



南宁供电局 J2EE 应用服务中间件监控系统的设计与实现

吴一鸣, 田永涛

(南宁供电局信息中心 南宁 530031)

摘 要: 电力信息系统中广泛采用 J2EE 架构, 监控 J2EE 应用服务中间件是信息运维工作的重要任务。研究对象为 Tomcat, 采用 JMX 技术实现监控系统核心功能。详细论述了 J2EE 应用服务中间件监控系统总体设计、功能模块以及系统实现。开发成果已投入试用, 运行状况良好, 是电力信息运维工作自动化系统化的成功案例。

关键词: 电力信息; J2EE; Tomcat; JMX

doi: 10.3969/j.issn.1000-0801.2013.11.023

Design and Implementation of J2EE Application Service Middleware Monitoring System of Nanning Power Supply Bureau

Wu Yiming, Tian Yongtao

(Information Center, Nanning Power Supply Bureau, Nanning 530031, China)

Abstract: J2EE architecture has been widely used in power information system, monitoring J2EE application service middleware is an important task in information operation and maintenance. The object of study is Tomcat, using JMX technology to achieve the core function of monitoring system. A discussion was made in detail about the design, function modules and system implementation of J2EE application services middleware monitoring system. Development results have been put into trial operation in good condition, which is an automatic and systematic successful case of power information operation and maintenance.

Key words: power information, J2EE, Tomcat, JMX

1 引言

J2EE(Java 2 platform enterprise edition)作为当前主流的 Web 服务架构, 广泛应用于电力行业的各个信息系统中。南宁供电局内, 包括生产 MIS(management information system, 管理信息系统)和营销 MIS 等多个信息系统都采取了 J2EE 架构。

在 J2EE 架构中, 应用服务中间件作为应用服务程序(服务器程序)的容器, 是用户业务请求和应用服务程序服务处理的中继以及桥梁。一方面, 应用服务中间件接收用户请求并屏蔽浏览器之间的请求差异; 另一方面, 应用服务中间件将应用服务程序请求处理结果响应请求客户端。

应用服务中间件是 J2EE 信息系统正常工作的保证。常见的 J2EE 应用服务中间件包括: Tomcat、WebLogic、JBoss 和 WebSphere 等^[1]。

正是由于应用服务中间件在信息系统中的关键位置, 当应用服务中间件出现内存溢出、进程僵死类似异常情况时, 轻则服务器超时响应用户请求, 重则信息系统崩溃无法访问, 均会导致系统服务异常, 影响用户工作。

应用服务中间件监控是电力行业信息工作人员日常系统运维的重要任务。在实际工作中, 采用人为方式监控信息系统应用服务中间件运行状况既不现实也不可靠, 以系统级别的自动化程序监控应用服务中间件是最好的解决方式。南宁供电局信息中心针对 J2EE 架构的应用服务



中间件自主设计并实现了监控系统,并在南宁供电局生产 MIS 中成功应用该监控系统。

2 系统简介

南宁供电局生产 MIS 架构体系可以分为 4 个层次,如图 1 所示。从上至下依次为负载均衡器、应用服务中间件、应用程序和系统数据库。负载均衡器是整个系统的对外接口,用户在以太网内发起服务请求,到达负载均衡器,负载均衡器根据指定算法,将用户请求分流至应用服务中间件。应用服务程序是用户请求的处理单元,处理用户请求并生成响应结果。系统数据库位于数据持久层,提供数据服务。

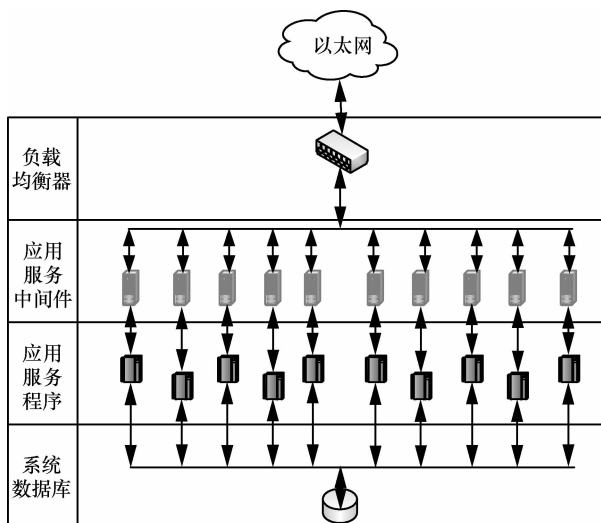


图 1 生产 MIS 结构

南宁供电局生产 MIS 采用的 J2EE 应用服务中间件是 Tomcat, Tomcat 对于其运行载体的计算机物理内存并非全数使用,为了充分利用服务器计算机的物理内存资源,生产 MIS 里采用的是 Tomcat 集群模式,即同一台服务器计算机上开启多个 Tomcat 应用服务中间件,提供无差别的系统服务。

南宁供电局监控系统的监控对象是生产 MIS 的应用服务中间件,获取应用服务中间件的各种实时运行参数,根据参数判断健康状态或警戒状态,在此基础上延伸高级应用。

在生产 MIS 新的架构体系中,监控系统位于负载均衡器与应用服务中间件之间;监控系统对下层应用服务中间件实时监控,对其上层负载均衡器提供接口,反馈应用服务中间件运行状态,以便负载均衡器优化分流用户请求。监控系统亦可以作为生产 MIS 对外接口,凭借其收集整理的应用服务中间件运行状态数据,导流以太网用户服务请求,集成负载均衡功能。生产 MIS 集成监控系统结构如图 2 所示。

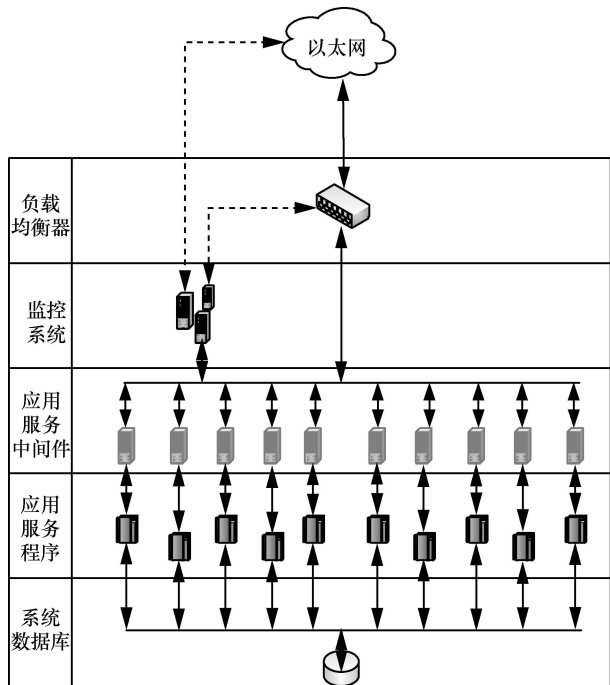


图 2 生产 MIS 集成监控系统结构

3 南宁供电局 J2EE 应用服务中间件监控系统设计

根据软件设计中功能聚合的原则,本次南宁供电局开发的 J2EE 应用服务中间件监控系统采取的是模块化模式,如图 3 所示。整个系统从总体上划分为:监控数据采集模块、监控数据处理模块、监控数据展示模块、监控控制模块、应用服务中间件控制模块、对外接口模块以及系统基础类库模块。

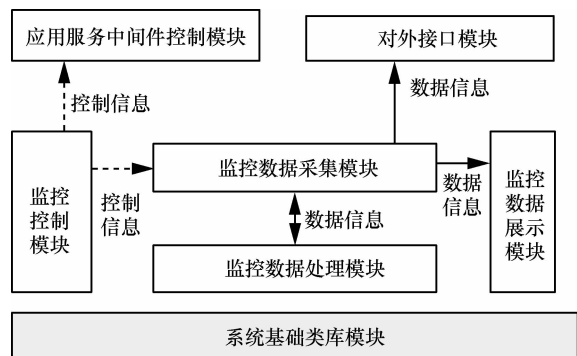


图 3 应用服务中间件监控系统结构模块

各个模块功能说明如下。

监控数据采集模块:此模块是整个系统的核心,采集模块获取到监控目标应用服务中间件的实时运行状态数据是监控系统运行的基础。通过对采集监控的数据分析以

及处理,系统做出正确的响应。

监控数据处理模块:此模块对监控数据结构化处理,并实现监控数据持久化。在此模块数据处理成果之上,可以实现应用服务中间件运行状态统计分析,进而实现中间件运行状态趋势预测以及对策预防等高级功能。

监控数据展示模块:电力信息工作运维人员通过此模块可以实时了解应用服务中间件的运行状态。展示模块是监控数据采集模块的“显示器”。

监控控制模块:运维人员人机操作的接口,操作人员通过控制模块将预设指令输入系统,完成各种系统功能。

应用服务中间件控制模块:系统通过此模块控制应用服务中间件,可对应用服务中间件实现启动、终止或者重启操作。

对外接口模块:为第三程序提供特定格式的应用服务中间件运行状态数据,在南宁供电局生产 MIS 中,负载均衡器可以接收监控系统的监控反馈,以实时更新分流策略。

系统基础类库模块:包括网络接口类、系统日志类、信息封装解析类等供系统其他模块直接调用的类库。

通过对各个模块功能分析可以看出,监控数据采集模块是整体系统开发的重点,是系统运行的数据之源,是系统功能实现的基础。南宁供电局 J2EE 应用服务中间件监控系统开发小组将此模块设为重点攻关对象。

4 南宁供电局 J2EE 应用服务中间件监控系统实现

4.1 监控数据采集模块实现

监控数据采集模块的开发采用 JMX (Java management extensions, Java 管理扩展) 技术。JMX 是一个为 Java 应用程序、Java 设备、Java 系统等植入管理功能的框架, JMX 中定义了 Java 进程管理的体系结构、监控模式和应用程序接口以及服务。Tomcat 中间件基于 J2EE 标准, 其本身集成了 JMX 框架, 使用 JMX 技术可以实现对 J2EE 应用服务中间件运行状态数据的实时采集^[2]。

JMX 框架本身定义了一系列进程监控数据交互的接口, 监控程序接入应用服务中间件内置 JMX 框架的监控服务端, 以此形成监控数据传输通道, 在通道内传输符合 JMX 框架定义格式的应用服务中间件运行状态数据。由此可见, JMX 框架可以看成一系列协议的集合, 包括基于 TCP/IP 的网络连接协议以及传输消息格式协议, JMX 技术

的体现就在于以 JMX 框架协议交互。

南宁供电局 J2EE 应用服务中间件监控系统开发小组经过对 JMX 框架的深入研究, 了解了其中的监控连接和传输传输的细节, 解析出网络连接以及传输消息格式的协议规范, 将 JMX 框架的网络连接协议和传输消息格式协议以类的方式在系统基础类库模块中实现。监控数据采集模块通过实例化这两个基于 JMX 框架的类, 调用类的方法, 实时获取到应用服务中间件的运行状态数据。

通过 JMX 技术, 开发小组实现了对单个应用服务中间件运行状态的监控, 但是在实际应用中, 南宁供电局生产 MIS 采用的是应用服务中间件集群。为了使一对一的 JMX 技术扩展为一对多的应用来应对现实的应用场景, 开发小组采取了线程池技术。监控线程池工作如图 4 所示。

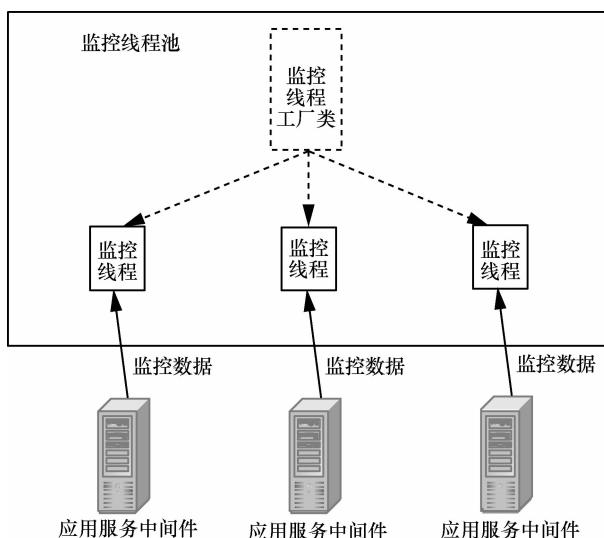


图 4 监控线程池工作

作为监控系统的子进程, 监控数据采集模块进程启动后, 等待来自监控控制模块进程的系统指令。当采集模块收到指向某个具体应用服务中间件的监控指令后, 通过工厂模式, 从应用 JMX 技术的线程池中新建一个线程, 以监控指令中解析出的参数实例化该线程, 线程开始对指定应用服务中间件的监控服务端以 JMX 框架网络连接协议发起监控请求, 应用服务中间件响应了监控请求后, JMX 框架网络连接通道建立, 应用服务中间件持续通过 JMX 框架网络通道发送以 JMX 框架传输消息格式封装的运行状态数据, 期间如果双方的 JMX 框架网络连接异常, 则监控线程保持对指定应用服务中间件发送监控请求, 直至采集模块收到停止对指定应用服务中间件监控指令, 采集模块释放该监控线程^[3]。采取线程池技术保证了线程工厂及时



生成和回收监控线程,监控线程之间并发运行互不影响之间的监控功能,线程池技术实现了监控数据采集模块一对多个应用服务中间件监控,监控数据采集模块功能开发完成。

以下是监控数据采集模块的3个监控线程在同一时间获取到的Tomcat中间件运行状态数据:

```
maxMemory:1398145024;
commitMemory:91029504;
usedMemory:33862408;
heap:37.19937658893539%;
maxMemory:1398145024;
commitMemory:102563840;
usedMemory:42737488;
heap:41.66915747304313%;
nonheap:175.77420874747304%;
nonheap:139.27205952029988%;
maxMemory:1398145024;
commitMemory:153288704;
usedMemory:76276240;
heap:49.759857060308896%;
nonheap:313.71510960663744%。
```

4.2 监控数据处理模块实现

监控数据处理模块的功能主要有两个:一是处理原始应用服务中间件监控数据,写入系统数据库;二是从系统数据库中读出历史监控数据,加以处理以实现高级统计功能。

从监控数据采集模块获取到的应用服务中间件运行状态数据,在监控数据处理模块汇聚,数据处理模块按照监控系统数据库表单定义字段,将监控汇聚数据生成结构化的运行状态记录,应用JDBC(Java database connectivity)技术写入指定的系统数据库(SQL Server 2005)表单,实现监控数据持久化。

数据是重要的资源,通过对监控数据的分析,可以得出许多有用的结论,作为判断决策的依据。监控数据处理模块的另一个功能是从系统数据库中应用JDBC技术抽取记录数据,作为高级统计应用的输入,经过高级统计处理后,可以得出中间件运行状态趋势预测以及预防对策等有效信息。

南宁供电局生产MIS中,某Tomcat应用节点某一时段堆内存使用率数据,见表1。

表1 Tomcat应用节点堆内存使用情况

日期时间	已使用堆内存 容量(KB)	最大堆内存 容量(KB)	堆内存使用 百分比
2013/09/22_16:15:23	289 244	1 040 512	27.79%
2013/09/22_16:15:24	232 953	1 040 512	22.38%
2013/09/22_16:15:25	293 710	1 040 512	28.22%
2013/09/22_16:15:26	294 372	1 040 512	28.29%
2013/09/22_16:15:27	295 122	1 040 512	28.36%
2013/09/22_16:15:28	296 840	1 040 512	28.52%
2013/09/22_16:15:29	242 761	1 040 512	23.33%
2013/09/22_16:15:30	250 081	1 040 512	24.03%
2013/09/22_16:15:31	252 146	1 040 512	24.23%
2013/09/22_16:15:32	254 691	1 040 512	24.47%

4.3 监控数据展示模块实现

监控数据展示模块是监控系统向系统用户传输应用服务中间件运行监控数据的单工通道,模块采用图形化技术直观地显示各类监控信息,以便系统用户实时了解应用服务中间件运行状态以采取正确的应对措施^[4],如图5所示。

4.4 监控控制模块实现

监控控制模块是监控系统人机交互的接口,系统管理员通过控制模块向监控系统发送预设的系统指令,系统收到指令后,解析指令以及指令参数,执行系统管理员指令,并将指令执行结果通过监控模块反馈系统管理员,完成一次人机交互。

在监控控制模块中,预设系统指令采用的是带参数的命令行模式,即指令字符串后拼接参数名和参数值。

新建监控线程指令:new -a 127.0.0.1 -p 10091 -dba 127.0.0.1 -dbp 20000。

监控系统解析指令类型和指令参数后,指令执行处理通过函数形式实现,函数形式保证了系统指令执行处理的可拓展性。

4.5 应用服务中间件控制模块实现

监控系统通过此模块远程控制应用服务中间件,远程控制操作类型当前主要有启动、终止和重启3种。启动操作实现的原理是,调用Tomcat批处理启动文件startup.bat;终止操作实现的原理是通过Tomcat监听端口号查询出该Tomcat进程的ID号,调用Windows操作系统命令taskkill终止该进程ID号的Tomcat;重启操作实现是将Tomcat调用终止操作后,再调用启动操作。

4.6 对外接口模块实现

对外接口模块取决于对外接口系统,针对不同的接口



图5 监控数据展示模块

系统,监控系统的对外接口模块亦有相应变化。在南宁供电局生产 MIS 中,对外接口系统是 Radware 负载均衡器,监控系统与 Radware 交互,实时更新应用服务中间件 Tomcat 集群中可用单体信息,保证 Radware 的分流服务均指向可用的 Tomcat^[5]。

Radware 与监控系统交互如图 6 所示。Radware 接口机制的实现是基于 TCP/IP 自定义网络传输协议交互: 对于一个 Tomcat, Radware 系统中可设定一个 IP 地址、TCP 端口和时间间隔作为接口属性, Radware 向指定 IP 地址的指定 TCP 端口以一定的时间间隔发送“detect”消息, 监控系统中该 Tomcat 运行状态正常时, 对外接口模块向 Radware 回发“succeed”消息, 如果该 Tomcat 运行状态非正常时, 对外接口模块回发“failed”消息, 网络传输交互完成。

5 结束语

5.1 系统开发小结

南宁供电局 J2EE 应用服务中间件监控系统开发完成后, 在技术上实现了以下几点突破。

- 深度了解了 J2EE 应用服务中间件运行机制以及技术体系, 应用 JMX 技术实现对 J2EE 应用服务中间件运行参数数据监控, 实时掌握 J2EE 应用服务中间件运行状态。

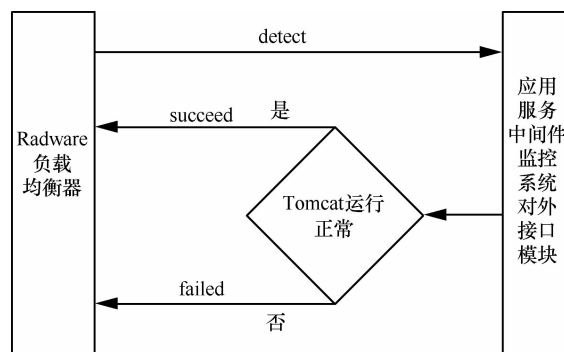


图6 Radware 与监控系统交互

- 监控系统内模块功能划分清晰合理, 模块间高聚合、低耦合, 体现了 MVC(model, view, controller) 模式软件设计典范。
- 监控系统对外接口设计规范、兼容性强, 系统间数据流转通畅, 数据共享程度高。在监控系统基础上与其他系统灵活集成, 功能对接。
- 实现了系统级别的自动化程序监控, 以信息化的方式提升供电企业的信息运维水平。

5.2 系统展望

南宁供电局 J2EE 应用服务中间件监控系统, 在技术上仍有以下拓展空间。

- 数据加密。无论是通信传输中的数据, 还是持久层存储数据, 都需要做到数据加密, 确保数据安全。

(下转第 151 页)

的关键,如 HTC 的 Sense UI、华为的 Emotion UI,而小米的 MIUI 更强调适合国人的使用习惯。

三星有专门负责软件方面用户体验的部门,完全从用户使用角度考虑,挖掘用户需求,从细节入手,持续改善 UI 设计,开发新的交互方式,最终只有一个目标——提高用户使用三星手机的便利性^[3]。而三星的 CEO 权五铨在就任讲话中提到,他们将会“特别专注于提供新的用户使用经验”,将会通过加强/改善软件、UI、设计和解决方案这几个方向达成。

Android 市场占有率越来越高,而 Google 实施越来越严格的控制,在手机同质化现象日趋严重的背景下,通过独特的设计、便捷的操作,从使用习惯上绑定用户,从用户直接体验方面建立品牌效应。

5 从分支走向独立

分支发展的自研操作系统利用上游操作系统 Android 的生态环境,确保新平台上市时有足够应用可供用户选择。为了降低上游操作系统的影响力,大力发展以 HTML5 为代表的 Web 应用生态系统,充分利用云端能力。平台实际承接了两个应用生态系统,分别是原生应用生态系统和 Web 应用生态系统。当 Web 应用生态系统不断发展并趋向成熟,不再仅作为原生应用生态系统的补充而是可替代时,则可以考虑将原生部分从自研操作系统中去除,成为独立、完整的自研操作系统。两个应用生态系统的对比如图 8 所示。

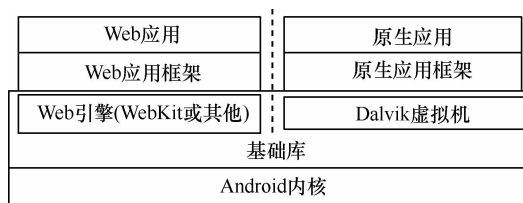


图 8 两个应用生态系统对比

6 结束语

自研操作系统应首先立足于应用平台这一智能终端操作系统的关键特征,将自有业务和能力部署到平台上,为广大的开发者所使用。同时,Web 化演进是目前操作系统发展的重要趋势,降低底层操作系统对生态系统的掌控,直接控制应用,建立自有生态系统。

[作者简介]



魏颖琪,女,中国电信股份有限公司广东研究院高级工程师、一级产品业务架构师,主要从事移动操作系统、移动互联网应用研发工作。

梁柏青,男,中国电信股份有限公司广东研究院高级工程师、政企客户开发部技术总监,主要从事移动终端、云计算、移动互联网技术研发工作。

(收稿日期:2013-08-25)

(上接第 135 页)

- 趋势分析。针对系统获取到的大量 J2EE 应用服务中间件运行数据统计分析,进而实现中间件运行状态趋势预测以及对策预防等高级功能。
- 机器学习智能故障处理。应用机器学习技术,对故障处理操作建模,分析每次故障处理控制信息,操作目标参数回归,最终实现自动故障处理。

参考文献

- 高张,康小军. 提高 Tomcat 服务器运行性能的研究. 计算机与数字工程,2008(10):153~154
- 孙仁鹏,何森. Tomcat 性能监视和调优研究. 软件导论,2012(12):56~59
- 林泊,周明辉,刘天成等. 一个 J2EE 应用服务器的 Web 容器集成框架. 软件学报,2006(5):26~28
- 田珂,谢世波,方马. J2EE 数据持久层的解决方案. 计算机工

程,2003(22):102~103

- 5 窦蕾,袁臻,刘冬梅. 基于构件的中间件技术 J2EE. 计算机科学,2004(6):98~102

[作者简介]



吴一鸣,男,南宁供电局信息中心助理工程师,从事电力信息系统运维工作。

田永涛,男,南宁供电局信息中心工程师,从事电力信息系统开发维护工作。

(收稿日期:2013-10-01)