性能数据,并将收集到的数据存储在后端的数据库中。数据管理层包括数据的存储管理和数据的格式化,性能监控所采集的数据量非常大,对包含一个应用服务器、一个 Web 服务器和一个数据库的系统进行每五分钟一次的采样,一天采集的数据量约为 20 M,因此对数据存储管理提出较高的要求。数据格式化的主要目的是便于数据表示,同时对数据进行一定程度的压缩。数据表示层的功能是显示性能监控的结果,包括实时数据、预警信息和对历史数据生成的统计报表。

3. 系统实现方案

3.1. 数据采集

数据采集是整个系统得以实现的基石,数据采集部分主要包括了数据采集技术的选取和数据采集系统的设计。

本平台涉及的应用与系统广泛,针对不同应用或系统采用不同的技术获取性能数据。对数据的采集实现的手段大致包括 JMX、SNMP 和 WMI 技术等。

1) 基于 JMX 的数据采集

JMX 是 SUN 创建的一套规范,是一个为应用程序、设备、系统等植入管理功能的框架。它提供应用程序接口,可扩充对象和方法的集合体,可以跨越一系列异构操作系统平台、系统体系结构和网络传输协议,灵活的开发无缝集成的系统、网络和服务管理应用。当今的主流应用服务器都实现的是 JAVA 内核,例如: Tomcat、Jboss、weblogic 和 websphere 等。它们都实现了 JMX 规范,所以可以方便地利用 JMX 在应用程序服务器里搜索管理模块,并通过它们得到配置信息和运行状态的监控信息,从而实现对应用服务器的性能监控。

2) 基于 SNMP 协议的数据采集

SNMP 设计简单,易于实现,且 SNMP 已经得到了相当广泛的应用,现有的各种网络设备都提供了对 SNMP 的支持。SNMP 使用户能够通过轮询、设置一些关键字和监视一些网络事件来达到网络管理的目的。它是一个应用级的协议,建立在 UDP 之上,在 SNMP 应用实体间通信无需建立连接,因而降低了系统开销。

3) 基于 WMI 的数据采集

WMI 是可伸缩的系统管理结构,它采用一个统一的、基于标准的、可扩展的面向对象接口。WMI 为您提供与系统管理信息和基础 WMI API 交互的标准方法。WMI 主要由系统管理员和管理员用来访问和操作管理信息。

4) 其他方法

Apache 服务器中可以使用的功能模块很多,这些模块工作的状态和性能是通过 mod_status 参数来记录

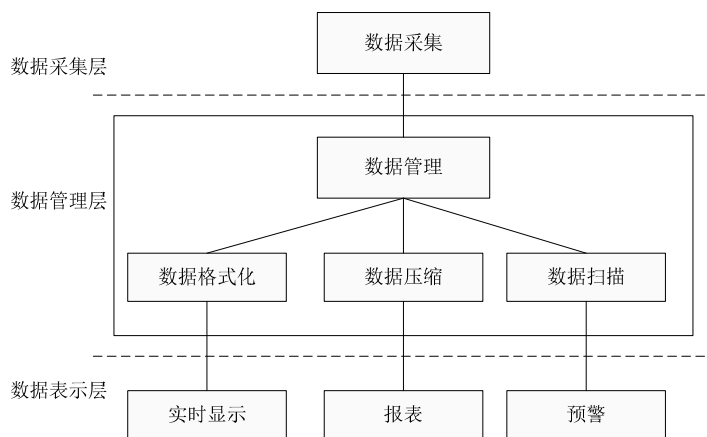


Figure 2. The system model
图 2. 系统模型图

的，可以通过该模块检查服务器当前的运行状况。数据库服务器则可以通过查询系统表来获取其性能参数。

以上针对各种不同的应用需求提出了相应的数据采集方案，而本系统需要一个统一的数据存储和呈现，因此各数据采集工具需要向数据管理模块提供统一的接口支持，而不同的系统或服务要监控的性能指标不同，需要用不同的技术手段来分别实现对他们的监控，因此，如何解决这一矛盾是系统设计的关键。

本平台采用了面向对象的设计方法，面向对象技术具有封装性和多态性，可以有效地解决上述问题。如图 3 所示，系统采用桥接 + 反射 + 单例的设计模式：抽象类 **Monitor** 继承 **Thread** 多线程类，监控不同服务器的各个监视器继承 **Monitor** 并实现 **Monitor** 类的虚函数，这样所有监视器都可以看成 **Monitor** 类，每新建一种监视器可以通过反射机制依赖注入找到相应的类以及方法；由于所有监视器都是 **Monitor** 类，只需在 **Composite** 组合类中建立一个链表即可；同时

由于对于整个系统来说只维护一个监视器链表，因此使用单例设计模式，使得 **Composite** 类中只有一个监视器链表 **MonitorListInstance**。

此外，每个监控类都实现了多线程的机制，使用户可以同时监控类的多个实例独立的进行监控。其次，数据采集频度可以由管理员自行设定，在与被监控的服务器建立连接的时候可以指定轮询时间，即多长时间进行一次数据采集。在建立连接后管理员也可通过编辑监视器来更改轮询时间。

3.2. 数据管理

本平台数据的存储和管理所面临的主要问题是：海量数据的存储和维护；高效的数据检索和处理。

为了提供历史报表并对系统性能进行评估，需要存储大量的性能数据。本平台采用数据库技术，利用数据库系统可以较好地解决大量数据的维护问题。选取 **MySQL** 作为后台数据库，数据库的设计如图 4 所示。

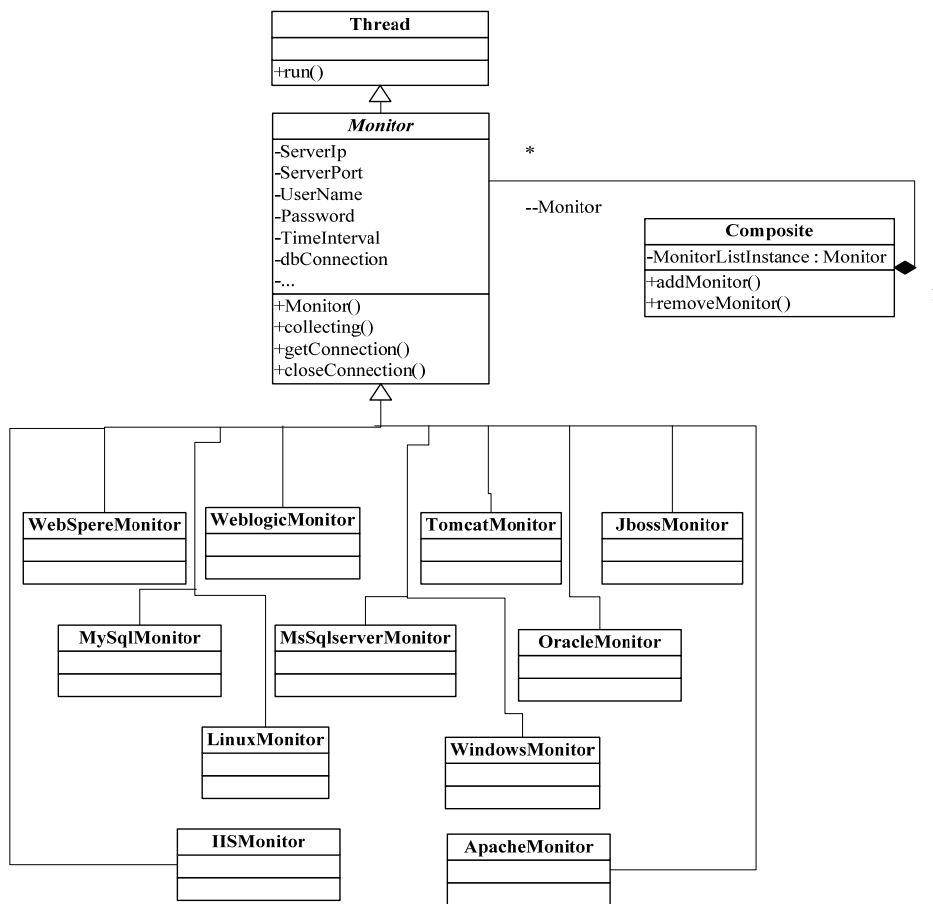


Figure 3. The overall design pattern
图 3. 总体设计模式图

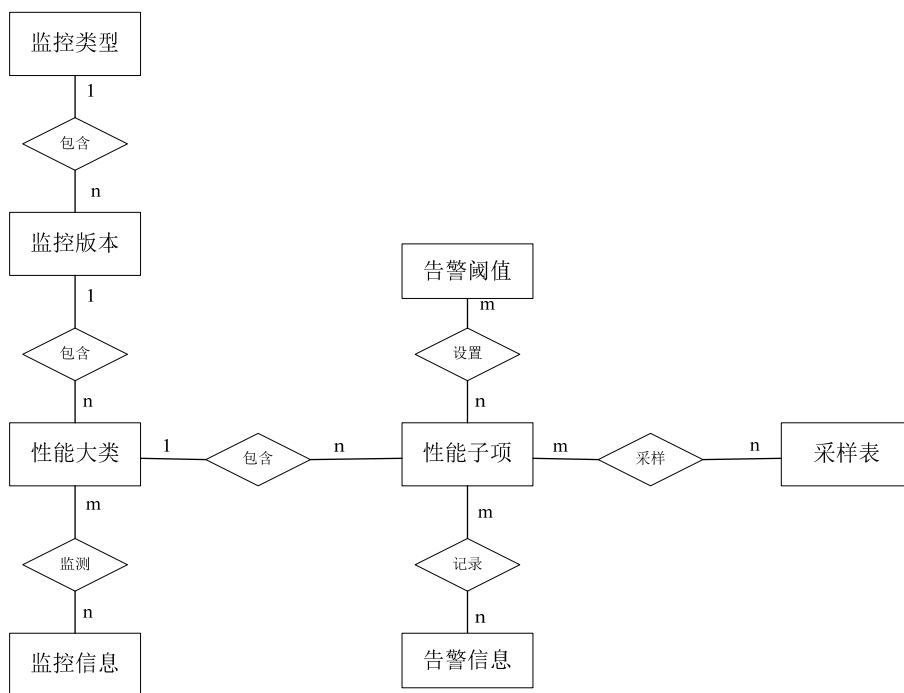


Figure 4. ER map: The relationships among the entities
图 4. 实体间的关系 ER 图

监控类型表表示系统可以监控的不同的信息资源，如应用服务器、Web 服务器、数据库服务器和操作系统等；监控版本表表示目前本系统可以支持的监控类型中服务器的不同版本，如应用服务器包括 WebLogic、Websphere 等，Web 服务器包括 Apache、IIS 等；性能大类表表示每类服务器需要监控的大类性能，比如对 WebLogic 而言，包括 JVM、Web 应用明细、EJB、servlet、线程池、数据库连接池；性能子项表表示每一个大类性能下要监控的具体属性；数据采样表存放监控性能的采样数据；告警阈值表记录了服务器的特定性能的特定性能子项的告警阈值，当监控到的性能不在阈值范围内时，系统会自动报警；告警信息表记录报警主机的 IP、报警时间、报警类型等信息。

数据库中的表按其属性可以分为两类：一是与时间无关的数据表，如监控类型、性能子项、监控信息表等，这些信息在设定后基本保持不变，可以认为是静态信息；二是与时间相关的数据表，即动态表，如每个采样表中都记录了采样的时间，又如告警信息表中发生告警信息的起止时间。这些动态数据的快速增长会导致检索性能下降和存储空间耗费过大等问题，因此对数据进行压缩处理是必要的。

数据压缩的依据是：当按不同的时间间隔考察性

能发展趋势时，所需数据采样的粒度不同，如我们需要至少 5 分钟每次的数据采样来画出一天的性能走势，而对于一个月的性能走势，我们则只需要一天一次的采样就足够了。为了更加准确的反映服务器的性能，在进行数据压缩的时候计算出极值、平均值等性能指标，这样使系统的平均响应时间和效率都提高了。

由于所采集数据数量庞大，并且无法保证监控人员能实时关注这些数据，所以在平台中实现了阈值告警的功能。系统的告警机制是基于阈值的设置的，如图 5 所示。由于绝大多数性能参数都可以体现为一个数值，故可以对其设定阈值，并通过数据扫描来检测当前的性能属性是否在阈值范围内。其中过滤器是用来对告警信息类型和优先级分类，然后通过发送邮件或发送短信的方式来通知管理员。

4. 系统运行实例

数据表示是性能监控功能的体现。数据表示包括了实时数据显示、告警信息和统计报表。实时数据显示是将最近一次采样的性能数据以图表的形式呈现给用户(管理员)，系统提供了饼图、直方图、折线图三种图形表现方式，图 6 为被监控的 Windows 服务器

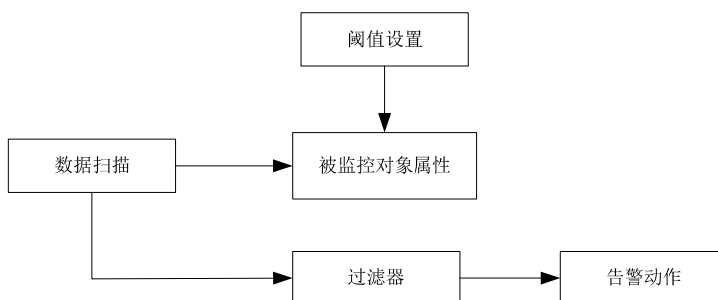


Figure 5. The alarm mechanism based on threshold
图 5. 基于阈值的告警机制



Figure 6. The details for real-time performance of a Windows server
图 6. Windows 服务器实时性能明细

的实时性能数据。当采集的性能数据超出阈值范围时,系统会通过邮件或短信告知管理员,管理员也可通过浏览器查看历史的告警信息及产生告警的原因分析。按用户需求将历史数据生成统计报表,如日报表、周报表、年报表、对比报表趋势报表、可用性报表、故障分析报表。

5. 结语

1) 针对现有监控软件无法实现应用系统进行全方位监控的问题,设计并实现了一个企业级应用系统综合管理平台。

2) 该平台提供了对企业的操作系统、数据库、应用服务器和 Web 服务器的综合监控,对采集到的性能数据做了后期处理,并将数据以友好的界面呈现给用户。

3) 该平台同时提供了了告警和报表的功能,方便

管理员判定企业级应用系统的工作状态,检测系统可能出现的瓶颈,从而保证整个系统高效、可靠的运行。

参考文献 (References)

- [1] 刘东红,郭长国,王怀民等. 监控使能的分布式软件系统构造方法[J]. 软件学报, 2011, 22(11): 2610-2624.
- [2] H. Sun. An integrated network performance monitor system. Third International Symposium on Intelligent Information Technology and Security Informatics, Jinggangshan, 2-4 April 2010, 88-91.
- [3] Monitoring Agents using a Large Integrated Services Architecture, 2011. <http://monalisa.caltech.edu/monalisa-.htm>
- [4] P. Sun. Structural analysis of the remote performance monitoring system used in modern Agricultural machinery. Intelligent Computation Technology and Automation (ICICTA), 2011, 2: 189-193.
- [5] L. Chen, W. Jian, Y.-B. Li, et al. Research of performance monitoring for spatial information system. Future Information Technology and Management Engineering (FITME), 2010, 2: 104-106.