

研究、探讨、传播

www.WebSpherechina.net 总第1期

WebSphere 中国

IBM WebSphere行业解决方案的专业纸质读物

封面报道

使用WAS搭建 高可靠性系统



免费阅读 审核赠阅



我们以创新的思维服务于不同行业客户

广州从兴电子开发有限公司
GUANGZHOU CONGXING ELECTRONICS DEVELOPMENT CO., LTD
www.congxing.com

和 WebSphere 一起拯救地球

题目有点吓人,但其实是我在看电视广告中产生的一个想法。

虽然最近我看电视的时间不多,但碰巧每回都能看到 IBM 的广告。说明在金融风暴之下,IBM 这次出招的决心狠,力度大。IBM 这次讲的不再是之前大家熟悉的“按需应变(ON DEMAND)”,而推出了新概念“智慧星球(Smarter Planet)”,着眼的不再是单个的业务、单个的企业,而是让物理基础设施和 IT 基础设施融合的智能电网、交通等等,遍布地球上的各个角落。



在大家心目中,IBM 是一家“硬件+软件+服务”的公司,它提供硬件、软件、架构理念、整体解决方案及咨询服务等等,渗透到我们这个星球上的各行各业。“智慧星球”这种超越一般客户公司而关注整个行业整个产业链乃至整个地球的理念,确实在职场中是超前的,体现出来的大气,环顾全球,恐怕也只有 IBM 有这个底气。当然,IBM 也为了在这中间赢得更广阔的发展空间。

从这个说起,为了实现这个更智慧的地球,在软件系统方面 IBM 还是少不了 SOA 的理念,同样也不可能缺少 WebSphere 下面的各种产品。怎样根据自己的实际情况,用好 SOA 的架构、用好 WebSphere Application Server、MQ、Message Broker 等等强大的产品、从中体现出智慧来,是摆在我们大家面前的难题及挑战。

那就让我们大家在这个平台上共同交流、好好学习 WebSphere 的各种知识和经验,虽然我不来自 IBM,但让我们自己开发和维护的系统更高效、更人性化、更节能环保,也是为了我们生活的地球更智慧的未来出的一份力。意义匪浅啊! ■

南航首席架构师: **彭海晖**



服务为本/资源为根
效率为主/客户为尊

<<<详情点击: www.digitalchina.com



中国最大的整合IT服务提供商神州数码公司为您量身定制解决方案。

作为IBM公司最重要的合作伙伴,神州数码系统科技公司IBM事业部整合集团各项资源,本着以客户为中心,以服务为导向的宗旨,为合作伙伴提供基于IBM p系列产品、IBM存储产品、IBM软件全线产品的最便捷的解决方案,最全面的项目方案咨询,最专业的产品技术培训。

了解更多IBM产品信息及解决方案,请致电:

北京: 010-62694234 / 广州: 020-87582246-8372 / 成都: 028-85459888-1978 / 上海: 021-22019408 / 武汉: 027-87222817 / 西安: 029-88342433
福州: 0591-83306303 / 济南: 0531-82382114 / 沈阳: 024-23582662 / 郑州: 0371-65825391 / 合肥: 0551-2834200 / 长沙: 0731-4161486-1361
杭州: 0571-87362179 / 长沙: 0731-4161486-1361 / 哈尔滨: 0451-86299696-3066 / 南京: 025-83351122-1219 / 深圳: 0755-82966699-8501

www.WebSpherechina.net



COVER STORY 封面报道

使用WAS 搭建高可靠性系统

为了满足移动公司BOSS2.0规范要求并适当超前,满足广东移动现状需求,广州从兴电子开发有限公司使用WebSphere Application Server 6 搭建集群服务器给移动建设了全省集中的BOSS系统,实现全业务多品牌的运营,实现对集团客户、数据业务支撑。

WebSphere MQ 消息通道的配置和维护介绍



WebSphere MQ 独特的安全机制、简便快速的编程风格、卓越不凡的稳定性、可扩展性和跨平台性,以及

强大的消息通讯能力,使得它在银行、电信,还是在交通运输、政府机关等各行各业,赢得了很高的市场份额。

专家专栏 P37

卷首语

- 1 和 WebSphere 一起拯救地球

新闻

- 4 IBM: 经济危机下突显BRM工具价值
4 IBM投入力量研发各种环保解决方案
5 IBM 欲扩大台湾研发中心
5 IBM 将部署大规模产品升级计划

封面报道

- 6 使用 WAS 搭建高可靠性系统

解决方案

- 15 知识管理 & 企业门户解决方案
18 药品监督局协同办公平台解决方案
20 按需应变的企业应用中间件平台
23 企业综合信息平台解决方案
25 WebSphere Business Integration Messenger Broker 在企业信息交换中的应用

- 26 浙江移动网络投诉处理综合平台解决方案

- 28 供应链管理系统解决方案

专家专栏

- 30 PA 与 WebSphere 的兼容性问题解决方案
34 WebSphere Edge Component 做网络负载均衡器的实施案例
36 WebSphere MQ 消息通道的配置和维护介绍
48 带您走进WebSphere sMash的奇妙世界

博客

- 50 WebSphere Cluster 调优过程
53 我和WebSphere的故事——在摸索中前进
54 韦氏紫血三国——DataPower 长板浴血
55 Tomcat 向 WebSphere 迁移的问题及解决方法
56 WebSphere 性能调优全讲解

有问必答

- 62 WebSphere 端口配置问题
63 如何监测WebSphere 6.1的性能?

WebSphere 中国
IBM WebSphere行业解决方案的专业纸质读物

电话: 010-68867107-806
地址: 北京市石景山区石景山路 23 号中融大厦 530 室 (100043)
邮件: chao_huang@twigroup.com.cn
广告: 010-68867107-806 联系人: 黄超

主 编: 黄超
副主编: 陈龙
专家顾问: 毛新生 甘荃 黄河
李俊杰 张东焕
设计总监: 肖晔

《WebSphere 中国》是一本审核赠阅的免费读物,由WebSphere开发与应用网站发起、协作创立并完成的第一本专注于研究、探讨、传播 IBM WebSphere 行业解决方案的专业纸质读物,服务于广大的中国 IBM WebSphere 用户。
申请免费阅读,请访问: www.WebSpherechina.net

IBM: 经济危机下突显 BRM 工具价值

IBM 认为, 不景气的时候更适合推动商业规则管理 (Business Rules Management; BRM) 系统。原因即在于在此环境下, 企业能否因应法规、市场变化, 以及竞争对手策略实时调整业务策略, 将是存续的重要条件。

IBM 大中华地区软件事业处 ILOG 营运总监刘荣耀指出, 弹性的商业规则管理系统帮助企业渡过不景气的关键工具。市场不景气除让金融保险业客户愈发重视风控管外, 半导体与制造业客户也比以往更重视库存等资源的优化管理。这也是 IBM 会于今年 (2009 年) 力推整合 IBM WebSphere BPM 与 ILOG BRM 的原因。而透过业务流程管理 (Business Process Management, BPM) 与商业规则管理系统, 某金融保险业者即将新产品的推出时间从原先的 9 个月缩减成 3 个月。



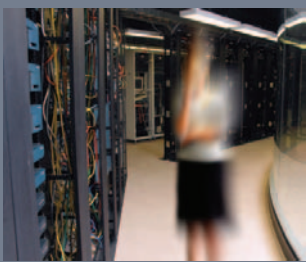
IBM 打造“智慧星球”开始第三次转型

日前, IBM 公司公布了 2008 年财报, 全年收入为 1036 亿美元, 较去年增长 5%, 利润大幅上涨, 增长超过了 18%。财报显示硬件部门的收入下降, 而最大的利润增长贡献部门来自软件和服务部门, 增幅分别超过了

11% 和 9%。IBM 从几年前开始进行全面转型, 从“硬件 + 软件”转型为“硬件 + 软件 + 服务”。而在国际金融危机的背景下, 以制造和硬件为重点的 IT 公司都遭遇重创, 服务成为产业利润增长的亮点。



IBM 将推新战略引领 IT 走出数据中心



IBM 公司计划帮助用户拓宽信息技术的应用范围, 从数据中心领域深入到管理物理环境, 云服务和虚拟资源。通过一系列应用工具来改善效率, 降低费用和削减人力成本。IBM 公司还在近日宣布了他们构建

最新基础架构的计划, 这个项目将把蓝色巨人的按需, 可适应性和自动化产品与全球服务部门多年积累的专业技术经验结合在一起。

IBM 公司计划帮助用户管理涉及物理, 虚拟, 数字和应用环境等领域的 IT 服务交付和性能, 将他们的软件和服务根据 7 个行业进行定制包装。公司日前在美国拉斯维加斯有 3000 人参加的 Pulse 2009 大会上透露了公司的最新服务和升级产品, 并对外公布了相关的战略方向。

IBM 投入力量研发各种环保解决方案

环境问题已经成为全球所面临的最严重的问题之一, “拯救地球”的呼吁反映了人类面临的环境问题的严重性, 首当其冲的就是全球变暖和能源短缺。中国环保的任务更加严峻。据有关资料表明, 2006 年中国占全球 GDP 的 5%, 却消耗了全球 15% 的能源。节能减排已经成为政府、企业、社会共同关注的热点问题。

IBM 公司一直关注中国的发展战略, 并积极配合国家的重大政策, 做出自己应有的贡献。目前, 创新、节能降耗、发展服务业、新农村建设、和构建和谐社会是中国社会发展面临的重大课题。IBM 公司希望结合自身的业务能力和全球资源, 在这些问题上提供自己的帮助。



IBM、亚马逊强强联手推进云服务

近日, IBM正在利用亚马逊的弹性计算云服务 EC2 为自己的 DB2、Informix Dynamic Server、WebSphere Portal、WebSphere sMash 聚合应用工具以及 Lotus Web Content Management 内容管理软件打造“亚马逊机器映像 (Amazon Machine Images)”。独立软件开发商只需每小时缴付亚马逊 10 美分的使用费就可以利用这些映像开发和测试运行在 IBM 软件上的应用。



IBM 欲扩大台湾研发中心



销和人力资源部等非理工科人才。

IBM 准备扩大的是其位于台北南港园区的“系统暨科技研发中心”,也是IBM公司在海外唯一的服务器产品研发中心,目前该研发中心所研发的x86服务器已经占IBM全球出货量的一半。随着研发项目的增加,该研发中心现有250名员工已不足以满足业务需要。IBM因此表示,将该研发中心团队扩编到400人的规模。

这次IBM的公开征才和美国总公司逆势而行,IBM公司希望新进员工有积极,进取的人格特质,更重要的是要富有创意。并且在本次征才活动里拨出其中40%的缺额,给担任工程师未满18个月的年轻工程师,希望能替公司找齐各种类型的人才。

据报道, IBM 公司准备在台湾征集人才近百人,主要为扩大其在台北的服务器产品研发中心储备人才。其中,软硬件研发工程师占六成,IBM 也将不定期寻找营

IBM 将部署大规模产品升级计划

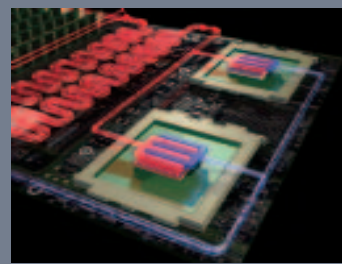
IBM 日前宣布,将对现有的产品及服务进行大规模升级,以建立那些高智能系统并妥善处理由此而产生的大量数据。IBM 公司的此项计划,将包括为美国电网安装数百万个感测器及对美国的医疗数据进行电子化等一系列内容。



据悉, IBM 此次所宣布的大规模升级计划具体包括: 为 XIV 及 DS8000 等系列的存储设备进行更新, 以更好地储存千兆位的信息, 并减少数据的重复性并自动为储存驱动提供加密保护; 开发并完善 Tivoli 能源管理软件, 以提供建筑物空调系统及城市街灯系统等的能源自动化管理; 成立新的 IT 技术咨询及服务团队, 从而为广泛行业的客户提供更加专业的咨询, 而这些行业将包括基础设施建设、通信工程、化学加工等等。

IBM 全球首次发布动态基础设施策略

2月8日-12日, 以 Pulse 2009 为主题的 IBM 服务管理大会在美国拉斯维加斯举行。作为 IBM Tivoli 软件连续第二届举行的盛会, 吸引了来自世界69个国家和地区数千名用户代表、合作伙伴和 IBM 员工参加。



作为实施 IBM 在 2008 年 11 月首度披露的“智慧的地球” (Smarter Planet) 的具体行动之一, 在本次大会上, IBM 软件集团 Tivoli 软件总经理 Al Zollar 首次针对服务管理提出了“动态基础设施” (Dynamic Infrastructure) 的新思路和新提法。他宣称, 针对当前业务环境的剧变和全球众多行业对于智能化的更多需求, IBM 在软件服务管理领域将致力于为用户打造一种“动态基础设施”, 帮助用户提升服务, 节省成本, 管控风险。这种全新的基础设施将具备三个重要特征: 集成物理和虚拟世界; 管理变化的信息; 避免无用功。



使用WAS 搭建 高可靠性系统

为了满足移动公司BOSS2.0规范要求并适当超前,满足广东移动现状需求,广州从兴电子开发有限公司使用WebSphere Application Server 6搭建集群服务器给移动建设了全省集中的BOSS系统,实现全业务多品牌的运营,实现对集团客户、数据业务支撑。

广州从兴电子开发有限公司 廖锦波 / 文

项目介绍

广东移动的网络发展规模不断扩大。至2005年底,GSM用户容量为5600万户,GPRS用户容量也将达到1000万户,即将进行的3G网络建设进一步增加了网络容量。广东移动综合业务运营支撑系统(BOSS2.0)是在集团公司BOSS2.0规范指导下,适当超前并在广东移动需求要求下进行规划建设的,该系统具有以下特点:

- 支撑超过6000万用户业务受理和准实时计费
 - 系统容量160T
 - 系统每月处理话单600亿张、其中清单200亿
 - 计费子系统是全球最大、最高效、最精确的移动计费系统
 - 集中化支撑体系:实现地域集中、系统集中,形成全省集中的BOSS支撑体系;
 - 基于J2EE的软件结构和基于WEB

的浏览器界面,建立面向服务的架构;

- 双中心结构,实现数据容灾和关键业务容灾,增强支撑网的业务连续能力;

解决方案

- ISV应用程序:从兴BOSS2.0
- 中间件:IBM WAS 6.0.2
- 硬件:16台IBM P595、3套DS8300存储阵列、2套DS4800存储阵列和PC Server若干

随着各地市数据割接完成,至2009年初,GSM用户容量增加为近8000万户,GPRS用户容量也达到1000多万户。

- 系统支撑超过8500万用户业务受理和准实时计费
 - 系统扩容为容量180T
 - 系统每月处理原始话单1420亿张、其中清单300亿

通过上面的数据可以看到,这几年来,BOSS系统支撑的数据量越来越大,

因此BOSS系统也进行了好几次扩容,WAS实施架构也优化了几次方案。

文章介绍的内容是BOSS项目中的服务开通平台子项目,是BOSS中的重要组成部分,相比BOSS主项目,这个集群方案经历数次变更,更有代表性。

- 业务大部分是并发请求,并发要求在3000以上
 - 24小时不间断稳定运行
 - 大数据量业务操作
 - 受BOSS主项目影响,不能占用太多数据库资源

搭建高可靠性系统

服务开通平台需要极强的稳定性与可靠性,使系统能够长时间安全、稳定地正常运转。要求应用系统能够有效规避任何单点故障,这些故障范围包括:应用程序错误、数据库系统故障、网络端口故障、网线接触故障、磁盘系统介质故障、系统瘫



痪等。万一上述故障发生，如何保障信息系统能够继续提供正常的服务。在任何应用环节故障的情况下，高可靠性系统应确保应用系统平稳运行。经过软、硬件产品的综合考察，我们采用了多种方案来保障系统的稳定。

1、使用6台P595服务器，共130CPU建立Oracle集群。基于移动用户数现状的分析，建立三个区域RAC集群，广州、深圳、东莞。每个RAC集群处理2千万用户。每个集群都有两个实例节点，一部分地市接入一个节点，另一部分地市接入另外一个节点。

2、服务入口使用F5 Networks的负载均衡器，F5采用了智能负载均衡技术，双机HA冗余做服务器负载均衡，采用F5特有的负载均衡算法和健康检查方法保证业务负载实时高效均衡。保证业务处理高效，有效性。使自动故障恢复得以实现，服务的时间可以延长，24×7可靠性和持续运行成为可能。

3、使用6台PC Server构成WAS集群服务器，每个机器上都同时部署HTTP Server与WAS进程，通过应用负载均衡技术，可以利用多台服务器同时为大量用户提供服务。当某台服务器出现故障时，负载均衡服务器会自动进行检测并停止将服务请求分发至该服务器，而由其他工作正常的服务器继续提供服务，从而保证了服务的可靠性。

4、使用6台PC Server部署WebSphere MQ，构成一个集群。MQ是一个数据传输的基础设施，是业界最成熟提供端到端数据传输的中间件，经过各个行业数量庞大的客户群体实际验证，具有可靠性。组成集群后进一步避免单点故障而影响系统。

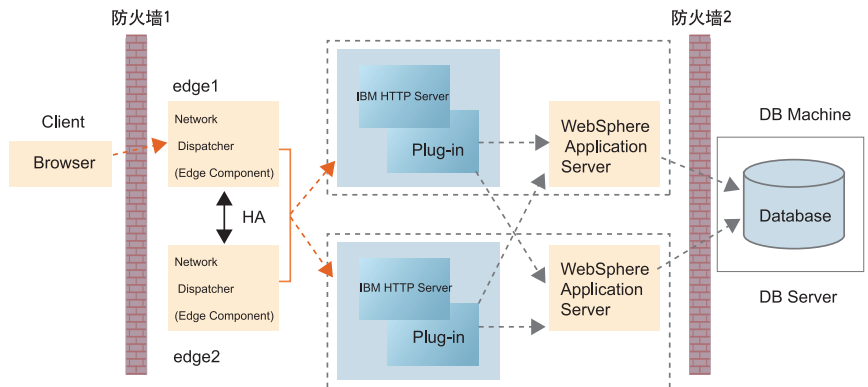
5、服务开通平台还有部分C++后台进程，用于给移动网元设备发送请求，为了提高这部分进程的可用性，我们按数据分区的规则，把6台服务器也划分为三个区域：广州、深圳、东莞，每个区域二个机器，两两互备。

6、当系统出现不可预知的重大故障后，可以紧急启动容灾服务器，接手整个平台的业务请求。

WAS 集群简介

WAS 集群由一组应用服务器组成，每个服务器上部署了同样的应用程序。通过集群可以实现可扩展性（服务更多客户，提高吞吐量），负载均衡（平衡负载资源，使资源得以有效利用），高可用性（提供故障恢复和补偿机制，在关键性业务中提供容错功能）。

下面是一个典型的 WAS 集群拓扑图



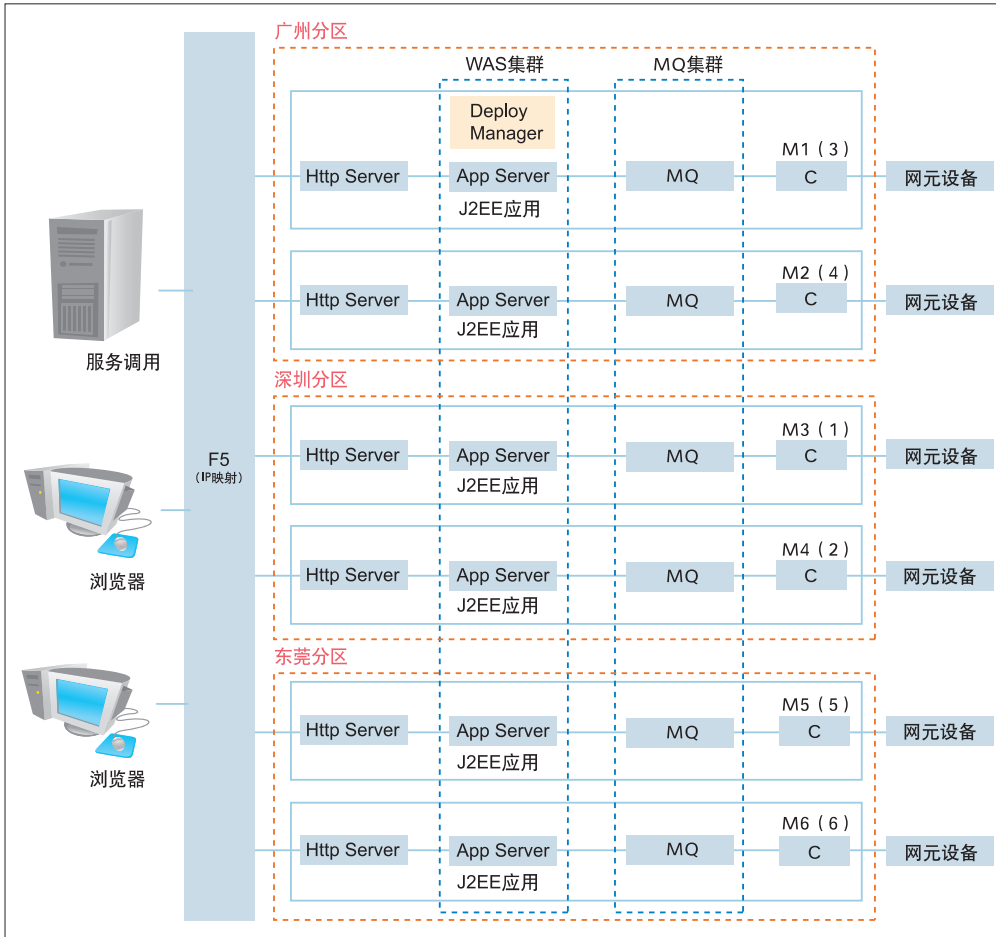
在集群环境中，先经过硬件负载均衡器F5之类的设备进行底层流量分流，再交给IBM HTTP Server进行业务请求分发，WAS plugin起到负载均衡的作用，在plugin-cfg文件中可以定义每个服务器的权重，根据权重值，plugin决定将一个请求定向到哪个服务器。

实施方案

服务开通平台架构拓扑图1是服务开通平台在BOSS项目开始时按上面所说的思路设计的拓扑图，整体结构与上面的WAS拓扑图相比，几乎一致。先经过F5进行负载均衡后，再进入WAS集群网络。整个集群由6个服务器组成水平集群，每个服务器上运行一个WAS进程，为了提高可靠性，每个机器都部署一个HTTP Server，避免其中一个故障而导致停止响应。其中一号服务器上运行DM管理进程，它们一起组成了服务开通集群。同时，服务器群上还部署一个MQ集群作为WAS上的应用与后台进程间的通信中间件，后台进程再与外部设备进行连接。

WAS集群中的J2EE应用对业务没有分区，任何一个服务都可以接受请求。而后台进程不同，其中一号服务器与二号服务器组成一个业务分区负责广州地区业

务，三，四负责深圳，五，六负责东莞。两两互备，避免由于硬件或者意外原因导致服务无法响应请求。这个方案在实施后，随着各地市数据不断割接后全省数据中心，压力不断增加，服务开通平台由于压力大而且服务器上还有足够资源，后期每个服务器上WAS进程增加到两个。



服务开通平台架构拓扑图 1

第一次优化

在广州区域中心割接后，BOSS系统的压力一下膨胀了很多，原来这个方案的一些缺点很快就暴露出来

- 单一集群，重新部署应用会导致服

务器停止服务。

- 因为是并发请求，所以耗时的请求将会持续等待响应，而占用资源不放，当慢速业务请求多而响应不及时时，将会不断占用服务器资源，导致快速业务也无法

响应请求。

- 业务故障会相互影响，某个地区的接口设备响应速度慢会导致其它地区也无法接受请求。

因此，我们对集群方案进行优化，如图 2

在这个方案里，重点是把WAS的服务进程划分为两个集群，各自负责不同类型的业务，来避免原来方案中的一些缺点。

- 双集群，上载时可以先停一个，更新完了再去更新另外一个集群。

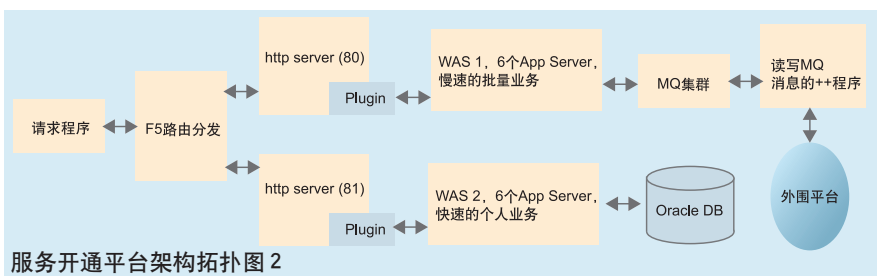
●批量和个人类业务分开部署，这样避免了慢速业务占用了快速业务的资源。

- 慢速业务还是会相互影响，这只能要求移动尽量提高这个业务设备的效率了。

第二次优化

在 07 年底，随汕头智能

网回割—>广州 2.0 割接—>佛山 2.0 割接—>深圳 2.0 割接的各地割接进程不断进行，系统的压力远大于最初预计，当全部地市割接完毕，现有的机器资源明显不足使用。按时间点交易量和所需资源情况如图 A



服务开通平台架构拓扑图 2

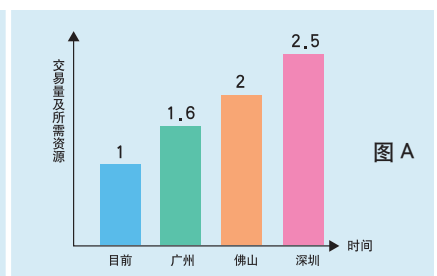


图 A



方案2 机器之间的消息交互非常频繁, 机器提供的性能有限, 随着移动公司统一数据割接的进行, 资源问题变得越发紧张. 为了进一步细化业务类型, 减少相互之间的干扰, 我们再进一步进行优化, 并提出方案3。

1. 因服务器的性能问题, 造成服务开通应用不够稳定, 所以此次扩容, 将服务开通迁移到 P595 应用主机上, 使用扩容后每台主机的第三个 LPart 分区, 共四个 LPart 分区, 每个分区 8 个 CPU, 24G 内存。

2. 每个服务器上新建4个app server, 分别称为 ai1、ai2、ai3、ai4 (i 表示第 i 个分区, 例 a32 表示第 3 个分区的第二个 server), 一共 16 个。

3. 把这16个WAS划分6个逻辑集群, 分别给3类业务使用, 每类业务两个集群, 大大提高可用性。

4. 业务2由于设备响应速度慢, 需要占用更多的资源, 所以每个集群配置为 4 个 WAS 服务, 共 8 个服务进程。

5. 每个机器上各跑三个 HTTP Server, 分别给三类业务进行负载均衡。

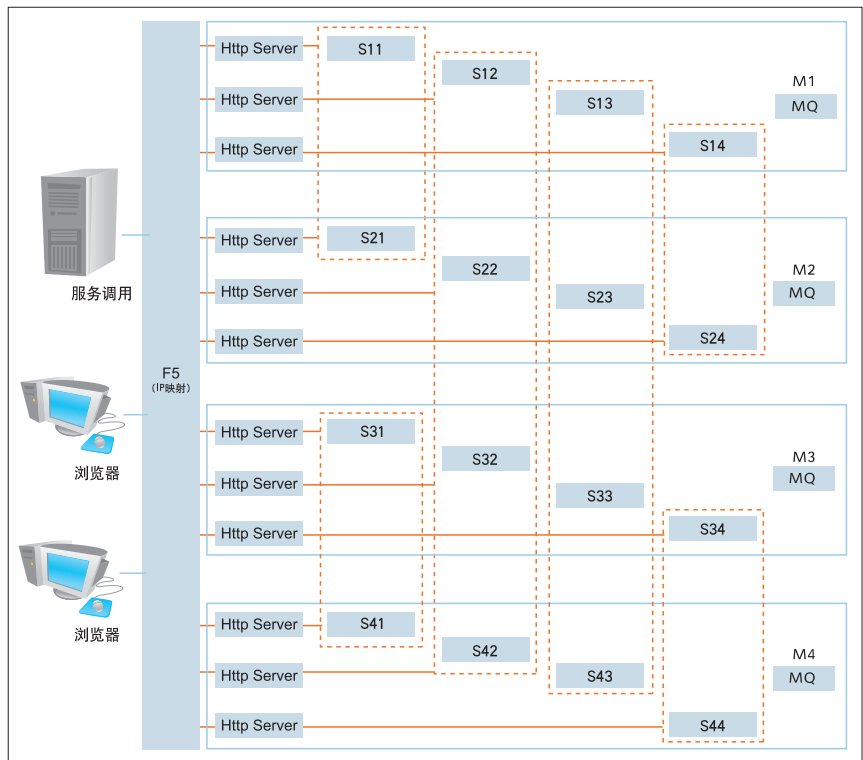
服务开通平台架构拓扑图 3

初期暂定 $4 \times 4 = 16$ 个 server, 后期割接如果需要更多 server, 将可以在每台机器上垂直扩展。

结束语

在搭建服务开通平台的过程中, 虽然由于开始对系统的压力估计不足, 机器资源与集群方案都出现些问题, 不过基于 J2EE 架构与 WAS 容易扩展的特点, 我们还是成功的变更了集群结构, 解决了问题。

集群	app server	备注	处理业务
1	a11、a21	和 2 形成互备关系, 上载程序不用中断服务	业务 1
2	a31、a41	和 1 形成互备关系, 上载程序不用中断服务	
3	a12、a13、a22、a23	和 4 形成互备关系, 上载程序不用中断服务	业务 2
4	a32、a33、a42、a43	和 3 形成互备关系, 上载程序不用中断服务	
5	a14、a24	和 6 形成互备关系, 上载程序不用中断服务	业务 3
6	a34、a44	和 5 形成互备关系, 上载程序不用中断服务	



服务开通平台架构拓扑图 3

解决方案 专家点评

富通东方技术总监: 张东焕

本方案采用 IBM WebSphere Application Server 设计了 BOSS 服务开通业务平台, 通过采用 WAS 的集群功能很好地解决了系统负载不断增加的情况下的业务平台横向扩展及纵向扩展问题。同时利用根据业务的种类, 将交易进行分类接入处理, 减少了不同类别交易的相互资源争用, 保证了需要快速响应的交易。本方案的另一个特点是利用 WebSphere MQ 实现了交易松耦合异步处理, 这也是对批量处理交易有效的提高效率的一种方式。

中联集团系统软件技术部总监：甘荃

作者具有很高的系统架构技术功底，文章通俗易懂，思路清晰。文章中的系统设计案例和优化方法对一些项目建设和系统规划有非常高的借鉴意义。该方案有比较高的前瞻性，但由于缺少对系统的压力的真实评估，带来系统的架构上的调整，这样也给生产系统带来维护带来风险。在今后的项目中，比较理想的情况，系统架构确定应该尽量少的调整，压力的不足只是简单通过

增加应用服务器数量来满足即可，从而实现最佳的可扩展性。

软通动力资深架构师：李俊杰

使用WAS搭建高可靠性系统，从项目的背景介绍到软硬件配置的详细说明，及在实施过程中的细致入微地描述每次优化原因及优化措施，包括网络拓扑的优化，确实是很难得很实在的第一手资料，是实践性和实用性很强的文章。同时又具高屋建瓴的理论高度、举一反三的实践效果。■

广东移动综合业务运营支撑系统解决方案

概述

挑战：如何对多区域、复杂业务产生的海量数据进行高效整合？如何优化流程，提高反应速度和业务效率？

解决方案：

ISV 应用程序：从兴 BOSS 2.0

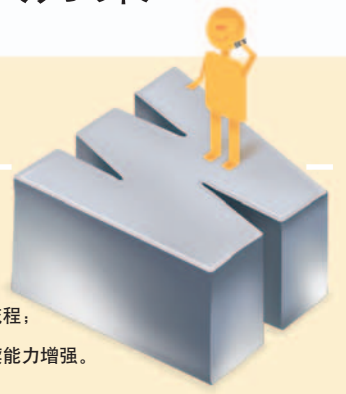
中间件：IBM WebSphere

硬件：16台 IBM P595、3套DS8300存储阵列、2套DS4800

存储阵列和 PC Server 若干

收益：

整合全区域、全业务的数据；统一管理界面、简化业务流程；缩短反应时间，可实时推出新业务；安全保障，业务连续能力增强。

**综合业务运营支撑系统解决方案**

广东移动综合业务运营支撑系统 (BOSS2.0) 是在集团公司 BOSS2.0 规范指导下，适当超前，并在广东移动需求要求下进行规划建设的，该系统集中整合了广东移动覆盖区域及所涉及复杂业务，同时实现了容灾和备份的优化，最大限度的保障了系统的高效、稳定运行。

该系统容量达到 160T，支撑超过 7000 万用户业务受理和准实时计费，系统每月处理话单 600 亿张、其中清单 200 亿，其中计费子系统是全球最大、最高效、最精确的移动计费系统。

区域整合：该系统可以实现地域集中、

系统集中，形成全省集中的 BOSS 支撑体系，支持广东省内所有的行政区域，形成广东移动真正的 100% 覆盖。

业务整合：该系统支持不同渠道、多种接入方式 (营业厅、呼叫中心、自助终端、营业厅、代办点等) 采用统一营销界面和业务逻辑向用户提供服务。同时该系统可以统一管理用户的所有业务订购关系、包括数据业务、梦网业务、12580 等订购关系，为 3G 时代多样化的业务种类和灵活的融合计费作支撑，并支撑全品牌用户管理、满足融合计费。

按需应变：该系统基于 J2EE 的软件结构和基于 WEB 的浏览器界面，建立起面向服务的架构和面向流程的系统设计、

统一服务营销流程，通过流程引擎实现业务流程灵活配置。有了这样的系统支撑，广东移动就可以根据市场变化和客户需求，随时推出新的业务品种。

容灾备份：该系统采用双中心结构，实现数据容灾和关键业务容灾，增强支撑网的业务连续能力。

面向未来：该系统能够满足集团 BOSS2.0 业务技术规范要求、满足 3G 初期业务运营要求、满足广东省快速发展的业务需求。

高质量、高效率的业务支撑

该项目要满足集团公司 BOSS2.0 规范要求并适当超前，满足广东移动现状需



求，建设全省集中的BOSS系统，实现全业务多品牌的运营，实现对集团客户、数据业务支撑，实现3G业务初期运营支撑。

该系统处理形成的海量数据，为广东移动的业务运营和决策提供了科学的数据基础。通过对数据的分析和提取，广东移动可以准确获取用户使用信息，并有针对性地对现有业务进行调整，及时推出更能满足客户需求的业务和服务，从而达到业务量和用户满意度的双提升。

成功实施

2006年，该项目顺利完成第一阶段、第二阶段的工期，并

顺利投入使用。

2007年初迈入新阶段的开发维护工期。

中国移动广东有限公司项目负责人指出：IBM和从兴联合实施的综合业务运营支撑系统(BOSS 2.0)高效完成了对负责业务的处理，帮助广东移动在激烈的竞争中随需应变的推出最具竞争力的产品和服务，从而更好地满足移动通讯客户的需求。

广东是移动通讯发展最为成熟的地区，同时也是竞争最为激烈的地区。依托完善的综合业务运营支撑系统，从服务内容、市场模式、经营理念等方面进行调准，全面提升核心竞争力，为客户提供差异化、个性化的优质服务，成为广东移动致胜市场的关键。■

广州移动全省经营分析系统解决方案

概述

挑战：为适应日趋激烈的市场竞争环境和市场发展需求，广东移动各地市公司迫切需要更新业务分析和运营的工具，从而提高广东移动核心竞争力。

解决方案：

ISV应用程序：从兴BI

中间件：IBM WebSphere

硬件：4台P5 590, 2台P5 570, 2台P690, 3台P670, 2台P630, PC服务器和存储设备若干。

收益：

丰富了经营分析的应用

有效地支撑了决策分析及地市子公司的个性化分析应用

新业务分析和运营的不可或缺的工具

新业务分许可期缩短

及时监控新业务开展情况

精确营销平均成功率提升3倍左右



全球最大的电信级数据仓库

提供全省性的经营分析系统和综合数据仓库，对未来业务发展进行科学预测，为市场经营工作提供及时、准确、科学的决策依据。

背景

为适应日趋激烈的市场竞争环境和市场发展需求，中国移动通信集团广东有限公司(以下简称广东公司)觉得应该充分利用业务支撑系统产生的大量宝贵的数据

资源，建设全省性的经营分析系统和综合数据仓库，实现对信息数据的智能化加工和处理，充分利用数据挖掘分析技术，对未来业务的发展进行科学预测，为市场经营工作提供及时、准确、科学的决策依据，从而提供广东移动核心竞争力。

多重深入的收益

系统在2006年通过市场分析体系、集团业务分析支撑平台、数据业务子系统等功能建设和完善，极大地丰富了经营

分析的应用，有效地支撑了移动公司市场、集团、数据业务等业务部门的决策分析以及地市子公司的个性化分析应用。

经过实践的历练，该系统的业界领先地位和优越性获得了运营商的高度肯定，其稳定性和易用性亦得到基层使用人员的一致认可，成为广东移动各地市公司新业务分许可和运营的不可或缺的工具。新业务分许可期缩短，从原来的以月为周期到现在的以日为周期，保证各项新业务开展情

况的及时监控;新业务精确营销活动从设计到推出市场的时间缩短,精确营销平均成功率比以前提升3倍左右。

经营分析系统

经营分析系统符合中国移动集团公司经营分析系统规范,基于J2EE三层架构的分析型系统。系统采用CUI图形用户接口,由下至上分为数据源、ETL、数据获取、数据仓库、数据提升(增强)、数据集市、商业应用和前端展现等几个层次。

ETL在数据抽取、提升的各个阶段都涉及到ETL的处理,它包括数据抽取、数据清洗、数据转换、数据加载、质量控制、作业调度等功能。

数据仓库:包含运营数据库(ODS)和企业数据库(EDS)。

EDS在ODS基础上进行提升,保存数据仓库中基本的规范化数据,对数据进行整合和管理,并创建数据集市的单一事实源,直接支持最终用户信息需求。

前端展现:提供数据库各个层面,各个方向的数据集中展示功能。包括:即席查询、预定义报表、OLAP分析、自定义

报表、元数据查询、KPI和EIS展示、批量查询处理。

主题分析:基于对电信运营市场和运营商需求的理解,提供的针对性的分析,主要包括:客户发展分析、业务发展分析、收益情况分析、营销管理分析、市场竞争分析等。

元数据管理:对数据仓库环境中的系统构建元数据,控制元数据和业务元数据进行集中统一管理,形成元数据库。

系统管理:协调各子模块内部的操作和子模块间的协调,以及同用户的友好交互。其主要包括:安全管理、健康与告警、角色管理、消息管理、参数管理和性能管理。

系统显著特点

全面的、多行业的分析主题,涉及客户发展、业务发展、收益分析、大客户分析、竞争分析、市场营销分析、服务质量分析等。

支持超过200TB级的海量数据的数据仓库,提供高性能、高可用性、高可扩展性的电信级经营分析及决策支持应用。

支持B/A/S体系结构、零客户端,满足决策支持的随机性和及时性。

灵活的体系设计支持与各种主流数据库平台,多维分析引擎,数据挖掘引擎和前端展示工具的无缝集成。

提供丰富的业务分析模型和二次开发的接口,更好地实现个性化的决策支持应用。

采用松耦合的设计,通过产品间定义的接口组合应用,实现更复杂更深入的挖掘分析。

快速构造企业商业智能信息门户,为各类用户提供丰富多彩的、个性化的决策支持信息服务。

中国移动通信集团广东有限公司项目负责人指出:在IBM和从兴公司的大力协助下,广东移动经营分析系统自从2004年上线以来,已经稳定运行了4年多。经过实践的历练,该系统在业界居于领先地位和卓越的优越性。成为广大移动各地市公司新业务分析和运营的不可或缺的工具。该项目还被评为2006中国电信业运营支持&IT系统优秀项目奖。■



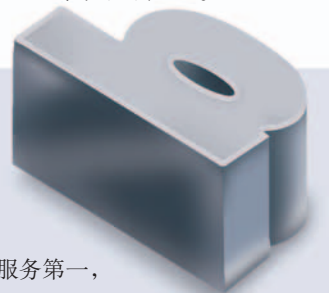
广州从兴电子开发有限公司

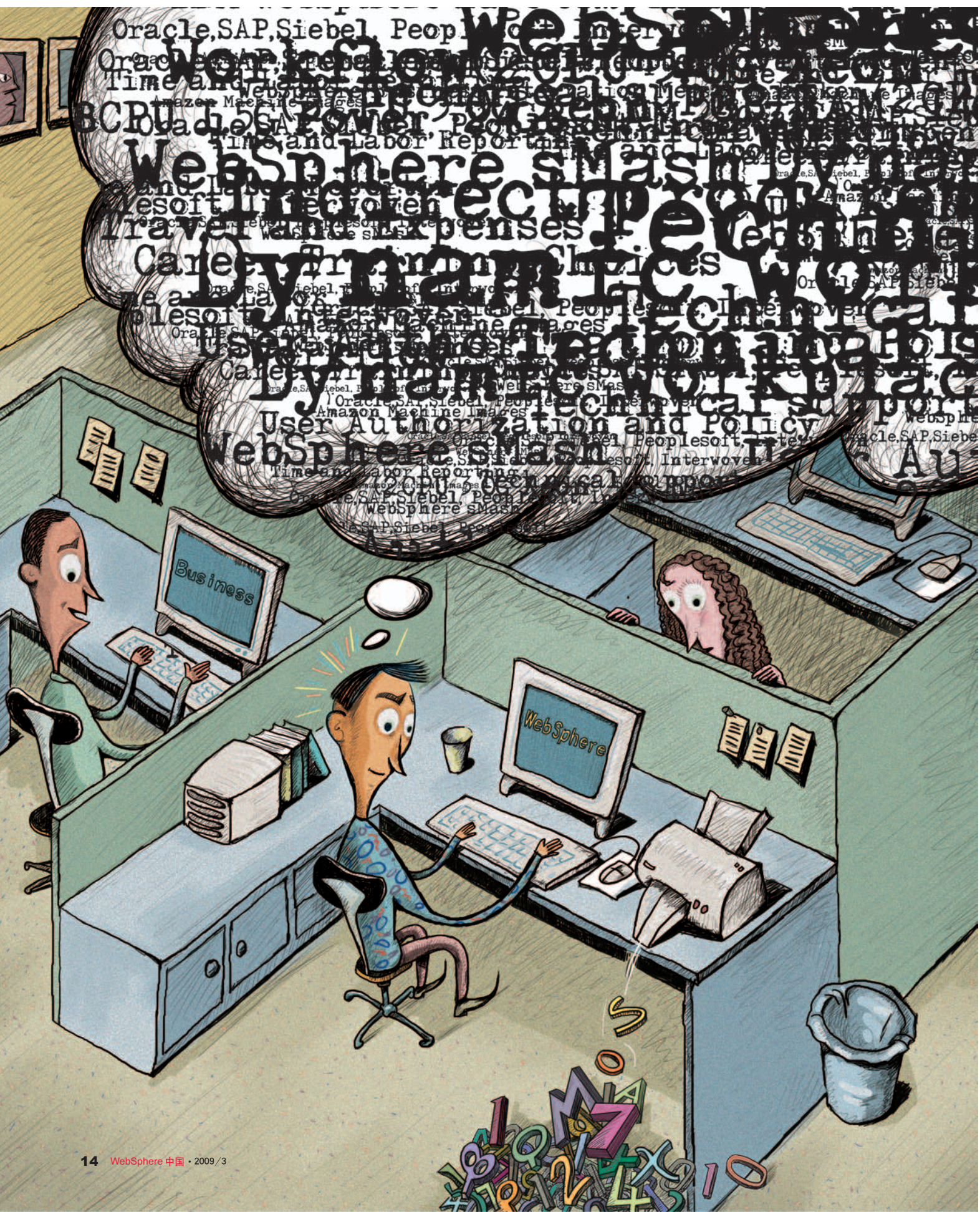
广州从兴电子开发有限公司隶属于广东立信企业(集团)有限公司,是国内一家实力雄厚的大型软件开发和系统集成商。公司自1999年成立以来,已逐步发展成为致力于为通信、电力、政府等相关领域提供大规模数据处理、信息管理、通信技术应用的设计开发、系统集成、运营维护和外包服务的全球领先行业解决方案提供商。

目前公司有员工近1000人,86%以上为拥有学士或学士以上学位的专业技术与管理人才。公司已经在北京、香港、南

京、重庆、沈阳等地设有分支机构,形成了立足广东、拓展全国的销售网络。

长期以来,从兴公司以“服务第一,客户满意”为目标,始终站在行业技术发展的最前沿,服务于移动、电信、网通、联通、电力、政府、公安等行业,以一流的软件产品,领先的集成技术和卓越的运营经验,不断地为客户提供全方位的服务,与广大客户共同发展,共创辉煌。■





知识管理 & 企业门户解决方案

上海慧广科技发展有限公司 / 文

一、方案概述

DNE 企业信息门户解决方案是使企业的员工、客户、合作伙伴通过一个统一的渠道个性化地访问企业内部、外部的各种信息的应用系统,使其达到协同工作的目的。整个解决方案分为三个层次:企业应用集成(EAI);支持门户的基础架构(流程与 workflow、内容与数据管理、用户身份管理、目录服务器);基于门户的软件(企业知识门户、企业信息门户、企业应用门户)。

通过DNE 企业信息门户方案,为企业带来的优势如下:

提高企业的应变速度,提升知识管理

提高资源利用的效率;

整合企业所有应用及信息来源为单一接口;

提升同仁工作生产力、实时反应与沟通效率;

提供高度市场个性化与客户化的设计;

与客户、合作伙伴及供货商间的商情

信息分享;

建立学习型组织与最佳工作协同互动机制;

大幅降低企业无关营业收益之人工成本。

二、方案特点

统一的信息访问渠道: 通过将内部和外部各种相对分散独立的信息组成一个统一的整体,使用户能够从统一的渠道访问其所需的信息。

单点登录(SSO): 一次登录,可以访问所有权限许可范围内的已经集成的系统。

强大的内容管理能力: 系统支持几乎各种结构化和非结构化的数据,并可以搜索和处理各种格式的文档。

个性化的应用服务: 信息门户的数据和应用可以根据每一个人的要求来设置和提供,定制出个性化的应用门户,提高了员工的工作效率,增强了对顾客的亲和力和吸引力。

短时间内无法解决的技术需求问题。

扩展企业内部网的范围: 授权给机动雇员、客户、使用者、合作伙伴及远距离工作者可随处访问企业防火墙后的任何资源(应用软件和数据等)。

协作及共享: 允许使用者存取或提供信息给特定的个人或群体。内部员工可以通过EIP 系统分享信息,而客户、供货商及合作伙伴也可以通过企业之间的网络达到信息分享的目的。

综上所述, DNE 门户解决方案一方面能提高企业内部的工作效率,减少不必要的成本和资源的使用,使企业能专注于新的业务发展;另一方面使企业与外部的交流和合作更有效率,与顾客和合作伙伴建立更好的互动关系,从而在快节奏、竞争激烈的环境中保持领先的竞争优势。

三、系统主要功能

1 个性化和定制

客户登录企业的企业信息门户以后,可以自己配置自己的桌面环境,包括选项(Options)、内容(Content)、布局(Layout)。

2 信息检索

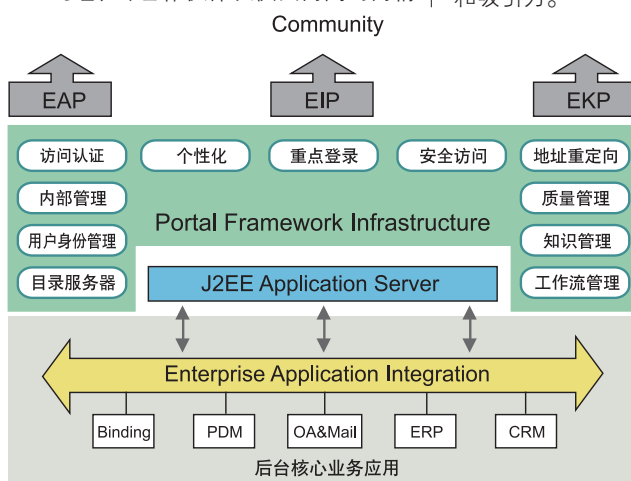
提供对于整个门户系统的信息检索功能。

3 员工自我服务(Self Service)

员工福利(Benefits)资料查询和自我更新

职业培训选择(Career Training Choices)

工时报告(Time and Labor Reporting)



与应用系统的集成: 能将企业现有的数据和应用无缝集成到一起,无需重新开发,保护了原有的投资,如下图所示。

高度的可扩展性: 能适应企业新的人员和部门的调整的变化,满足企业业务调整 and 扩展的要求,解决企业与 IT 部门

E-learning

电子报销 (Travel and Expenses)

间接采购 (Indirect procurement)

技术支持 (Technical support)

4 workflow管理 (Workflow)

DNE 门户解决方案可以方便的集成 DNE 工作流引擎系统, 帮助企业实现整个业务流程的管理。

5 动态工作空间(Dynamic workplace)

建立在门户上;

用户在一个工作界面上看到所有相关的政务/业务系统, 服务和数据;

结合各级领导员工的权限, 角色和业务范畴, 不同公司, 部门和等级的人员仅看到与己有关的内容;

容纳新业务的开发集成, 无需改变用户端的技术设置和界面;

不论何时何地, 使用何种设备, 用户均可安全的通过工作空间调用相关服务处理业务。

6 内容和信息系统集成

企业各部门和项目组已经积累了很多

的文档信息, 这些信息往往是专门为某个部门、某个项目或某个服务而存在的, 门户必需能够整合这些分散的和孤立的内容和信息, 以提供企业的统一信息入口。

7 应用和服务集成

企业内部存在多种类型的应用, 从 Web 的标准来划分, 可以分为 HTML 和非 HTML 两种类型。

HTML 类型, 既是 Web-Enable 类型的应用主要包括:

员工出差审批应用;

员工报销等;

非 HTML 类型, 既是传统的 Client/Server 类型的应用, 主要包括:

电子邮件系统;

文件系统(Windows NT 文件系统和 Unix 的 NFS 文件系统)

远程登录和文件传送的应用;

OA 系统的应用(如: Lotus Notes);

ERP;

CRM 等。

8 网域用户群的建立和管理功能

企业信息门户提供用户验证的功能。验证的方式可能是以下方法中的一种或多种:

LDAP 目录验证;

Radius 验证;

S/Key 验证;

SafeWord 验证;

SecurID 验证;

基于电子证书的验证;

基于 Unix 用户的验证;

基于 NT 域用户的验证;

用户访问控制功能(User Authorization and Policy)。

门户通过用户验证模块进入一定的网域后, 门户还必需具有访问控制的策略, 网域由信息、内容、服务和应用构成, 它们按频道(Channel)组织, 网域还包含属于该域的角色和用户, 控制策略是指门户管理者能控制用户对网域里的资源的访问权限, 如某个域、角色和用户能否访问某个 URL、某个应用等。

系统环境

服务器端:

IBM Websphere Portal Server;

IBM Lotus Domino R5/R6;

内存: 1G 以上。

客户机:

操作系统: Windows 9x/2000/XP;

所需软件: IE 5.5 或以上的版本 内存: 64M, 建议 128M

以上;

硬盘空间: 4G 以上;

内存: 64MB 以上。

系统网络结构

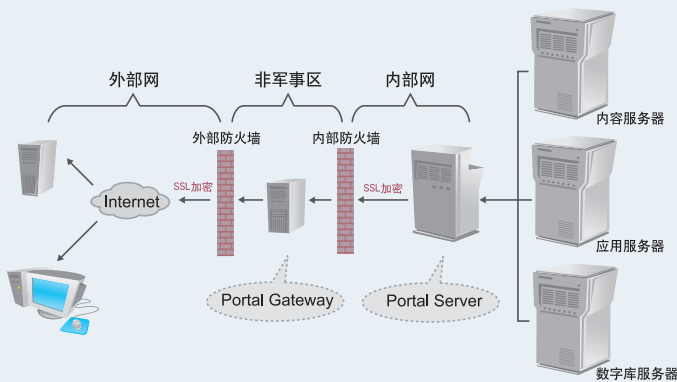
一般来说, 企业在互联网上部署了一个新的网站时, 无法让访问者通过它来直接访问企业内部网的资源, 因为企业将自己的网站服务器托管到 IDC(Internet Data Center) 时, 只获得一个合法的能被互联网用户寻找到的 IP 地址和域名, 至于内部网的资源, 包含公司一级的网域及其属下的子域, 访问者通过新的网站是无法寻址到公司内部。另外, 出于安全方面的考虑, 公司的内部资源也不敢直接向互联网用户开放。

面临以上的问题, DNE 企业信息门户是一个非常好的解决方案。不但能够让访问者顺利的访问到公司内部网的资源, 而且

还提供绝对的安全保证,因为所有的访问都必需经过验证,并且所有的联接都使用 SSL 协议。

通过为 PS Gateway 配置 Web Proxy(Web 代理)和 URL 重写规则,互联网的访问者就可以通过 Gateway 访问到内部网的网站。Web Proxy 的寻址表告诉 Gateway 要访问某个 Domain 或 Sub Domain 或某台 特定的机器应该访问那个 Web Proxy,而 URL 重写规则保证企业内部网资源在 通过 PS 访问时能正确无误的工作,因为用户对内部网的结构是透明的,所有的访问和资源定位均通过 PS Gateway 来实现。

如图所示,门户系统分为两部份,PS Gateway 和 PS Server,在为企业部署门户服务时,采取的方式为:



外部防火墙 → PS Gateway → 内部网防火墙 → PS Server

成功案例

Capitaland 凯德置地企业信息门户平台

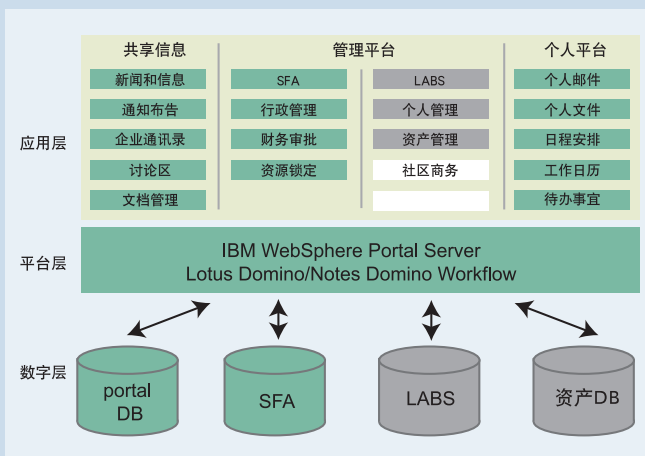
项目背景:

凯德置地中国控股集团是一家活跃于中国房地产市场的投资与开发公司。凯德置地致力于国内高品质住宅和商用房的开发,进而为人们创造更美好的生活环境。自 1994 年进入中国以来,凯德置地在上海、北京等地开展业务,在华总资产达 50 多亿人民币。Capitaland 经过多年的信息化建设,建设了各种应用系统包括销售自动化系统、基于 Lotus Domino 的文档管理系统、物

资管理系统、windows 域管理等。随着数据量增大及地域的分散,各种系统形成"孤岛"。企业员工要访问多个应用系统才能获取相关的信息,并且用户在切换应用系统需要重新输入密码,管理员则需要管理多个应用系统的用户信息。为了解决相关的问题,Capitaland 决定采用 DNE 企业门户解决方案,旨在实现实现内部信息共享;实现部门间跨地域协同工作;实现与外部相关组织机构间的信息共享和协同工作;

采用方案:

DNE 公司针对 Capitaland 信息化现状和需求,提出了企业信息门户的解决方案。整个方案采用"整体规划,分步实施"的原则,第一期工程实现对企业内部的信息门户,整个系统的建设包括了用户统一管理平台建设 (Microsoft 活动目录与 Domino LDAP 集成),企业应用集成 (销售自动化系统),企业工作流平台建设,企业内容管理平台建设,及门户表示层的建设,并实现了单点登录。



实施效果:

构建了统一的信息访问渠道,通过门户系统用户能够通过统一渠道访问所需信息。实现了单点登录和用户统一管理。实现了个性化服务和桌面的频道功能。员工可以通过 Internet 方便、安全的访问集成到 Portal 中的各种应用系统。构建了用户身份管理平台、工作流程平台、门户展示平台。使企业整个应用系统规范性和扩展性提高。方便于系统今后的扩展,及其他应用的开发。■

药品监督管理局协同办公平台解决方案

1、药品监督管理局协同办公平台概述

药品关系到国计民生，利用先进的信息化手段，加快药监行业信息化的建设步伐，已成为提升药监行业管理水平的主要措施之一。药品监督管理局协同办公平台，就是针对药监行业的应用特点，推出药政系统解决方案。全面掌握全省食品药品监管系统公文流转、通知交流、行政审批管理、行政审批监察管理、咨询投诉管理、待办事项管理、公示公告管理、会议管理、消息管理、论坛管理、会议管理、工作邮箱管理、网站信息管理动态信息。

2、药品监督管理局协同办公平台的实现目标

- (1)、建设提高办公效率和质量的自动化协作平台，优化管理流程、规范工作程序，提高各项工作的准确性和可控程度，最大限度实现无纸化办公，提高工作效率、降低办公成本；
- (2)、建设行政许可审批服务平台；

- (3)、满足全省单位用户快捷地进行内部门户和外部网站信息管理；
- (4)、建立一套科学的评价体系，为各项工作和领导决策提供准确、及时的数据依据；
- (5)全面掌握全省食品药品监管系统公文流转、通知交流、行政审批管理、行政审批监察管理、咨询投诉管理、待办事项管理、公示公告管理、会议管理、消息管理、论坛管理、会议管理、工作邮箱管理、网站信息管理动态信息，并为第二阶段食品药品安全数据中心建设和第三期业务应用系统实施奠定基础；
- (6)、建立一套科学、高效的食品药品安全及单位内部知识管理平台。

3.药品监督管理局协同办公平台系统功能

药监端系统

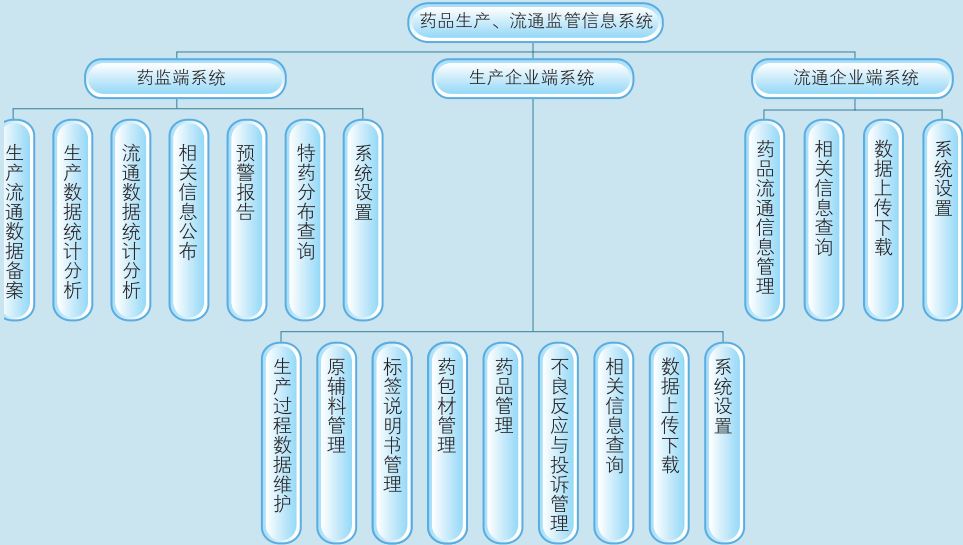
各级政府的药品监管部门能及时查询统计药品生产企业和药品流通企业上报的药品生产和流通情况，对药品在生产、流通过程中的不正常现象及时给予报警提示，以及公布药品相关信息等。

其功能包括：生产流通数据备案，生产数据统计分析，流通数据统计分析，信息公布，预警报告等。

生产企业端系统

药品生产企业对于生产数据的维护以及与药监端系统的数据的上传下载。

其功能包括：生产过程信息维护，原辅料管理，标签说明书管理，药包材管理，药品管理，不良反应与投诉管理，数据上传下载。



流通企业端系统

药品流通企业对于流通数据的维护以及与药监端系统的数据的上传下载。

其功能包括：药品流通信息管理，数据上传下载

支撑服务

支撑服务为业务应用提供的一组共性和关键的服务,包括统一用户管理、工作流管理、数据交换、门户管理、统计报表/打印组件、字典管理、即时消息中间件、数字签名/电子印章等。



● 统一用户管理：主要分为组织机构管理、权限管理和单点登录管理三部分。在山东省食品药品监督管理局内建立统一的组织机构层次；对用户角色授权进行统一的管理；实现用户登录一次就可以访问各个应用系统的功能。

● 工作流管理：提供定义、创建、执行工作流的平台。具体包括：工作流引擎、工作流工具（流程设计工具、表单设计工具和流程监控工具）、工作流系统管理和个人工作流应用等部分。

● 数据交换：为需要数据集成的应用提供数据服务，提供数据的提取、复制、传输、转换、交换、筛选等。

● 门户管理：实现统一门户单点登录，提供完备的界面定制组件实现个性化页面定制。通过信息发布平台统一发布门户信息，并可通过搜索引擎实现对网站、数据库和文件等的检索。

● 统计报表/打印组件：通过报表设计器设计报表，能按多种格式导出统计报表，并可对生成的报表实现精确打印。

● 字典管理：对系统数据字典和数据对象的注册、维护、使用进行统一管理。字典管理主要包括：数据字典管理、数据对象管理。

● 即时消息中间件：建立操作系统级的即时消息系统，集成邮件系统和短信系统，实现用户对各种消息传递方式的灵活选择。

● 数字签名/电子印章：数字签名基于PKI（公钥基础设施）

的电子签名，为所有网上应用，透明地提供加解密和数字签名等安全服务所需要的密钥和证书管理。电子印章实现在 Word，Excel，Html 等文档上的电子签名、盖章。

● 集成开发工具：提供一个集设计、开发、调试、发布、运行、版本管理于一体的可视化的集成开发环境，提交基于支撑平台开发的易用性，解决传统软件开发过程中软件重用性差，可维护性差等问题。

● 支撑服务是与具体业务无关、通用的基础服务。应用系统一般通过应用开发接口或声明性的描述来使用支撑服务。

4、药品监督局协同办公平台运行环境与配置

来源	名称	功能说明	数量
典型配置 (小型机)	IBM P5-55A(标准配置)	药品生产监管服务器	1台
典型配置 (小型机)	IBM P5-55A(标准配置)	药品经营监管服务器	1台
典型配置 (小型机)	BM P5-55A(标准配置)	药品生产服务器	1台
典型配置 (小型机)	IBM P5-55A(标准配置)	药品经营服务器	1台
IBM 软件产品	WebSphere Application Server	应用服务器	4套
	DB2	数据库	4套

5、成功案例

吉林药品生产质量非现场监控系统

吉林药品生产质量非现场监控系统是以网上数字监控和视频监控相结合的方式对药品生产质量有效实施监管。该系统的建设目的是提高企业以信息化手段提升内部质量管理的主动建设意识，逐步实现企业自身和药监部门对企业药品生产质量的实时监控，有效控制药品生产质量管理关键环节，以信息化手段实现非现场监管的目的。该系统已成为药品生产企业质量管理和药监部门监管工作的重要手段，达到了“药监部门管得住、药品生产企业用得上”的双赢目的。

系统特色功能模块

备案查询模块：将各企业上报的涉及GMP质量信息的关键环节数据进行自动电子备案。同时提供丰富的查询接口对数据进

>>>

按需应变的企业应用中间件平台

神州数码系统科技公司 · IBM 产品本部供稿

如何降低实施成本，有效地建立电子商务、电子政务等信息系统？

如何实现组织结构和工作流程的重组优化？

如何超越时间、空间和部门分隔的制约，建成一个精简、高效、廉洁、公平的运作模式？

如何将企业的进、存、销管理与网上销售及订单管理融为一体？

这些都是现代信息化建设所面临的新的挑战。

此应用中间件平台是基于软件构件复用技术的业务基础平台，正是面对这一挑战所开发的产品，使用此平台能够快速按用户实际需求，配置企业信息系统或电子政务系统，将企业或政府机关内部各个分离的信息资源组织成一个有意义、灵活的

整体，用技术手段帮助企业提高运行效率和服务质量，增强综合竞争实力，达到用信息化带动传统产业发展、提高办公效率的目的。

方案功能描述

按需应变的企业应用中间件平台是面向应用开发商和最终用户提供的便捷、高

效、强大、快速的应用开发平台。采用纯 Java 平台开发，支持 J2EE、EJB、XML 等标准。它封装了诸如内容管理、Web 应用生成、信息采集、权限管理、报表分析、用户安全、交易生成等大量实用功能，可帮助用户快速开发从小型工作组级到大型企业级的应用产品。此平台通过一系列组件实现各种重要的功能，如：所有的表单

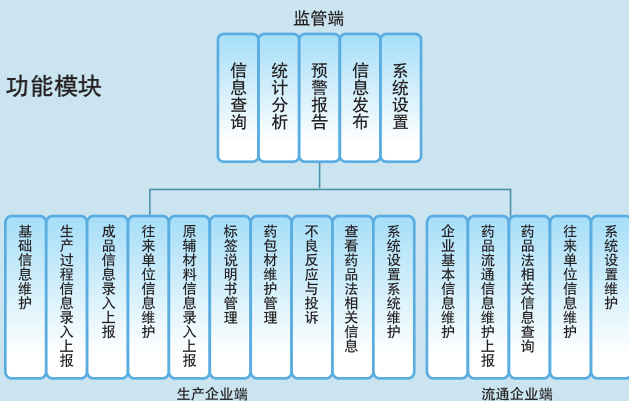
>>>

行快速检索。

统计分析模块：将各企业上报的涉及 GMP 质量信息的关键环节数据根据药监监管的要求，进行智能化的实时统计分析功能。

预警报告模块：药监局根据自身的监管需要设置预警规则，系统自动进行实时数据预警，报告给药监相关人员。

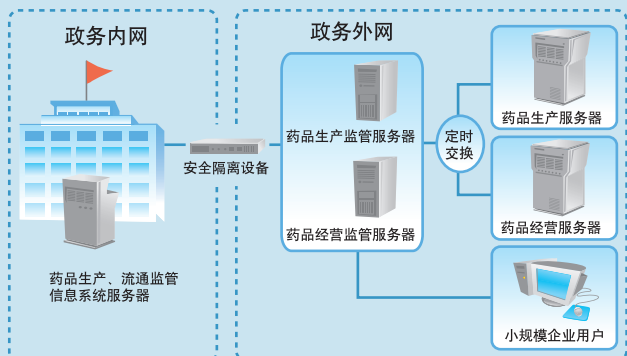
视频监控模块：通过实时的、不可修改的视频信息作为有效的辅助监管手段，为药监局及企业管理人员提供了一种远程管理的手段。



山东省食品药品监督管理局

山东省食品药品监督管理局药品生产、流通监管信息系统软件涉及到众多的部门、大量的业务、业务过程的梳理和整合，要建立网络方式下各部门协同、共享的药品生产、流通监管信息系统，系统的建设不仅是处理手段的转变，更是工作方式的变革——用一种全新的工作方式代替已经习惯了的现有的工作方式。■

拓扑结构



可以自动产生、表单的结构可以动态调整、表单的页面效果可以随时变化。体现了一种按需配置的设计理念,最大限度的满足用户对于数据采集方面的灵活要求。

● 企业应用控制台

将标准的后台管理系统封装成组件提供给用户,方便用户进行菜单的灵活搭建、权限管理的设定、动态配置的系统参数。用户开发企业应用系统时,只需在系统中设置即可,无需编写任何代码。

● 内容管理

内容管理和信息发布是电子商务应用的基础。此平台不仅提供了浏览器界面使内容的提供者录入信息,更重要的是,它还可以把数据库中存放的内容按照模板规则,生成最终用户需要的Web页面,同时对页面的链接关系和数据的存储进行管理。

● 表单及报表的无代码生成

此系统的最大特点是所有基本的表单和统计报表都可以由用户直接通过生成向导生成,不需编写任何代码。对于复杂应用只需在基础表单上简单修改即可。

● 表单的数据结构、数据表现动态调整

此平台支持数据库的数据结构动态调整,表单的结构调整后,系统不需修改任何代码,只需重新配置一下表单即可随之改变。整个过程完全由用户随心所欲,轻松完成。

● 统计图及报表管理组件

提供灵活定制报表及统计图的功能,且所有的报表都可以在无需对源代码做修

改的情况下灵活调整。

● 企业级的支持

基于开放的体系结构,支持最新的J2EE标准。系统支持流行的硬件平台和主流操作系统如(Windows NT、UNIX、LINUX),有强大的伸缩性。n 数据接口EPod内部应用数据可与与其它各类应用无缝的交换数据,可以使用XML数据交换标准,可以符合cXML 1.2 (Ariba Standard)、ebXML、BizTalk (Microsoft Standard)保证用户能将其业务平滑的与其他系统相衔接。

带给客户的收益

此系统能够帮助使用者可靠、高效地开发各种基于Web的应用系统。开发者不用再象以往一样开发大量复杂、耗时的底层代码,可直接进行实际业务系统开发,极大的缩短了开发周期,提高开发效率,降低开发成本,赢得更多的满意度。为企业和政府提供一个开发成本低、维护成本低、实施周期短、通用、安全,不需要专业知识的人员就可以进行项目实施的中间件产品,解决企业信息化网络软件的基础平台支持、面向应用的技术开发、整合企业现有信息系统,使企业在进行信息化建设时不再无所适从。

最主要价值是帮助开发商开发各种基于Web的应用系统,可以做到:

● 缩短应用的开发周期

Web应用目前流行的开发方式是在各种应用服务器上如WebSphere Application Server、WebLogic Application Server上直接开发业务系统。虽然大量的底层服务开发商已经不需要关心了,如负



载均衡、数据库连接、硬件支持等,但是涉及到实际的开发工作,开发商还是要做大量的代码工作,采用此平台将极大的缩短开发时间。通过大量的封装组件,应用开发商将可以把更多的精力放在实际的业务系统中,将传统的开发代码时间缩短至少三分之二。

● 节约应用的开发成本

采用此平台开发实际业务系统时极大的缩短了开发时间,带来的明显好处是节约了开发成本。同时因为大量的封装组建对于整个项目的开发队伍的技术水平的要求也有所下降,客观上也节约了人员费用。

● 减少系统初期的建设成本

提供对于各种底层应用服务器的支持(WebSphere、WebLogic、Tomcat),对于大型数据库的支持(DB2、Oracle、MySQL),对于各种硬件的支持(PC Server、RS6000、AS400、SUN),由此使得应用开发商不必关心各种基础系统的差异,极大的减少了系统初期的基层建设成本。

● 降低应用开发的失败率

良好的封装、精简的代码、可靠的底层系统可以很好的降低应用开发的风险。

● 简化应用集成

虽然 Java 技术承诺可以一次编译多次使用但是在实际的系统中还是有少量的差别的，特别是多语言的移植问题一直是困扰开发者的一个难点，通过此平台可以简化系统的集成和移植。

● 减少维护费用

企业的信息系统，因为企业自身的成长和变化经常需要调整，传统的开发模式对于系统的应变能力都比较弱，一般会增加企业的信息系统的维护费用，采用此平台的运行时动态调整技术，可以减少系统的维护费用。

方案的独特之处

● 高效率

此平台上集成了模板、内容管理、页面编辑等实用的功能模块，同时提供完善的后台操作界面，使得在构筑信息门户网站过程中可以提供极高的开发效率。

● 高可靠性

经过长期的时间和项目验证，多次优化的代码和面向对象的设计理念、UML 的开发方式为企业应用提供强壮的内核，因此在其上构筑的应用具有更高的可靠性。

● 开放性

全面支持 J2EE 标准和基于中间件的应用服务器技术，为用户提供优秀的扩展性，具有更完善的开放性。

● 安全性

强大的权限管理为用户的网络应用安全保驾护航：

● 易使用性

采用标准的浏览器作为用户的操作界面，增强了易用性。使用者经过短期的培训便可掌握如何使用为企业创造价值。

● 可维护性

无论是最终用户还是系统的开发人员均可通过标准的浏览器来维护整个系统，包括添加新的模板、增加新的应用等功能，方便您维护整个系统。

推荐的软硬件产品：

硬件平台	IBM System P520
存储系统	IBM DS4700
软件平台	IBM DB2 UDB、IBM WebSphere Application Server
操作系统	AIX
适用的行业	政府、中小型企业等
适用的用户群	软件开发商，最终用户等

IBM DS4700

具有 4 Gbps 能力的端到端光纤通道接口技术，有助于优化性能；

高达 1600 Mbps 的带宽，适合高吞吐量应用；

支持光纤通道硬盘驱动器；

包含 IBM System Storage DS4000 Storage Manager，有助于集中管理 DS4000 系列；

共八个主机通道，配备双控制器，以提高连接能力；

最多支持 112 个磁盘驱动器模块，可连接 6 个 DS4000 EXP810 扩展单元；

IBM P5 520

POWER5+ 微处理器：

- 旨在提供卓越的应用程序性能和高可靠性；
- 包括并发多线程以帮助提高商业系统性能和处理器利用率。

高内存 / I/O 带宽：

- 快速的处理器减少了系统间传输数据的等待时间；
- 能更快地传输数据以满足 HPC 和其它内存密集型应用程序的需求。

安装灵活：

- 可以选择桌边式或 19 英寸机架式（4U 扩展抽屉），拥有灵活的配置。

共享处理器池：

- 能够在分区间透明地共享处理能力；
- 帮助平衡处理能力，并确保高优先级的分区能够获得所需的处理器周期。

微分区：

- 允许将共享处理器池中的每个处理器最多划分为 10 个分区；
- 调优处理能力以匹配工作负载。

虚拟 I/O 服务器：

- 通过共享昂贵的资源，从而节约成本和简化系统管理。

虚拟 LAN：

- 以内存速度进行分区间的内部通信。

动态逻辑分区：

- 支持系统资源的重新分配而无需重新引导受影响的分区；
- 在使用可用容量方面具有更大灵活性，能更为快捷地调整资源以满足不断变化的业务需求。

秉承大型机技术的 RAS 特性：

- 使用通常在更大、更昂贵系统中才有的特性，这些特性包括：服务处理器、Chipkill 内存、首次故障数据捕获、动态释放所选系统资源、热插拔 PCI-X 插槽、热交换磁盘支架、冗余热插拔风扇、热添加 I/O 扩展抽屉、动态固件更新等。

支持多操作系统：

- 允许客户灵活地选择合适的操作系统和应用程序，以满足他们的需求；
- 能够扩展应用程序的选择范围，包括许多开放源码应用程序。

AIX 5L 操作系统：

- 旨在为混合工作负载提供最大的吞吐量而不必进行复杂的系统配置或调优；
- 为系统保护提供集成的安全性功能；
- 利用 Linux 同源性扩展应用程序选项。

Linux 操作系统：

- 支持访问 32 位和 64 位开放源码应用程序；
- 提供跨 IBM 服务器平台的公用操作环境。

表明此功能（部件）是可选的、适用于特定型号，或需要另外的软件。■

企业综合信息平台解决方案

胡伟红 / 文

一、项目背景

为了进一步增强企业活力,企业管理水平和科学技术水平,完成企业信息化建设,提高企业运作效率,降低企业运营成本,更快更好更多地提供针对性的服务,企业有必要尽快实施企业综合信息平台系统,为企业内部员工提供更好的沟通渠道,获得更多的知识和经验,从而达到为客户提供更好的服务的目的。

二、综合信息平台系统的建设目标

- 提供一个信息整合、信息发布的标准的综合门户平台;
- 提供企业办公协作的平台;
- 建立一个风格统一的个性化业务信息与业务集成的平台;
- 整合现有系统,通过统一的企业信息门户展现;
- 提供对分布的、异构的信息整合的平台;
- 提供对用户管理、认证的统一平台;
- 展示企业的业务成果,宣传业务工作内容,扩大企业在各方面的影响力,更好的为客户提供服务。

三、目标分析

实现上述目标,企业将会实现客户显示逻辑的整合,强大的办公协作和信息发布,数据信息整合平台,和用户管理和安全认证的集中化等。特别是应该已综合门户系统的构击。根据我们的经验,这些我们应该分期分批的逐步实现。这样,即降



低了投资风险,又缩短了项目周期,很快得到投资回报。那么,根据系统的关联和依赖性,我们应以综合信息门户平台建设,和办公协作、信息发布为二期建设的重点。而数据整合需要面向多个异构和分布的数据信息系统、集中认证则需要汇总用户信息和规范安全策略。所以,我们建议放到二期中实现。但是,我们在综合门户系统建设中,需要考虑到集中认证的需求,从而为后续工作做好铺垫。

四、项目中采用的 IBM 产品

WebSphere 企业版(J2EE 应用服务器)
WebSphere Portal Server
DB2 UDB 8.1
Lotus Domino
IBM SecureWay (LDAP 服务器)

SameTime (电子会议)
QuickPlace (小组工作室)
Portlet 套件
WSSD+Portal Toolkit
Collaboration Center
Tivoli WebSite Analyzer
Domino 套件
DB2 Information Integrate
IBM Tivoli Access Manager

五、综合信息系统方案概述

企业综合信息平台系统是一个提供个性化和适应性接口的软件系统,用户通过这个接口能够找到相关的人、应用程序和信息内容。用户使用不同的用户身份登录门户后,可以根据自己的角色查看相应的资源,并且透过门户可以访问到多种异构

数据源和应用,包括关系数据库、ERP系统、文档管理系统、日常办公模块、信息发布系统、 workflow 模块、协作模块和网上培训等内容。用户不仅可以定制私人工作区快速访问自己感兴趣的信息,而且还能方便地参与到群组工作区实现实时和异步的协同工作。

企业各级领导每天需要的信息量非常多,同时不同人员需要的内容又不尽相同,对不同级别的员工还需要有不同的权限控制,即需要一个容易定制的个性化服务。另外,在真正的业务运行环境中,除了需要充分的信息以外,还需要提供一个完善的协作服务,以便最终领导的信息决策。因此建立企业综合信息平台

(一期)应该达到如下的功能:

- 通过企业综合信息平台系统,建立集文档、数据管理于一体的信息数据管理平台,实现 Web 方式信息的动态整合、发布与交流;

- 通过采用先进的技术,实现高效快速的信息检索、友好的信息维护管理、分级灵活的用户登录、会员管理等功能,做到扩展自如、使用便捷、内容丰富、功能齐全,达到综合信息平台系统的先进水平;

- 为企业的办公协作、公文管理和共享提供一套技术实现和安全运行管理的单一的场所;

- 提供一个柔韧性好的门户构架,集成有序和无序的信息,为用户提供一个信息共享入口;

- 为企业内部的信息使用者提供必要的协作工具和服务,供他们快捷获取和交流所需信息;

- 为企业的信息发布和集成提供统一的渠道,将内部站点、后台的数据仓库、

ERP、数据中心、文档管理、项目管理以及其他信息系统有机集成;

- 针对办公协作、公文管理等新建项目实现用户统一管理规划、集中认证、单点登录;

- 提供个性化设置的功能,每个用户或部门可以定制自己的门户;

- 系统具有良好的可扩展性和可伸缩性,能满足不断发展的综合信息平台应用的需求,并为将来的数据整合和集中认证做准备。

六、企业综合信息平台功能介绍

WebSphere Portal 提供搭建及部署

具有扩展性的门户网站所需的基础架构,使用户通过 Portal 可以随时、随地、按需访问信息,具体功能如下:

内容发布:提供 Web 内容管理解决方案,它给予非技术

专业用户对发布到门户网站和 Web 站点的内容的更大控制权。

众多的用于连接后台应用的 Portlets:系统拟采用三层构架,运行在先进的应用程序服务器之上。除了能给用户丰富的、有价值的服务和内容,为了满足门户网站服务内容类型的扩展性和不可预期性,门户系统应该被设计成一个平台框架,基于这样的系统基础结构,可以将企业今后的扩展服务方便的加入门户网站。

集成优秀的开发工具:提供 J2EE 为基础的开发环境;提供 Web Service 为基础的开发环境;提供网页、网站建设的人性化编辑环境;提供 XML 的开发环境。

日志管理与日志分析:JLog 是 IBM 框架的一部分,它提供许多日志和跟踪能力。WebSphere Portal Server 使用此工

具包来帮助操作、维护门户网站及诊断其故障。

多个 Portlet 协作功能:click-to-Action 提供了简便、菜单驱动的方法,以在 portlet 之间传送兼容数据,这消除了用户手工将数据从一个 portlet 输入到另一个 portlet 的需要,并避免了在目标 portlet 上发现与当前页面上的源兼容的操作所需的一般常识。

协同工作:协作功能部件帮助您的组织中的人员一起联机工作并共享信息来完成他们的业务目标。

多用途框架:WPS 能够以任何类型的标记语言编制门户页面,用户可以通过桌面浏览器和移动设备进行访问。

个性化服务:本系统有必要针对不同的用户提供不同的服务内容与接口,而这种个性化服务不是简单的内容过滤,它是要在应用层面实行个性化。

门户系统管理功能:IBM WebSphere Portal Server 提供了丰富的 portlet 对门户系统进行管理和定制,用户只需使用浏览器,就可以完成门户系统的管理和维护操作。

安全管理:安全管理子系统允许管理员通过设置何人可对系统做何种操作的限制来保护资源。

认证:用户可以在进入系统时立即标识自己,或在用户未标识自己的身份而尝试访问受保护的资源时由系统提问。

单点登录:WPS 中的提供对单点登录的支持机制,该机制辅助 portlet 检索用户的已认证身份的任一表示法,然后 portlet 可以将它传递给后端应用程序。

授权:WPS 支持资源的精细访问控制,定制期间,只允许用户选择和查看他们具有访问权的那些资源。



WebSphere Business Integration Messenger Broker 在企业信息交换中的应用

王庆磊 / 文

行业：快速消费品

应用软件：WBI Messenger Broker

WebSphere Business Integration(简称 WBI) Message Broker(本文中简称 MB) 是 IBM 的应用整合中间件, 是 IBM WebSphere 业务整合解决方案的重要组成部分之一, 用于企业应用整合领域。

行业需求:

当前, 消费品行业的快速发展带来前所未有的、更多的管理层次、更多的跨地域网点、更复杂的数据管理需求。企业原有系统普遍面临的困难和压力是业务模式的快速变革、并购重组带来的中央运营和控制、系统整合、IT 集成、项目管理和维护等等。显然, 如果这些问题单靠企业自行规划和开发是难以解决的。

功能描述:

Message Broker 首先保证在一个异构的环境中实现信息稳定、可靠的传输, 屏蔽掉用户实际中的硬件层、操作系统层、网络层等相对复杂、烦琐的界面, 为用户提供一个统一、标准的信息通道, 保证用户的逻辑应用和这些底层平台没有任何关系, 最大限度地提高用户应用的可移植性、可扩充性和可靠性; 最重要的是它提供一个基于 Application-Hub 的先进应用整合理念, 最大限度地减少应用系统互联所面临的复杂性。基于 WBI Message Broker 系统的实现维护都相对简单, 保证每一个应用系统的更新和修改都能够实时地实现, 真正体现了应用整合的精髓, 同时当新的应用系统出现时能够简便的纳入到整个 IT 环境当中, 与其它的应用系统相互协作, 共同为用户提供服务, 是我们实现企业应用互联和应用整合的最佳实现方案。

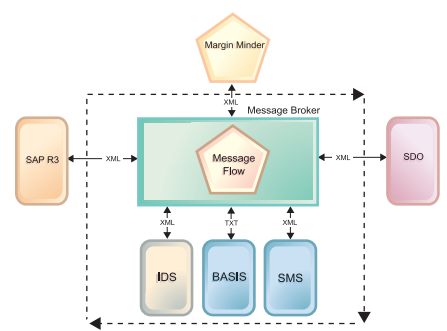
由于“Hub&Spoke”模式的采用, Message Broker 可以将

复杂的网状结构变为星型结构, 大大简化系统配置; 它为各种应用提供一个统一接口, 从而大大减少系统间接口的个数; 同时, 它可以作为一个数据中心, 提供各种数据处理服务, 如: 数据的计算、过滤、数据库操作等; 它可以实现各种不同数据格式之间的转换, 如: 自定义格式、传统数据格式与 XML 格式之间的转换, 针对不同系统所处理的消息格式各不相同的特点, 它提供了专门的消息格式解析器在不同的消息格式之间按照预先定义好的转换规则进行自动的格式转换, 然后将结果自动路由到目标应用系统。它提供强大的连接性, 利用各种适配器, 可以与多种应用系统进行无缝连接, 如 SAP, Siebel, Notes, SWIFT, People Soft, I2 等。

成功案例:

某饮料企业在实施 ERP 项目中, 由于新旧应用系统各自的平台相对独立, 数据格式不相同, 造成决策支持数据缺乏, 影响数据的一致性、及时性和可靠性, 影响管理深度和细度, 使企业难以迅速获得管理决策中所需的准确信息。各系统间的数据转换以及共享是企业信息整合过程中一个巨大的障碍。通过运用 IBM WebSphere Business Integration Messenger Broker 产品, 为各个应用系统制定统一的接口, 解决了上述出现的问题。

三石汇通根据企业各系统之间的应用关系, 共设计了包括销售订单、断货、短信查询、客户主数据、外部派车等接口, 为企业各系统的信息交换和共享提供了稳定强大的支撑。提高了整个企业计算机管理系统、软件应用系统的集成度, 解决了信息“孤岛”现象, 实现了企业内外资源及信息的充分共享, 从整体上提高企业对市场迅速作出反映的能力。右图为 WBI Messenger Broker 在企业应用中的系统结构



(王庆磊 北京三石汇通信息技术有限公司 总经理)

七、技术方案总结

本方案的设计, 以完整地实现企业信息平台的建设的一期目标为基点, 并充分

考虑整个系统在未来应具备稳定优异的性能和良好的可扩展性及可维护性。

通过技术方案的实施, 将在网上建成一个完整的数字化的信息整合集服务

系统, 并将原有的管理和服务进一步完善, 大大地提升传统的管理和服务品质, 将各个环节的进一步数字化作为系统的重要目标。■

浙江移动网络投诉处理综合平台解决方案

杭州东方通信软件技术有限公司 / 文

一、现状及背景

随着移动通信技术的飞速发展，移动网络规模越来越大，越来越复杂，投诉处理范围也相应扩大，对网络运维和处理网络类投诉带来了极大的挑战。除了发现各种故障、性能的实时告警，加大对网络的监控外，及时处理用户的投诉，更是解决网络问题、提高用户满意度的重要手段。目前浙江移动网络投诉处理综合平台一期工程通过对各种不同的移动业务支撑投诉系统进行整合，将分布于不同网段、不同地理位置的旧有投诉处理支撑系统中的常用投诉处理功能统一整合到网络投诉处理综合平台当中，初步实现了一个统一有效的网络投诉处理综合平台，很大程度上提高了解决投诉处理问题的效率，并提升了客户的满意度。

二、项目中采用的设备及软件部署

1、项目中采用的硬件产品

应用服务器：IBM P55A (8CPU@1.5G Power, 24GB RAM, 2 × 73GB)

数据库服务器：IBM P55A (8CPU@1.5G Power, 24GB RAM, 2 × 73GB)

WEB服务器 (2台)：IBM P505 (2CPU@1.5G Power, 4GB RAM, 2 × 73GB)

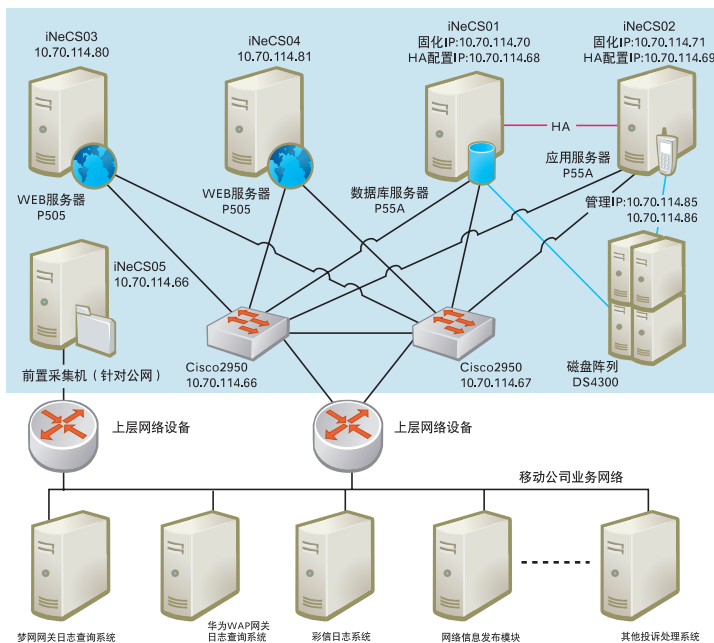
前置采集机：IBM X346 (2CPU@3.0G Xeon, 5GB RAM, 146GB)

磁盘阵列：IBM DS4300 (12 × 300GB RAID5)

交换机 (2台)：Cisco 2950 (24 × 10/100Mbps)

2、硬件设备实际组网 (包括 IP 地址占用)

- 核心服务器包括WEB服务器、应用服务器和数据库服务器。
- 应用服务器和数据库服务器实现HA高可用性机制，通过IBM自带HACMP软件将两台硬件设置成Dual-active互备援方



式。双机都安装所有的软件(除采集服务器软件、WEB软件外)，双机均连接至CISCO 2950以太网交换机。

- HA互备援机制通过光纤共享IBM DS4300磁盘阵列。
- 两台WEB服务器对外提供WEB服务，安装IBM WAS软件，以及HTTPSERVER和分发器软件，直接连接至CISCO 2950以太网交换机。
- 前置采集服务器安装RHEL4.2系统，部署东信自主开发EC采集探针和数据预处理分析程序，主要采集WAP网关日志、彩信日志。

3、项目中采用的IBM软件产品

MessageBroker6.0

DB2 数据库
MQ6.0
WAS6.1

三、投诉综合平台的软件架构

浙江移动网络投诉处理综合平台在二期建设时,存在被整合的系统较多、复杂性较高的特点和难点,为了提高开发效率和资源利用率,平台采用企业信息集成体系架构方法论 SOA (Service Oriented Architecture),分析各投诉处理支撑系统,进行面向服务的模型建设和系统整合。

为了使系统具有可移植性和扩展性的特点,浙江移动网络投诉处理综合平台在二期建设时采用了组件化的设计理念,系统包括应用组件、服务组件和数据组件。应用组件满足投诉处理综合平台用户功能需求;服务组件提供各种形式的系统间的数据交互接口;数据组件实现业务数据的收集与采集,给应用组件和服务组件提供数据来源。

通过适配器 (JDBC 适配器、消息适配器、HTTP 适配器、文件适配器、API 适配器等) 把各个待整合系统整合进系统,并实现服务封装,封装好的服务通过ESB总线向上层应用或第三方应用提供服务。

详细系统软件架构图如右所示:

系统实现的价值

●浙江移动网络投诉处理综合平台一期工程已为投诉处理工作人员提供一个统一的投诉处理综合平台,提高了投诉处理效率,降低了用户的二次投诉,提高了用户满意度。

●浙江移动网络投诉处理综合平台一期工程,整合了原先12类投诉处理支撑系统,达到了划一和平整,增强了功能。平台的一期建设既有效利用原有系统,保护了投资,同时又有创新、发展。

●浙江移动网络投诉处理综合平台一期工程强调系统平台化的整合,使得新系统在易用性、便捷性上得到了较大的提高,贯彻了“投诉前移”的思想。

●资源的重复可利用性,网络投诉处理综合平台一期建设方案采用业界最先进的SOA方法论

进行设计,整体对外以服务方式、可重用组件方式提供。

四、浙江移动网络投诉处理综合平台可实现的功能

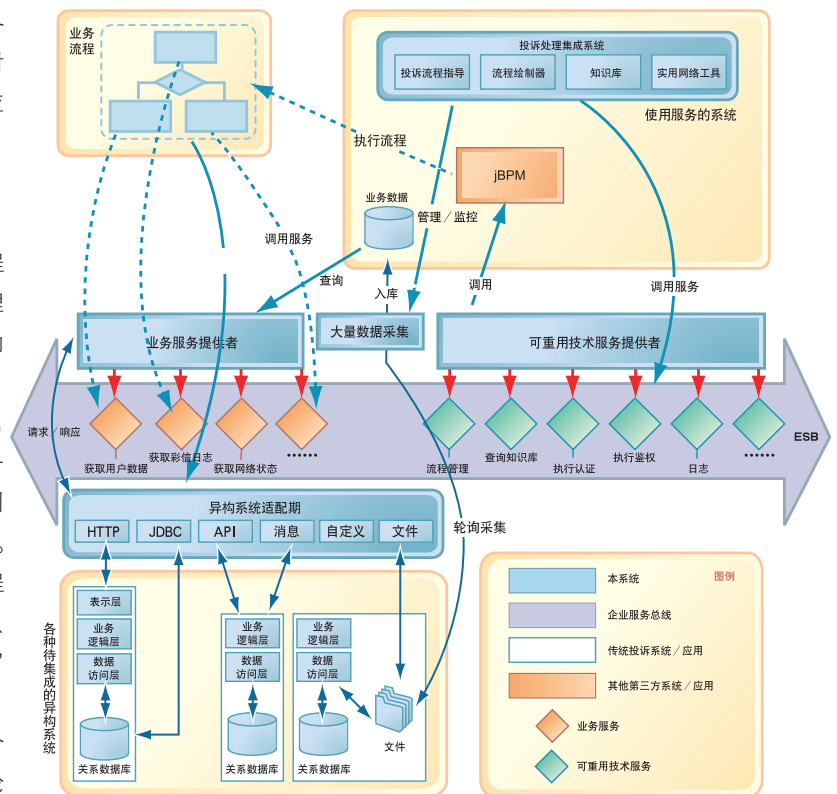
1) 投诉处理向导: 主要呈现各类投诉处理向导图和拓扑图。目前已经梳理出40多个投诉处理向导,划分成四类: 漫游障碍类、通话障碍类、数据业务障碍类、计费类。

2) 投诉向导管理: 该模块对投诉向导进行管理,包括向导的绘制、发布、向导分类、向导状态启停和删除向导。

3) 单点登录模块: 投诉处理综合平台提供到目标投诉处理系统的链接及单点登录,用户可以在平台中启动并切换到对应的投诉处理系统。

4) 应用工具模块: 提供打开或配置用户常用的一些第三方工具或软件。功能包括: 工具启动、工具路径配置。

5) 系统管理模块: 包括用户管理,角色管理,权限管理,日志管理,系统基础数据配置,采集日志管理,数据备份与恢复功能。■



供应链管理系统解决方案

- 如何将自己的长期生产计划发布给供应商?
- 如何了解供应商在关键产能和 Long Lead time 的物料备料瓶颈信息?
- 如何将自己的短期生产计划调整信息发布给供应商?
- 如何让供应商按照流程安排实现及时送货?
- 如何控制、追踪、分析供货商的交货记录?
-

神州数码系统科技公司 · IBM 产品本部供稿

在以上这些问题的情况下，对供应链的反应速度要求就特别高，仅仅依靠手工作业不仅速度慢，而且容易出错，所以借助信息系统的帮助势在必行。

此供应链系统的重点是如何让企业的采购需求能够同供应商的供应进行快速平衡。通过使企业之间进行快速、安全的信息共享，整合企业间作业流程，从而提高供应链上企业间的协作，帮助企业有效地降低成本、提高盈利能力，进而取得竞争优势。

本方案的功能描述

供应商入口门户

提供了一个易用的、基于互联网的供应商入口门户，对于规模较小或IT程度较差的供应商也可以详尽而安全地访问其订单和计划信息。采购方和供应商之间有多种通信手段，包括7X24小时在线和消息通知，这实现了关键业务流程的实时协作和执行。

提供滚动的送货看板协作

连续、高频率的供货是JIT采购显著特点，也是采购作业的一个核心业务过程，大量的数据需要同供应商进行交换，因此采购方和供应商之间需要畅通无阻的通信和协作。通过供应商入口门户，供应商可以即时查看交货看板。供应商可以对交货看板进行确认，以表示其履行交货看

板的能力，并能够通过消息的形式同采购人员进行信息交流。如果企业的生产发生变动，其相应的物料需求状况会实时的反应到送货看板中，供采购方进行调整和供应商进行快速反应。对看板的变动信息，采购方和供应商都能够根据设定的权限进行查看，比较不同版本的修改情况和全部的历史记录。

提供闭环采购订单协作

采购订单处理是一个核心的业务过程，需要畅通无阻的通信和协作。通过供应商入口门户，供应商可以即时查看采购订单。供应商可以确认接受订单，以表明其履行订单的能力，并可以提交承诺交货日期的变更订单请求。如果由于生产环境或其它市场条件发生变化而必须修改采购订单，供应商可以在线发送一个更改承诺日期的申请，以及对此更改的解释。提交后，更改请求被发送到采购方的采购部门以获得批准。

发货通知 (ASN)

ASN (Advance Shipping Notice) 使采购方可以直接查看针对特定交货看板和采购订单的供应商的发货情况。通过供应商入口门户，供应商可以在线输入ASN。这些信息会导入到采购方的ERP系统内。采购方当货物运到时，接收人员能够快速准确地完成收货过程。实时的发货通知与

简化的接收过程的结合使采购方和供应商能够在可能的交付问题出现之前就预先做好处理准备。要提交ASN，供应商只需通过简单的步骤，选择交货看板或采购订单以及相关的货物数量。

收货和质检过程追踪

采购方和供应商可以按照其采购方所作的记录查看采购方收货历史信息。供应商可以查看交付和到货验收方面的情况。供应商还可以查看质检情况以及质检不合格的原因。所有这些能够帮助供应商获得更好的客户满意度。

对帐和付款过程

对帐和付款过程自动化对于提高效率是极其重要的。供应商可以直接通过供应商入口门户查询如对帐清单、付款清单、送货清单、报关单等，大大增加了采购方的采购和付款部门的间接管理费用。通过供应商入口门户，供应商也可以直接得到各种问题的完整数据，而无需使用电话、传真和电子邮件。

按任务行动

供应商希望能迅速得到问题警报，以便能够同采购方进行更好的协同。供应商入口门户具有内置的任务功能，因此供应商可以查看并按照他们收到的任务来行动。通知可能包括计划的变更、发出的新采购订单、采购订单变更、送货看板变更等。供应商可以使用待办事项功能轻松而迅速地找到其所需信息。

异常状况监控

采购方希望能迅速得到问题警报，以便在发生意外状况时能够前瞻性地采取措施。入口门户具有内置的示警功能，因此当意外状况发生时，采购方能够迅速得到示警信息，提示其进行相应的处理。示警包括订单将要超出确认时间、送货看板将

要超出确认时间、库存 Balance 异常、送货数量短缺、质检合格率过低等。采购方能够通过 Alert List、Mail 和短消息迅速获知示警信息。

查询功能

● 查看接收和质检的历史信息：

供应商可以按照其采购方所作的记录查看采购方收货历史信息。供应商可以查看他们在按时交付和货运验收方面的情况。供应商还可以查看质检的历史信息以及质检不合格的原因。所有这些能够帮助供应商获得更好的客户满意度。

● 查询供应商库存

通过在线界面，供应商可以通过录入或数据上传的方式提供自己的库存情况。采购方可以根据供应商提供的数据来实时了解供应商的备货情况，是否达到了双方的约定。

● 可查看存货量并支持 VMI 过程

通过在线界面，供应商可以查看采购方仓库、Hub 或其它存货地点的存货量。另外，该应用程序还支持供应商管理库存 (Vendor Managed Inventory, VMI) 过程，这使得供应商能够管理特定货物的库存。供应商可以查看当前库存余量和正在处理的货运事务。供应商可以检查其最小和最大存货量、再订购点和补给量。通过在线界面，供应商可以通过采取补给措施来确保不会断货。

带给客户的收益

此系统是一套企业与其战略供应商之间的协同采购平台。系统基于因特网技术，可实现 VMI (供应商管理库存)、Consignment (寄售)、JIT (即时采购) 和电子订单采购等多种采购管理模式。将企业的原料和零部件采购计划与生产排程计

划直接集成，使得企业生产与采购无缝对接，改变按库存采购的原料供应模式，实现原料的零库存和即时供应。

此供应链系统还能够帮助企业与其供应商建立紧密的战略伙伴关系，实现双方在线业务协同和信息共享；缩短采购供应周期，降低总采购成本，提高供应链对市场和客户需求响应的敏捷性。

系统为企业提供了供应商入口门户，基于 B/S 的多层结构和 XML 技术，实现企业与供应商数据的双向在线传输。供应商只需要通过因特网登录，即可与企业对送货看板进行协同；企业通过采购平台，将生产计划转换为对每个供应商的送货看板，保证各供应商的原料或零部件供应与企业生产计划协调一致。

方案的独特卖点

支持多种采购管理模式

系统支持传统订单方式的采购。ERP 中的采购订单 (PO) 以电子订单方式在系统中发布、处理，供应商通过系统进行电子订单的确认，修改和送货。

系统支持 JIT 的采购方式。由企业的生产投料计划和库存信息直接自动生成对每个供应商的原料供应计划，双方对供货计划进行协商确定，供应商按供货计划进行连续生产送货，直接满足企业的生产需要。

系统支持 VMI/Consignment 的采购供应模式。企业在生产地建立原料 HUB，由供应商对 HUB 原料进行管理，根据企业的生产计划与原料消耗信息进行生产和供应，满足企业的生产需要。

以送货看板为中心的协同采购模式

系统不是简单地实现传统的采购管理功能，而是以 e 化采购的方式实现供应商

与企业的紧密集成，供应与生产的无缝对接，其核心是基于 XML 技术实现的供货计划，以“一个计划”统一采购供应流程。供货看板可由企业生产计划按原料供应规则直接生成，每个供应商通过一个送货看板来与企业协同，企业和供应商在“一个计划”上实现符合双方特点的采购供应模式。

主动知会、示警

系统对整个交易过程实施了多种监控，例如供应商对订单的确认时间、送货看板确认时间、送货通知 (ASN) 是否按时开出、供应商出货状况是否异常等等。通过用户自定义的管理规则来对整个交易过程进行监控，系统会根据用户定义的规则，在异常状况发生时自动通知相关人员。

基于企业门户的供应链采购平台

相对与第三方交易平台和基于卖方的电子市场，系统以买方企业为中心，建立了基于企业门户的采购平台，供应商根据自己的登录帐号进入企业门户，进行自助式服务，并可与企业就送货看板进行实时交互。

基于因特网的技术架构

系统采用 Web 技术，以企业为中心建立了供应链采购平台，支持 B/S 多层结构，企业内部部署 Web 服务器，应用服务器和数据库服务器，供应商通过浏览器访问企业采购门户，以 XML 方式在因特网上传输送货看板数据。系统用户超越了地域和终端的限制。

推荐的软硬件

硬件平台：	IBM System P52A
软件平台：	IBM DB2 UDB, IBM WebSphere Application Server
操作系统：	AIX 5L
适用的行业：	流通行业

PA 与 WebSphere 的兼容性问题的解决方案

软通动力资深架构师 李俊杰 / 文



概述

考虑到 OR 映射的产品多样化，为了更富有柔性化的设计，我们的产品采用 JPA 作为 Persistence layer 的持久化技术，hibernate3.2 (hibernate3.2 才支持 JPA)，spring2.0 作为 IOC 和 AOP 的工具。产品的运行环境为 Websphere6.1 (因为我们的产品是基于 jdk1.5)。JPA (Java Persistence API) 作为 Java EE 5.0 平台标准的 ORM 规范，得到所有 Java EE 服务器的支持，是一个易于使用、伸缩性强的 ORM 规范。目前 Hibernate 3.2、TopLink 10.1.3 以及 OpenJpa 都提供了 JPA 的实现。

错误信息

```
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'entityManagerFactory' defined in ServletContext resource [/WEB-INF/context/rdf/applicationContext-dao.xml]: Invocation of init method failed; nested exception is java.lang.ArrayStoreException
```

错误原因

这个错误是 IBM JVM 的 bug，同样的程序在 tomcat5.5，和 jboss4.0.3 上都没有问题，迁移到 WebSphere 上就有如上的错误。

解决方案

下载 WebSphere 软件升级包，升级 WebSphere。

下载 WebSphere 升级软件

地址：<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg24012718>，下载 download.updii.61017.windows.ia32，可以根据具体的操作环境和 WebSphere 版本下载相应的升级软件。

下载支持操作系统及 sdk 的补丁

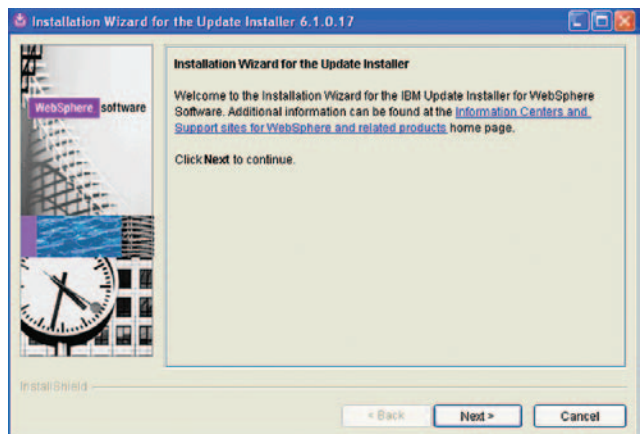
地址：<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27007951>，下载操作系统的补丁：6.1.0-WAS-WAS-WinX32-FP0000017.pak，根据具体的操作系统下载相应的补丁；

下载 sdk 补丁：6.1.0-WAS-WASSDK-WinX32-FP0000017.pak。也要根据具体的操作系统和 sdk 要求来下载。

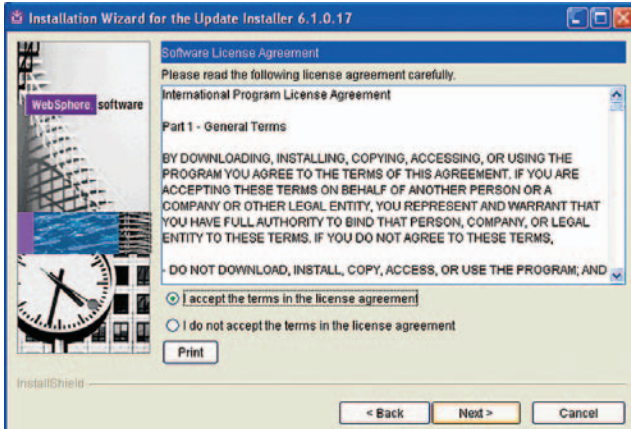
升级 WebSphere

升级 WebSphere，这儿是使用 Windows 的升级程序来作为例子。

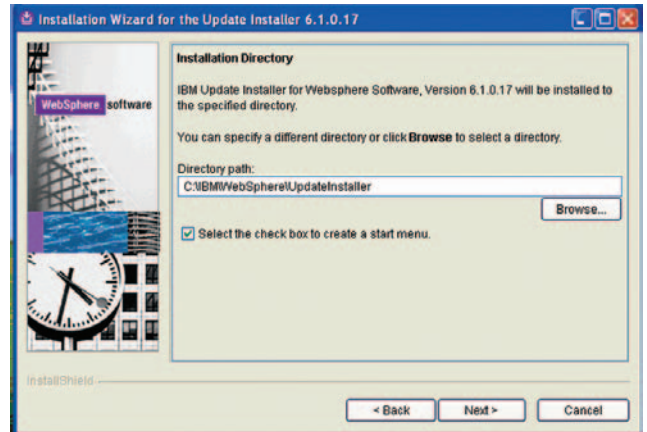
1) 解压并到 download.updii.61017.windows.ia32\UpdateInstaller 目录下，点击 install.exe



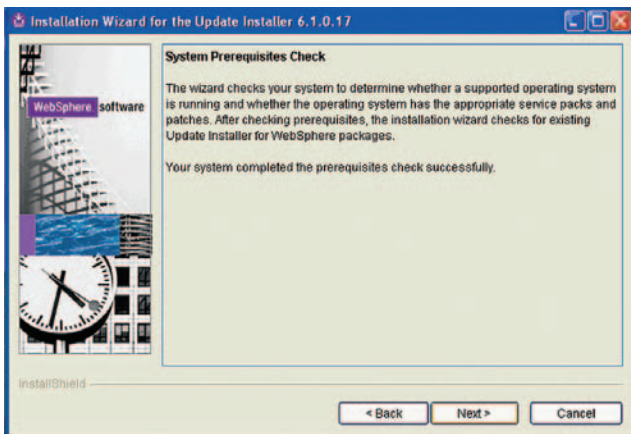
点击 next, 选中 I accept 选项:



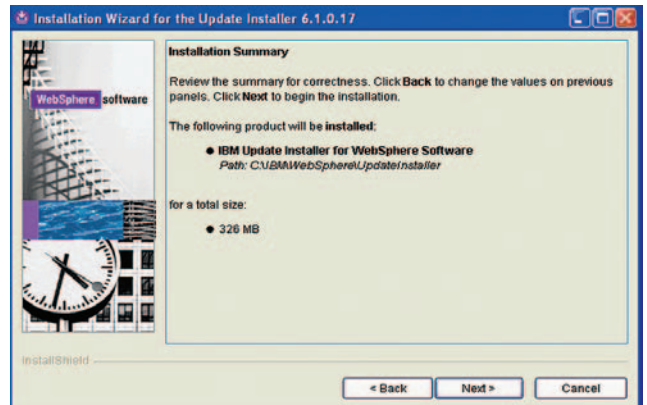
下面的地址你不用修改, 是升级软件自动给出的, 升级软件会创建这个目录并把相关需要的文件 copy 到该目录下:



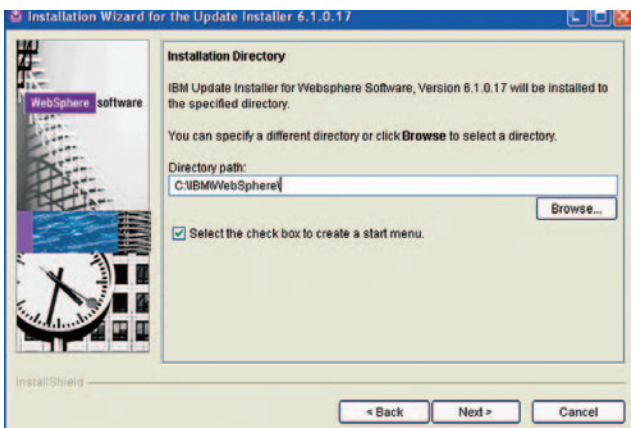
校验此升级软件是否支持该操作系统:



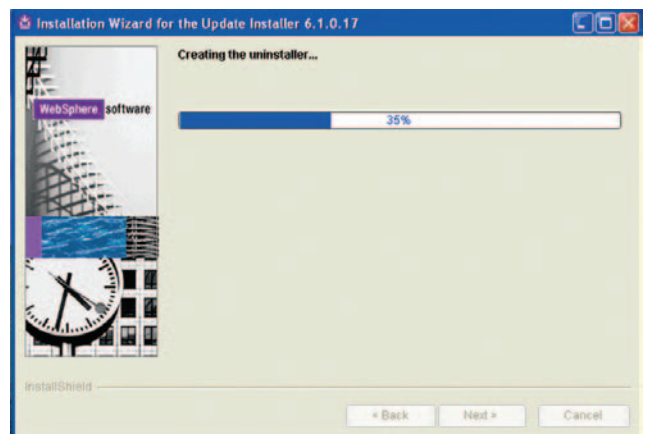
确认升级包安装地址及大小:



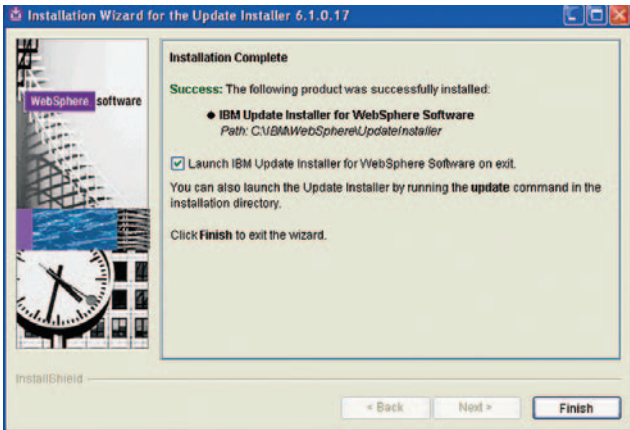
输入 WebSphere 安装地址:



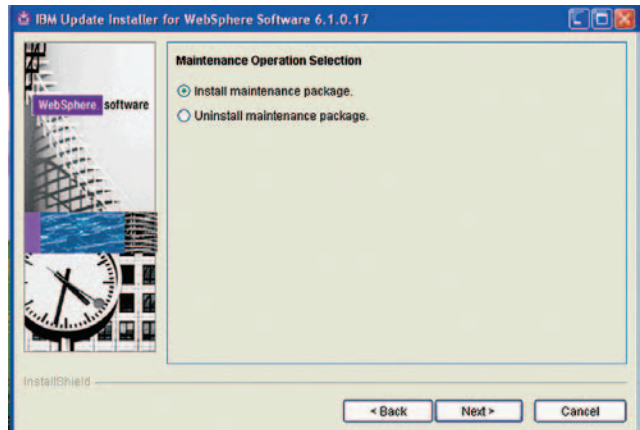
创建升级目录并把相关文件 copy 到该目录下:



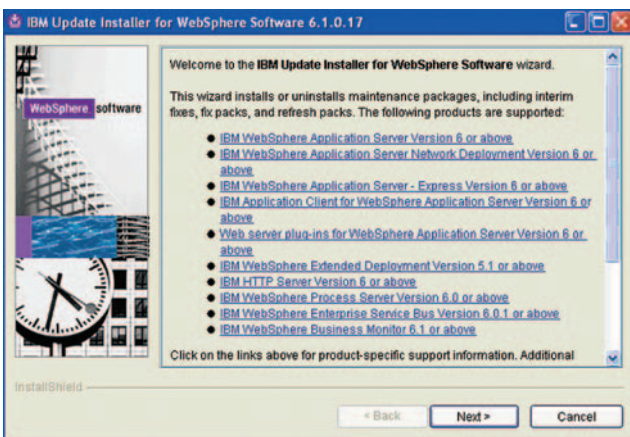
升级软件 copy 文件结束:



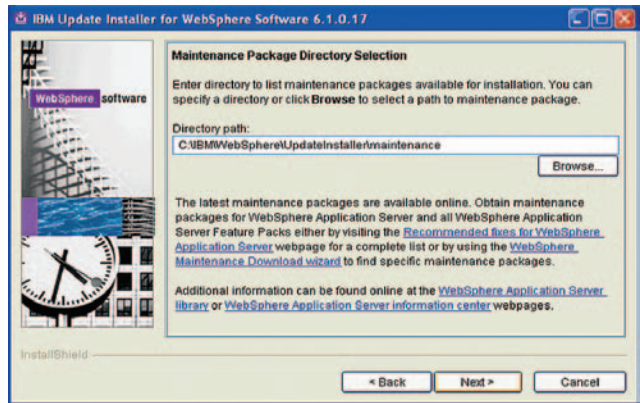
默认选择安装升级包:



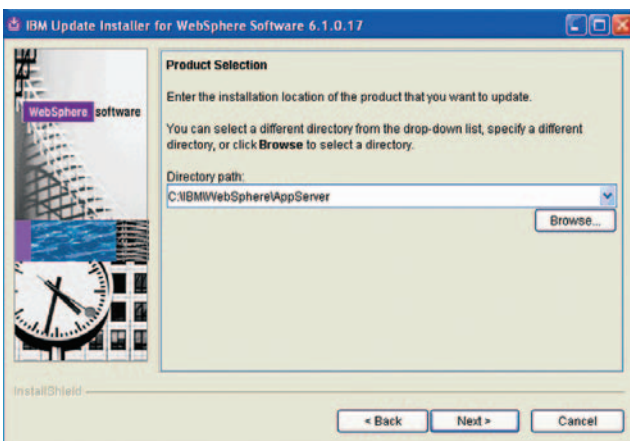
2) 点击 finish, 则开始安装具体的升级包:



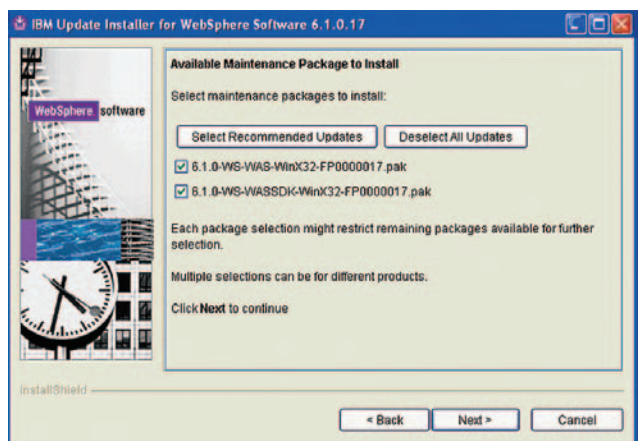
输入那两个升级补丁所在的位置, 最好把这两个升级包 copy 到如图的目录下:



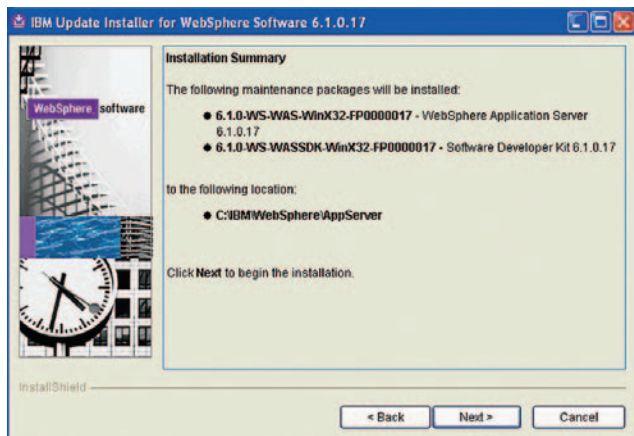
输入要升级的 WebSphere 的 AppServer 目录:



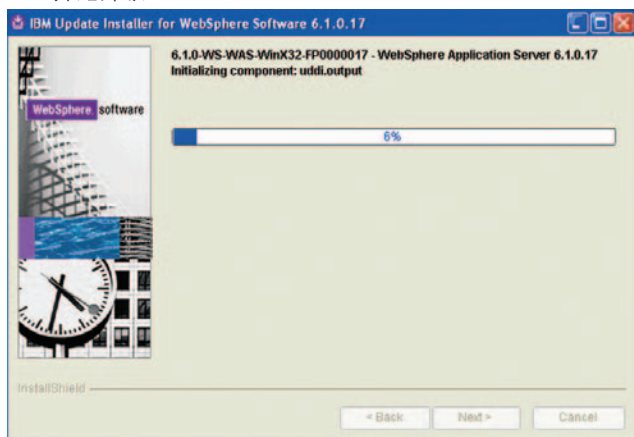
选择两个需要升级的包:



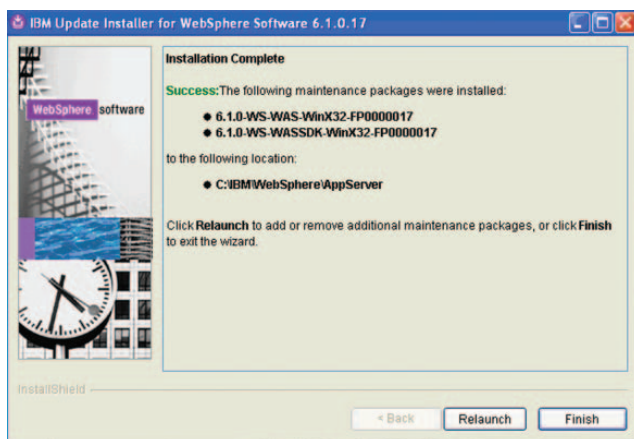
确认升级包和升级的应用服务器



开始升级:



升级成功结束:



这样重新启动 WebSphere，应用系统正常运行。

努力，在于我热爱我的事业，与中国的软件一起走向成熟，走向世界。■

知识连接 >>>

什么是 WebSphere Application Server?

基于Java的应用服务器，整合企业数据和事务以用于动态电子商务环境，包括高级集群能力和Web服务特性。

WebSphere Application Server V6.1 是 IBM WebSphere 软件平台的基础。它提供了面向服务架构(SOA)所需的安全、可伸缩、富有弹性的应用程序基础设施。

通过新的特性包简化了使用 WebSphere Application Server 6.1 的方法。您可以下载最新的开放标准和创新技术。

利用业内主要的 Java2 Enterprise Edition (J2EE) 1.4 和 Web 服务应用程序平台来帮助构建、运行、整合和管理动态的按需应变的业务应用程序。

通过一组简单的工具和接口，迅速轻松地构建和部署可重用的应用程序服务。

在可靠、可伸缩、高度可用的环境中运行服务，确保不会因应用程序宕机而错失商业良机。

以富有弹性、基于标准的安全性基础设施保护应用程序和数据，这样的基础设施将消除漏洞隐患，同时最大化开发人员生产力。

通过基于标准的信息传递和最新的 Web 服务标准来重用软件资产并扩展其范围，使用强大、易用的管理和监控工具轻松管理应用程序。

随着需求的演进而发展，利用核心资产和技能，现有数百种 IBM 和业务合作伙伴产品均构建于 WebSphere Application Server 的基础之上。

支持的操作系统: AIX、HP Unix、i5/OS、Linux、Sun Solaris、Windows。■

IBM WebSphere Edge Component

做网络负载均衡器的实施案例



安装环境：RedHat Enterprise Linux AS 4 Update 2 (相关IP地址说明见附录)

富通东方技术总监 张东焕 / 文

1. mount 光盘，执行 launchpad.sh
选择必要的组件 Load Balancer 安装即可。

2. 配置 Load Balancer (以下简称 LB)

我们只用到 Dispatcher Server

2.1 启动 LB，执行 dsserver

2.2 用图形界面管理工具 lbadm

2.3 配置内容

2.3.1 主节点 lb1

```
dscontrol set loglevel 1
```

```
dscontrol executor start
```

```
dscontrol cluster add lbcluster address 10.9.162.100 primaryhost 10.9.162.41
```

```
dscontrol cluster set lbcluster proportions 49 50 1 0
```

```
dscontrol port add lbcluster:80 re-set no
```

```
dscontrol server add lbcluster:80:WAS2 address 10.9.162.44
```

```
dscontrol server set lbcluster:80:WAS2 weight 10
```

```
dscontrol server add lbcluster:80:WAS1 address 10.9.162.43
```

```
dscontrol server set lbcluster:80:WAS1 weight 10
```

```
dscontrol port add lbcluster:443 re-
```

```
set no  
dscontrol server add lbcluster:443:WAS2 address 10.9.162.44
```

```
dscontrol server set lbcluster:443:WAS2 weight 9
```

```
dscontrol server add lbcluster:443:WAS1 address 10.9.162.43
```

```
dscontrol server set lbcluster:443:WAS1 weight 9
```

```
dscontrol highavailability heartbeat add 10.9.162.41 10.9.162.42
```

```
dscontrol highavailability backup add primary=10.9.162.41 auto 12345
```

```
dscontrol highavailability reach add 10.9.162.22
```

```
dscontrol manager start manager.log 10004
```

```
dscontrol advisor start Http 80 Http_80.log
```

2.3.2 备份节点 lb2

```
dscontrol set loglevel 1
```

```
dscontrol executor start
```

```
dscontrol cluster add lbcluster address 10.9.162.100 primaryhost 10.9.162.41
```

```
dscontrol cluster set lbcluster proportions 49 50 1 0
```

```
dscontrol port add lbcluster:80 re-
```

```
set no  
dscontrol server add lbcluster:80:WAS2 address 10.9.162.44
```

```
dscontrol server set lbcluster:80:WAS2 weight 10
```

```
dscontrol server add lbcluster:80:WAS1 address 10.9.162.43
```

```
dscontrol server set lbcluster:80:WAS1 weight 10
```

```
dscontrol port add lbcluster:443 re-set no
```

```
dscontrol server add lbcluster:443:WAS2 address 10.9.162.44
```

```
dscontrol server set lbcluster:443:WAS2 weight 9
```

```
dscontrol server add lbcluster:443:WAS1 address 10.9.162.43
```

```
dscontrol server set lbcluster:443:WAS1 weight 9
```

```
dscontrol highavailability heartbeat add 10.9.162.42 10.9.162.41
```

```
dscontrol highavailability backup add backup auto 12345
```

```
dscontrol highavailability reach add 10.9.162.22
```

```
dscontrol manager start manager.log 10004
```

```
dscontrol advisor start Http 80
```

Http_80.log

3. 配置 HTTP Server

在 /etc/rc.d/rc.local 追加一行
 ip addr add 10.9.162.100/32 scope
 host dev lo
 执行
 #sysctl -w net.ipv4.conf.all.
 arp_ignore=3 net.ipv4.conf.all.
 arp_announce=2

确认返回正常后, 在 /etc/sysctl.conf
 追加两行

```
net.ipv4.conf.all.arp_ignore=3
net.ipv4.conf.all.arp_announce=2
#shutdown -r now
```

4. 配置 HA 的相关脚本

复制 /opt/ibm/edge/lb/servers/
 samples 下的 goActive.sample
 goStandby.sample goInOp.sample
 highavailChange.sample

到 bin 下并改名成 goActive
 goStandby goInOp highavailChange

修改 CLUSTER=10.9.162.100
 INTERFACE=eth0:1 NETMASK=255.
 255.255.0

各脚本内容如下:

```
goActive
!/bin/bash
5639-D57, 5630-A36, 5630-A37,
5724-D18, 5724-H88, 5724-H89 (C)
COPYRIGHT International Business Ma-
```

chines Corp. 1999, 2004
 All Rights Reserved * Licensed
 Materials - Property of IBM

goActive script

Configure this script when using the
 high availability feature of

Load Balancer.

This script is executed when Load
 Balancer goes into the
 'Active' state and begins routing
 packets.

This script must be placed in Load
 Balancer's bin directory

(/opt/ibm/edge/lb/servers/bin/) and it
 needs to have root execute permission.

eth0=Ethernet

The netmask must be the netmask
 of your LAN. It may be hexadecimal or
 dotted-decimal notation.

CLUSTER=10.9.162.100

INTERFACE=eth0:1

NETMASK=255.255.255.0

echo "Adding device alias"

ifconfig \$INTERFACE \$CLUSTER
 netmask \$NETMASK up

goStandby

!/bin/bash

5639-D57, 5630-A36, 5630-A37,
 5724-D18, 5724-H88, 5724-H89 (C)

COPYRIGHT International Business Ma-

chines Corp. 1999, 2004

All Rights Reserved * Licensed
 Materials - Property of IBM

goStandby script

Configure this script when using
 the high available feature of
 Load Balancer.

This script is executed when Load
 Balancer goes into the

'Standby' state. Monitoring the
 health of the 'Active' machine

but not routing packets.

This script must be placed in Load
 Balancer's bin directory

(/opt/ibm/edge/lb/servers/bin/) and
 it needs to have root execute
 permission.

eth0=Ethernet

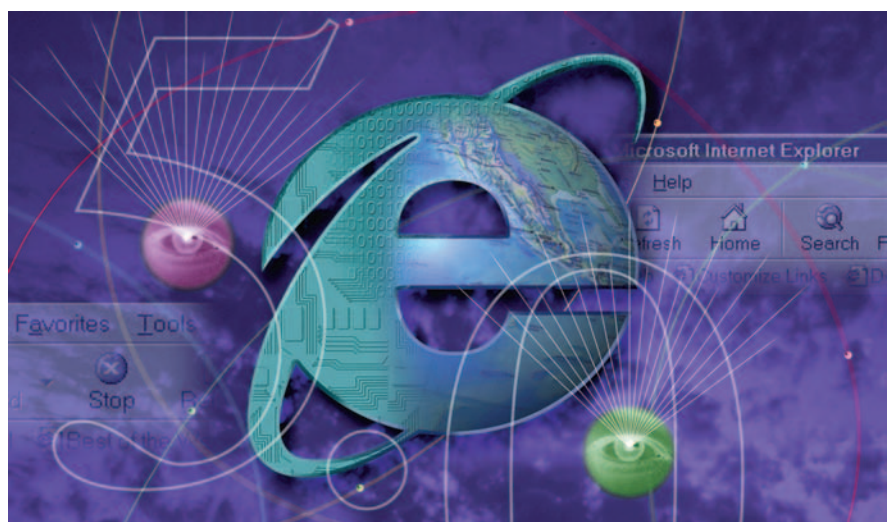
CLUSTER=10.9.162.100

INTERFACE=eth0:1

NETMASK=0xffff800

echo "Deleting the device alias(es)"

ifconfig \$INTERFACE down



```
golnOp
!/bin/bash
5639-D57, 5630-A36, 5630-A37,
5724-D18, 5724-H88, 5724-H89 (C)
COPYRIGHT International Business Ma-
chines Corp. 1999, 2004
```

All Rights Reserved * Licensed
Materials – Property of IBM

```
golnOp script
Configure this
script when using the
high availability feature of
Load Balancer and
optionally when using
Load Balancer in a
standalone
environment.
```

This script is ex-
ecuted when the Load
Balancer executor is
stopped

(and before the executor is initially
started).

This script must be placed in Load
Balancer's bin directory
(/opt/ibm/edge/lb/servers/bin) and it
needs to have root execute permission.

```
eth0=Ethernet
CLUSTER=10.9.162.100
INTERFACE=eth0:1
echo "Removing device alias(es)"
ifconfig $INTERFACE down
highavailChange
!/bin/bash
5630-A36, (C) COPYRIGHT Inter-
national Business Machines Corp., 2001,
2002
```



All Rights Reserved * Licensed
Materials – Property of IBM

highavailChange script
Configure this script only if you are
using the Highavailability
feature of Dispatcher.

Configuration of this script is
optional.

This script must be placed in
Dispatcher's bin direc-
tory

(by default this is /
opt/ibm/edge/lb/servers/
bin/) and it needs to have
root execute
permission.

This script is ex-
ecuted whenever the
Highavailability state
changes within
Dispatcher such that

one of the go* scripts is
called. The single parameter passed
to this script is the name
of the go* script just run by
Dispatcher.

This script may use this information

in many ways, such as to alert
an Administrator or simply record
the event. Below is an example
of how to record the data when
Dispatcher has detected a problem.

```
DATE=`date`
OUTPUT="$DATE LB just ran $1."
echo $OUTPUT >> /opt/ibm/edge/
lb/servers/logs/lb.log
:echo $OUTPUT | mail -s
"highavailChange" root@localhost
```

```
chmod 700 go*
chmod highavailChange
```

5. 测试 Load Balancer

分别停止 LB
#dsserver stop
先启动主节点的 LB
后启动备份节点 LB
主节点状态时是“活动”，集群地址在
这台机器上。

备份节点状态是“备用”
拔掉主节点的网线，主节点状态变为
“备用”，备份节点的状态变为“活动”，集
群地址在备份节点上。

接上主节点的网线，主节点状态变为
“活动”，备份节点的状态变为“备用”，集
群地址在主节点上。■

附录：相关 IP 地址说明：

Load balancer 对外提供的 IP 地址：10.9.162.100
Load balancer 主节点的 IP 地址 10.9.162.41
Load balancer 备份节点的 IP 地址 10.9.162.42
IBM HTTP Server 的地址分别是 10.9.162.43 10.9.162.44
10.9.162.22 是用于测试 Load balancer 是否处于激活状态的 IP 地址。这是数
据库服务器的服务地址。

WebSphere MQ

消息通道的配置和维护介绍

中联集团系统软件技术部总监 甘 荃 / 文

概述



WebSphere MQ 作为 IBM 软件家族的消息传输中间件产品，以其出色的特性和功能在业界享有盛誉。

WebSphere MQ 独特的安全机制、简便快速的编程风格、卓越不凡的稳定性、可扩展性和跨平台性，以及强大的消息通讯能力，使得它在银行、电信，还是在交通运输、政府机关等各行各业，赢得了很高的市场份额。

在中国，WebSphere MQ 同样拥有广泛的用户基础和许许多多的成功案例。它不仅具有跨平台、跨网络的特性，而且以其特有的先进机制保证对消息的 "Once and Once only" 的传输，做到不丢失、不复传。在 WebSphere MQ 给客户带来的众多价值中，有一点十分重要，就是它的通讯感知和恢复机

制，尤其适用于我国目前的现状，在我国很多地方存在网络线路质量差，网络状态不稳定的现状。因为 WebSphere MQ 在支持同步通讯的同时，提供了基于消息队列存储-转发机制的异步通讯模式，应用程序只需将消息交给 WebSphere MQ，就由 WebSphere MQ 负责将消息安全、可靠地发送出去，不再需要应用和人工的干预，当网络出现故障的情况下，或对方主机发生故障时，WebSphere MQ 能够作到不需要人工干预，自动探测网络状况的好坏，并且在网络恢复正常之后能够继续正常工作，也即断点续传。

在 WebSphere MQ 的系统配置和维护中，通道(Channel)的配置和维护是较复杂也是最重要的部分，本文将对如何配置和维护 WebSphere MQ 消息通道进行介绍，并附录有关实现 WebSphere MQ 队列管理器双向通讯对象配置脚本。

1. 消息通道的配置

1.1 消息通道的类型

消息通道是把消息从一个队列管理器送到另一个队列管理器的通道。不要和 MQI 通道混淆，MQI 通道是在 MQ 客户端和 MQ 服务器队列管理器之间发送消息的通道。我们这里讨论的是消息通道。WebSphere MQ 中通信双方的消息通道类型有 6 种：

- Sender
- Receiver
- Server
- Requester
- Cluster sender
- Cluster receiver

上述消息通道类型配对共有 5 种：

- Sender-receiver
- Requester-server
- Requester-sender(callback)
- Server-receiver
- Cluster sender-cluster receiver

下面分别对各种组合进行介绍，并给

出配置脚本：

1.1.1 Sender-receiver

Sender/Receiver 通道是最常见的通道配置方式，Sender 作为通道的发送方也是通道连接的主动发起方，Receiver 作为通道的接收方也是通道连接的被动监听方。在 Receiver 端要配置并运行相应的侦听程序，如图 1 所示：

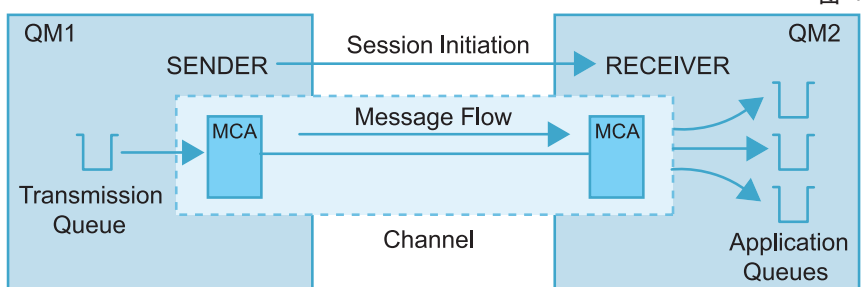


图 1

在下面配置脚本中，通道连接两个队列管理器 QM1 和 QM2。其中，QM1 为 Sender 方，QM2 为 Receiver 方。在 QM1 上配置了远程队列（QR2）和传输队列（TO.QM2），其中 QR2 指向队列管理器 QM2 上的本地队列（QL2），且 QR2 与 TO.QM2 对应，即凡是要放入 QR2 队列的消息，在加上传输消息头后直接放入 TO.QM2 中等待发送。QM1 上配置 Sender 通道（QM1.QM2）需要指定传输队列名以及对方的通信参数（IP 地址和端口），通信参数必须与 QM2 上的侦听程序设置匹配。

Sender 通道与传输队列 TO.QM2 对应，表示凡是在 TO.QM2 中等待发送的消息最终都可以由该通道送出。双方通道必须同名。

在连接通道的时候，我们只需在 QM1 端启动通道 Start channel(QM1.QM2)。

1.1.2 Server-receiver

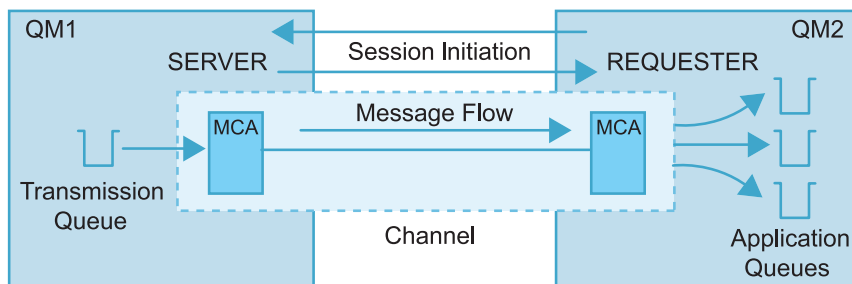
Server/Receiver 与 sender/Receiver 类似。Server 端是消息的发送方，也是连接的发起方。在配置的时候，必须指定对方的通信参数，由 CONNAME 设定。

1.1.3 Server-requester

Server/Requester 通道也是一种较常见的通道配置方式，如图 2 所示。从消息流向来看，Server 作为消息的发送方，requester 作为消息的接收方。但是从连接方式来看，requester 却是连接的主动方，server 是被动方。这种模式常用于动态 IP 地址的环境中，Server 是静态 IP 地址的服务器，Requester 的机器上网后自动分配到一个 IP 地址，所以是动态的，由 Requester 发起连接后接收数据。在本例

中，由 QM2（Requester）启动通道 Start channel(QM1.QM2)，如图 2 所示。

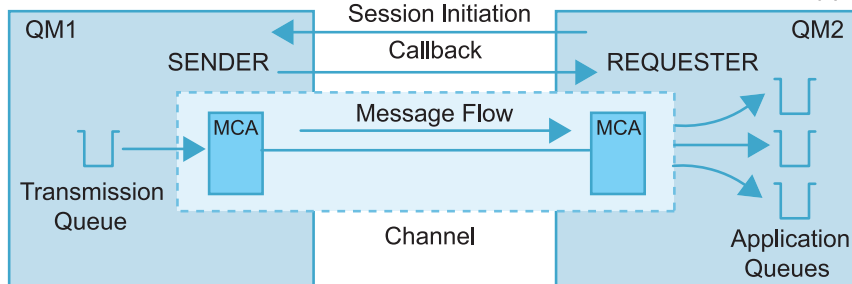
图 2



2.1.4 Sender-requester

Sender/Requester 的连接过程稍微复杂一些，如图 3 所示。Requester 首先与 Sender 连接，在通知对方连接参数后连接断开。Sender 进行反向连接，消息也是反向传送的，即由 Sender 传给 requester。这种反向连接的方式，称为回调连接（callback connection）。这种模式也常用于做双向验证，即双方必须都要知道对方的通信参数才能完成回调连接。

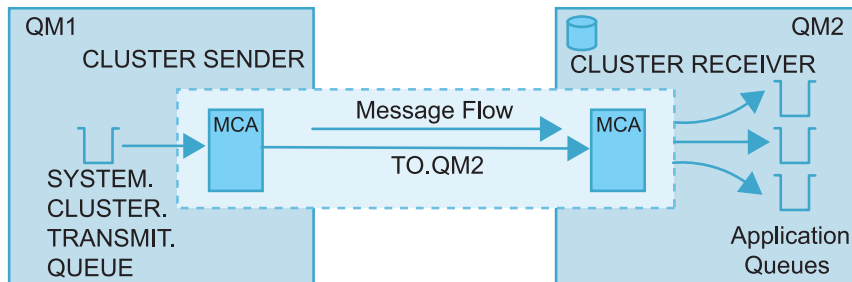
图 3



1.1.5 Cluster sender-cluster receiver

在一个 cluster 环境中，每一个队列管理器都有一个 Cluster sender 通道，通过这个通道把消息发送到完全资源队列管理器，每一个队列管理器也有一个 cluster-receiver 通道可以用来接收 cluster 中的消息。

图 4



在本文中我们不重点介绍 cluster sender-cluster receiver 通道。

1.2 消息通道的触发机制

1.2.1 触发器原理

触发 (Triggering), 是一种自动启动应用程序的机制。

队列管理器把某种条件称为触发事件。如果队列被设置为触发类型, 并且触发事件发生了, 那么队列管理器将发送一个触发消息到一个称作启动队列的队列中。触发消息被放置到启动队列的过程意味着产生了触发事件。

处理队列管理器中的消息是触发监控程序 (Trigger-Monitor Application), 他的工作是读取触发消息并根据触发消息的信息做出相应的处理。触发监控程序没有什么特殊, 它只不过是启动队列读取消息的应用程序。

当队列管理器发现由一条消息到达被触发的队列之后, 它产生的触发消息将被存放到启动队列中, 触发监控程序将从启动队列中取出触发消息, 并根据触发消息中的内容, 启动相应的消息处理程序来处理被触发队列中的消息。

触发所涉及的对象如下所示:

1) 应用队列 (Application Queue): 一个本地队列并设置为可触发。当触发条件满足时, 将会产生触发消息。

2) 传输队列: 如果用触发方式来启动通道, 则传输队列将对应了触发的应用队列。在传输队列的 TriggerData 属性中设置为将被启动的通道名, 这将省略进程的定义。

3) 进程定义 (Process Object): 一个应用队列可能由一个进程定义对象和它关联。进程定义中包含应用程序的信息。该应用程序负责从应用队列中取出消息。

4) 触发事件: 它是一种引起队列管理器产生触发消息的事件。

5) 触发消息: 当触发事件发生时, 队列管理器将产生触发消息。触发信息来自于应用队列和于应用队列关联的进程定义, 它包含了将要被启动的程序名。

6) 启动队列: 一个本地队列。被用来存放触发消息的队列。一个队列管理器可用拥有多个启动队列。一个启动队列可以为多个应用队列服务。

7) 触发监控器: 是一个持续运行的程序, 当一个触发消息到达启动队列时, 触发监控器获取触发消息, 并利用触发消息中的信息, 启动应用程序来处理应用队列中的消息, 并把触发消息头发送传递给应用程序, 消息头中包含应用队列名。

在所有平台上, 都有一个特殊的触发监控器叫做通道启动器 (Channel Initiator), 它的作用就是启动通道。

1.2.2 触发类型

● EVERY:

应用队列中每接受到一个消息时, 都将产生触发消息。如果应用程序仅仅处理一个消息就结束, 可采用这种触发类型。

● FIRST:

应用队列中消息从 0 变为 1 时会触发事件。如果当队列中的一个消息到达时启动程序, 直到处理完所有消息才结束, 则采用这种触发类型。

● DEPTH:

应用队列中消息数目和 TriggerDepth (引起触发事件发生时, 队列中的消息数目) 属性值相同时, 才会产生触发事件。当一系列请求的回复都收到时, 才启动应用程序, 则可以采用这种方法。

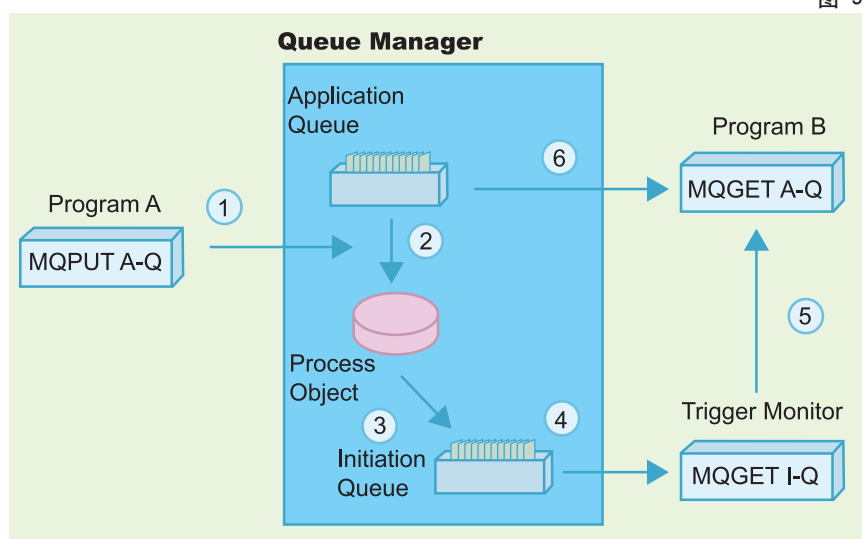
需要注意的时, 当 DEPTH 属性值为 0 的时候, 实际上就形成了同步通信。另外, 当采用 Depth 触发时, 产生触发消息以后, 队列将被修改为非触发方式, 如果需要再次触发, 需要重新设置成允许触发。

一般而言, 在实际应用中, 如果通道设置成触发方式, 触发类型往往设置成为 FIRST 和 DEPTH。

1.2.3 触发器工作流程

消息触发的工作原理如图 5 所示:

图 5



1) 本地或远程应用程序 A, 往应用队列(Application Queue) 中 PUT 了一条消息。

2) 如果队列的触发类型设置为first, 当队列原来深度为0时 (队列为空), 这时 PUT 一条消息到队列中将形成触发事件, 同时产生一条触发消息, 触发消息中将包含进程定义中的信息, 因为进程定义中包含启动程序B所需的信息, 所以触发消息中也包含了启动程序 B 所需的信息。

3) 队列管理器创建触发消息, 并把它 PUT 入与应用队列相关的启动队列 Initiation Queue。

4) 触发监控器(Trigger Monitor)从启动队列(Initiation Queue) 中 GET 触发消息。

5) 触发监控器处理触发消息, 发出启动应用程序 B 的命令。

6) 应用程序 B 打开应用队列(Application Queue), 并处理队列中的消息。

注: 如果是通道触发将可以不需要创建进程对象 (process object), 只是在传输队列的 trigdata 中设置需要启动的通道名。触发监控器采用系统自带的 Channel Initiator, 如图 6 所示。

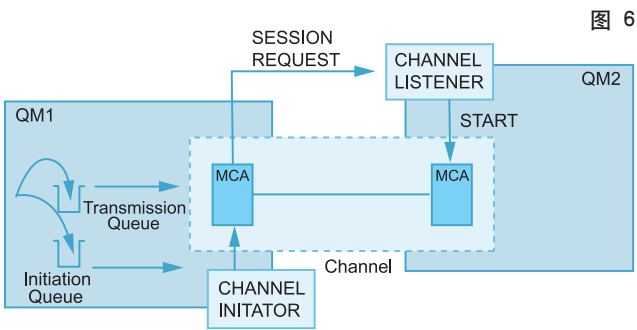


图 6

1.2.4 配置消息通道触发

配置消息通道触发启动, 需要使用到的对象有传输队列, 通道启动队列, 发送通道, 通道启动器。我们本配置案例中传输队列名是 QMB, 通道启动队列采用 SYSTEM.CHANNEL.INITQ, 发送通道名 QMA.QMB, 通道启动器为 runmqchi, 该进程在队列管理器启动的时候自动启动。下面我们通过举例来演示配置实现消息通道触发启动。

●首先我们来查看一下传输队列 QMB 都有哪些属性, 显示如下清单所示, 其中清单中的标注红色的属性和通道触发配置相关。

dis ql(QMB)

1 : dis ql(QMB)

AMQ8409: 显示队列细节。

QUEUE(QMB)	TYPE(QLOCAL)
ACCTQ(QMGR)	ALTDAT(2009-02-06)
ALTTIME(11.41.44)	BOQNAME()
BOTHRESH(0)	CLUSNL()
CLUSTER()	CLWLPRTY(0)
CLWLPRTY(0)	CLWLUSEQ(QMGR)
CRDATE(2008-12-05)	CRTIME(10.37.53)
CURDEPTH(0)	DEFBIND(OPEN)
DEFPRTY(0)	DEFPSIST(NO)
DEFSOPT(SHARED)	DEFTYPE(PREDEFINED)
DESCR()	DISTL(YES)
GET(ENABLED)	HARDENBO
INITQ()	IPPROCS(1)
MAXDEPTH(5000)	MAXMSGL(4194304)
MONQ(QMGR)	MSGDLVSQ(PRIORITY)
TRIGGER	NPMCLASS(NORMAL)
OPPROCS(1)	PROCESS()
PUT(ENABLED)	QDEPTHHI(80)
QDEPTHLO(20)	QDPHIEV(DISABLED)
QDPLOEV(DISABLED)	QDPMAXEV(ENABLED)
QSVCI(0)	QSVCIINT(999999999)
RETINTVL(999999999)	SCOPE(QMGR)
SHARE	STATQ(QMGR)
TRIGDATA()	TRIGDPHT(1)
TRIGMPRI(0)	TRIGTYPE(FIRST)
USAGE(XMITQ)	

●设置传输队列为触发模式 TRIGGER。

●设置触发类型为 every/first/depth 其中的一种, 例如 TRIGTYPE(FIRST)。如果设置为 TRIGTYPE(DEPTH), 则还需要设置触发深度属性, 例如 TRIGDPHT(4), 表示当队列中的消息数由 3 个增加到 4 个的时候才能形成触发事件, 但需要注意的是深度触发事件只能产生一次事情, 下次传输队列的消息数由 3 个增加到 4 个时候不会产生触发事件, 所以在消息通道触发中, 我

们推荐采用 TRIGTYPE(FIRST)。

- 在通道触发中,推荐使用SYSTEM.CHANNEL.INITQ队列作为初始队列,该队列为MQ专用的通道启动队列,不需要手工启动其触发监视器,设置传输队列 INITQ(SYSTEM.CHANNEL.INITQ)属性。

- 通过 TRIGDATA 属性设置需要触发的通道名,例如 TRIGDATA(QMA.QMB)。

- 对于 TRIGMPRI属性的含义是基于消息优先级触发,也即某类优先级的消息满足的触发条件才产生触发事件,我们在消息通道触发中不推荐使用。

完整 MQSC 命令参考如下:

```
ALTER QL(QMB) +
TRIGGER +
TRIGTYPE(FIRST) +
INITQ(SYSTEM.CHANNEL.INITQ) +
TRIGDATA(QMA.QMB)
```

- 通道的触发监控器我们采用系统的自带的runmqchi程序,缺省队列管理器启动的时候,runmqchi进程会自动启动。

1.3 通道主要属性

消息通道的属性非常多,下面对一些重要属性进行介绍。

1.3.1 Channel type 属性

通过的类型(CHLTYPE)有很多种,需要根据通道配对方式,设置您所需要的通道类型。请参看 2.1 小通道类型。

1.3.2 Connection name 属性

作为消息通道,只有 Sender, Server 和 Requester 通道才使用 Connection Name(CONNAME)属性。如果通讯协议是 TCP/IP,该属性设置可以指定通信对方的 IP 地址和端口号。

1.3.3 Transmission queue name 属性

Transmission queue name(XMITQ) 属性是设置传输队列名,就是通道需要从这个队列取出消息然后发送到对方。sender 和 server 通道需要设置 XMITQ 属性。

1.3.4 Disconnect interval 属性

Disconnect Interval(DISCINT)是发送和服务类型的通道所具有的一个参数,它的作用是:在它设置的时间间隔内,如果传输队列为空即通道上没有消息通过时,通道就会被停止。设置完 Disconnect Interval 参数之后,当发送方重起通道时,通道就会被正常启动。

Disconnect Interval 的值会地影响通道的性能。如果把 Disconnect Interval 的值设置得非常小,会导致通道的频繁启动;反之,如果把 Disconnect Interval 的值设置得很大,则意味着即使通道上很长时间没有消息,系统资源也会被长期占用,从而影响系统的性能。因此,利用改变 Disconnect Interval 的值,我们可以有效地改善通道的性能。

当传输队列中没有消息要传送时,发送方通道(SDR)、服务器通道(SVR)将在等待了该参数指定的时间间隔后断开连接,停止通道。该参数以秒为单位,定义新的通道时,如果没有特别指定,该参数会继承系统对象的属性,设为6000秒,约两个小时。亦通道连续两个小时没有消息发送后就会停止。DISCINT参数设定为0,通道永远不会停止。

1.3.5 Heart Beat Interval 属性

与 Disconnect Interval(HBINT)相对应的是 Heart Beat Interval 这一参数(仅针对 WebSphere MQ for AIX、HP-UX、OS/2、Sun Solaris、Windows NT/2000 V5.1 以上)。它的作用是:在 Heart Beat Interval 指定的时间间隔内,如果传输队列上没有一直没有消息到达,发送方 MCA 会向接收方 MCA 发送一个心跳信号,据此给接收方通道以停止的机会,在这种情况下,它不必等待 Disconnect Interval 超时,也会将通道停止下来。同时,它会释放用来存储大消息的内存空间并关闭接收方的队列。

为了使 HeartBeat Interval 和 Disconnect Interval 这两个参数更有效地发挥作用,一般情况下需要让 Heart Beat Interval 设置值小于 Disconnect Interval 设置值。

另外,如果我们使用的传输协议是 TCP/IP,我们也可以利用设置 TCP/IP 的 socket 的 SO_KEEPALIVE 参数来实现这一功能。设置完 SO_KEEPALIVE 参数,并设置时间间隔之后,TCP/IP 本身就会定期去检测另一端连接的状态,如果对方连接已断开,通道也会被停止。在这里,TCP/IP 的时间间隔也应小于 WebSphere MQ 通道的 Disconnect Interval 的值。

1.3.6 ShortRetry 和 LongRetry 属性

在发送类型等类型的通道属性中,还有四个参数是与通讯恢复和通道连接有关的,它们是: shorttrty,shorttmr,longtrty,longtmr,它们的缺省值分别是: 10, 60, 999999999, 1200, 分别代表短重试时间间隔和次数以及长重试时间间隔和次数,它们的作用和含义在于当通道从running变为 retrying 状态时,按照这四个参数规定的时间间隔和次数进行通道重新连接的尝试,并且先进行短重试,短重试结束后,再进入长重试。

在设计这四个参数时,要注意以下两点:

1) 要确保短重试+长重试的时间\故障恢复时间

例如: 假设您估计您的系统故障恢复时间为1个小时,则要设置 shorttmr*(time of shorttrty)+longtmr*(time of shorttrty)> 2 hours 这样,才能保证在故障恢复之后,通道仍然能够自动进行重新连接的尝试。

2) 重试间隔将影响自动恢复的效率

例如: 如果您把短重试总时间设定为10分钟,而长重试时间间隔设为1小时,而网络在15分钟后,便已经恢复,可是此时,由于通道已经进入长重试阶段,它将在1个小时之后,才能通过长重试恢复通道的正常运行。相反,也不必把重试间隔设置得太短,应根据需要和用户的实际情况进行适当的设置。

1.3.7 Batch size 属性

通道的Batch size(BATCHSZ)值是影响通道性能的一个关键参数。在MQ进行消息传输时,通道对消息的处理也是在同步点的控制之下并具有交易特性的,在以下条件满足时,它将统一提交一批消息:

当发送的消息个数达到 BATCHSZ时;或传输队列为空,并且在 BATCHINT 指定的时间间隔内一直没有消息到达时。

缺省情况下,通道的 Batchsz 是 50,这是一个较为合理和优化的设置。一个小的 Batch size 值会使每条消息占用大的资源。比如,假设我们在局域网的情况下, Batch size 值越大,通道的性能越好。然而,在广域网环境下,要根据网络状况的好坏

来设置该参数,若网络状况很差, Batch size 值越大,可能会导致通道的性能越差。

1.3.8 NpmSpeed 属性

WebSphere MQ的消息分为永久性消息和非永久性消息两种,对于非永久性消息,通道属性NPMSPPEED可以设置为FAST和NORMAL,为了提高性能,可以设定该属性为FAST。

1.4 通道通讯优化

1.4.1 选择侦听程序(Listener)的运行方式

MQ的侦听程序有两种配置和启动方式,一种是通过配置/etc/inetd.conf文件选择使用inetd方式,也可以使用MQ自身提供的runmqslr程序,所不同的是:runmqslr是一个线程应用,所以比inetd需要更少的内存消耗。因此,采用runmqslr方式可以提高通道相关的性能。

1.4.2 选择通道的运行方式

与侦听程序类似,与MQ通道对应的MCA代理程序也可以有线程和进程两种运行方式,可以通过定义通道的MCATYPE属性来设置通道MCA的运行方式为thread方式。让通道以线程方式运行而非进程方式运行,这样可以减少通道运行的进程个数和消耗的内存资源。

对于以上两点,要注意的是:当通道的连接个数很多时,如在Unix平台上超过500个时,建议MCATYPE和listener采用进程的方式。

1.4.3 设置侦听程序的 trusted 运行方式

与MQ应用程序类似,MQ的通道侦听程序也有trusted(fastpath)和non trusted(standard)两种运行方式,采用trusted运行方式可以降低CPU和内存消耗。将通道和侦听程序设置为trusted方式运行的方法是通过修改qm.ini配置文件中的MQIBindType参数来实现,即创建或修改qm.ini文件中与Channels相关的小节,例如:

Channels:

MQIBindType=FASTPATH 或者

MQIBindType=STANDARD

其中,FASTPATH表示trusted运行方式,而STANDARD



表示非 trusted 运行方式。

1.4.4 通道的 PipeLineLength 属性

从MQ版本V5.2开始，MQ提供了一个新的通道参数，即PipeLineLength属性，通过该属性，可以设置MCA采用多个线程的方式来传输消息，从而成为提高通道性能的又一个手段。缺省情况下，该参数数值为1，任何大于1的设置MQ本身都会将其设置为2。

设置PipeLineLength的方法是修改qm.ini文件的Channels一节，如下：

```
Channels:
```

```
PipeLineLength=2
```

注意，必须在通道两端都进行设置，否则会取两者之间的最小值。

1.4.5 设置 MaxChannels 和 MaxActiveChannels 属性

MaxChannels和MaxActiveChannels分别代表队列管理器允许配置的通道的最大个数和允许同时运行的通道的个数，MaxChannels的缺省值是100，MaxActiveChannels的缺省值与MaxChannels相同。如果您的并发通道连接个数超过了100，您需要修改这两个参数。这对于大并发的Client/Server间通讯尤为重要。

1.4.6 使用 AdoptNewMCA

在WebSphere MQ的5.2版本以上，新增了一个控制通道运行的新的参数，既AdoptNewMCA,它可以通过修改qm.ini文件的Channels一节进行修改,如:

```
Channels:
```

```
AdoptNewMCA=ALL
```

它可以被设置为: NO,SVR,SNDR,RCVR,CLUSRCVR,ALL,FASTPATH等值。当MQ接收到启动通道的请求，但是同时它发现与该通道对应的amqcrsta的进程已经存在，这时，该进程必须首先被停止，然后，通道才能启动。AdoptNewMCA的作用就是停止这种进程，并且为新的通道启动请求启动一个新的进程。

该属性可以将状态处于running状态的接收端通道强行终止，从而使发送端的通道启动或请求操作得以成功。

如果为某一通道指定了AdoptNewMCA的属性，但是新的通道由于"channel is already running"而启动失败，则它可以：

1) 新的通道通知之前的通道停止

2) 如果旧的通道在AdoptNewMCATimeout的时间间隔内没有接受该停止请求，相应的进程（或线程）被kill掉

3) 如果旧的通道经过步骤2仍未终止，则当第二个AdoptNewMCATimeout时间间隔到达时，MQ终止该通道，同时产生"channel in use"的错误。

1.4.7 实现断网续传和故障恢复

WebSphere MQ作为一个消息传输产品，本身是架构在TCP/IP之上的，因此与操作系统或网络底层的TCP/IP特性有着密切的关系，很多情况下我们要借助于修改操作系统的TCP/IP参数，来使它更好地为WebSphere MQ服务，从而更加完善地发挥WebSphere MQ的强大功能。

当我们要在WebSphere MQ的两个队列管理器之间建立通讯时，由于WebSphere MQ的通道是单向的，我们必须要建立两条通道(Channel)，比如建立两条类型分别为发送(sender)类型和接收(receiver)类型的通道来实现两个WebSphere MQ Server之间的通讯，在通道的两端WebSphere MQ利用MCA(消息通道代理)来管理和监控通道的启停等运行状态，并对通道两端的消息序列号(Message Sequence Number)等进行协同管理，以保证WebSphere MQ对消息的"Once and Once Only"的传输。对WebSphere MQ而言，在其系统配置配置文件mqsc.ini文件中，在mqsc.ini文件中，包含了对队列管理器的日志大小、通道属性以及通过XA标准与数据库协同工作时的有关参数的设置，除此之外，有一节用于控制有关TCP/IP特性的信息，即：

```
TCP:
```

```
KeepAlive=Yes 或
```

```
KeepAlive=No
```

它的作用在于：当设置KeepAlive=Yes时，表示操作系统的TCP/IP参数设置对WebSphere MQ生效。

由于WebSphere MQ接收通道的MCA处于通讯的被动方，它一直等待从发送方传来的消息，因此它不知道什么时候发送方会停止发送消息，也不知道当网络出现故障时，发送方什么时候会从工作状态变为停止状态。这时由于出现网络故障，网络连接被断开，发送方通道状态会由running状态变为retrying状态，

发送方会试图重新建立网络连接,而这时接收方的通道却没有停下来,仍处于一种假"running"的状态,相应的我们会得到一个"Channel is in use"的错误信息,导致发送端想重起却重起不了。出现这一现象的原因是:当发送方MCA启动通道并长时间没有断开连接,这时出现网络故障,TCP/IP的socket连接被破坏,当发送停止通道并重新启动时,它需要建立一个新的socket连接,而接收方仍停留在原来的RECEIVE调用上,它的socket特征与发送方新的socket特征不一致,因此新的socket连接建立失败。

我们可以利用TCP/IP特性来克服这一点,更好地实现断网续传。通常,操作系统的TCP/IP参数的缺省设置是2个小时(常见的操作系统平台如:Windows 2000/NT以及AIX,HP-UX,Sun Solaris, Linux等,缺省设定均为2个小时)即发送KeepAlive探测包的时间是2小时,所以需要2个小时的时间它才会获知网络连接已经断开,这对于我们来说无疑是无法接受的。在这种情况下,我们可以通过配置TCP/IP KeepAlive参数来提高TCP/IP的响应速度,从而实现网络故障时WebSphere MQ能够迅速断开通道连接从而重新启动通道,实现断网续传的强大功能。只有这样,在发送端,MQ的channel才能由running状态变为retrying状态,同时,在接收端,MQ的channel才能由running状态变为not found状态,这样,在网络或主机重新恢复时,通道才能重新建立起连接,恢复running状态。

要实现上述功能,我们需要作以下两方面的工作:

1) 修改WebSphere MQ系统配置文件mqsc.ini,增加如下节:

```
TCP:
KeepAlive=Yes
```

目的是使系统的TCP/IP设置对WebSphere MQ生效。

2) 修改操作系统的TCP/IP参数;

在不同的系统上,修改TCP/IP参数的方法略有不同,现仅以Windows 2000/NT、RISC6000和HP为例作一简单说明。

- 在Windows NT平台上,我们利用regedit来修改系统注册表,修改HKEY_LOCAL_MACHINE\CurrentControlSet\Services\Tcpi\Parameters下的以下三个参数:

KeepAliveInterval,设置其值为1000

KeepAliveTime,设置其值为300000(单位为毫秒,300000代表5分钟)

TcpMaxDataRetransmissions,设置其值为5

- 在RISC6000平台上,用no命令修改如下参数:

tcp_keepidle保持TCP/IP连接的时间,单位为0.5秒,缺省值为14,400,即两个小时,我们可将它设为5分钟;

tcp_keepinttcp连接初始timeout值,单位为0.5秒,缺省值为150,我们可将它设为50;

tcp_keepintvl连接间隔,单位为0.5秒,缺省值为150,我们可将它设为50;

我们也可以修改/etc/rc.net文件,

```
/usr/sbin/no -o tcp_keepidle=240
```

```
/usr/sbin/no -o tcp_keepinit=50
```

```
/usr/sbin/no -o tcp_keepintvl=50
```

注意:直接使用命令行修改,在机器重启后,会失效;修改rc.net文件,可以做到永久生效。

- 在HP平台上,

对于HP-UNIX V10.20及其在此之前的版本,用/usr/contrib/bin nettune命令来修改有关参数;

对于HP-UNIX V10.30及其以上版本,用/usr/bin/ndd命令来修改有关参数。

- 在SUN Solaris平台上,

用ndd -set /dev/tcp/tcp_keepalive_interval NNN命令来修改有关参数,tcp_keepalive_interval的单位为毫秒,缺省值为7200000毫秒,即2个小时。

- 在SCO OpenServer平台上,

tcp_keepalive和tcp_keepidle相同,其原先默认值为7200秒,可设为600秒。tcp_keepintvl其原先默认值为75秒,可设为15秒。均以"秒"为单位。

运行命令ifconfig命令修改:

```
/etc/inconfig tcp_keepidle <value>
```

```
/etc/inconfig tcp_keepintvl <value>
```

需要注意的一点是通道两端的KeepAlive参数要保持协调一致,若发送端的KeepAlive值小于接收端的KeepAlive值,则当网络出现故障时,发送端的通道停下来之后,接收端的通道会仍然停不下来。

1.5 通道的维护

通过使用MQSC命令DEFINE来创建通道,DELETE删除

通道。用 DISPLAY 显示通道属性，用 ALTER 修改属性。注意，通道只有在停止状态下才可以被删除或修改。

例如：

```
* 创建接收端通道(QMA.QMB)
DEFINE CHANNEL(QMA.QMB) CHLTYPE(RCVR)

* 创建发送方通道(QMA.QMB)，连接对方的IP地址为192.
168.1.100，端口为1415，通道连接传输队列TO.QMB
DEFINE CHANNEL(QMA.QMB) CHLTYPE(SDR)
CONNAME('192.168.1.100(1415)') XMTIQ(TO.QMB)

* 删除通道(QMA.QMB)
DELETE CHANNEL(QMA.QMB)

* 修改通道(QMA.QMB)的批次消息数量为100
ALTER CHANNEL(QMA.QMB) CHLTYPE(SDR) BATCHSZ
(100)

* 显示通道(QMA.QMB)的断开间隔时间
DISPLAY CHANNEL(QMA.QMB) DISCNT

* 显示通道(QMA.QMB)的状态
DISPLAY CHS(QMA.QMB) AL
```

由于通道是一种特殊的MQ对象，它的某些状态会随着通信环境的改变而变化，比如通道状态、通道流量、通道消息序号等，我们称之为通道的动态信息。MQSC 也提供了一些命令用来动态管理通道。

1.5.1 通道 start 命令

通道的启动可以通过 MQSC 的 start channel 命令完成，例如格式为：

```
START CHANNEL(ChannelName)
```

也可以在命令行通过 runmqchl 控制命令完成，两者效果相同，在 Windows 中还可以用 MQ 服务配置成自动启动方式。

runmqchl 在通道连接的主动方使用，使用时需要指定队列管理器名和通道名。

```
$runmqchl [-m QMgrName] c ChannelName
```

选项 -m QMgrName 表示队列管理器名，缺省为缺省队列管理器。 c ChannelName 示通道。

```
例如：runmqchl m QM1 c QM1.QM2
```

在 Websphere MQ for Windows 产品中，可以通过

WebSphere MQ 服务（本地）工具来进行配置。

1.5.2 通道 stop 命令

用 MQSC 命令 STOP CHANNEL 可以停止通道，停止通道也只有在连接通道的主动方发起才有作用。例如格式为：

```
STOP CHANNEL(ChannelName)
```

1.5.3 通道 reset 命令

通道为传送的每一条消息分配了一个序列号，它会自动累计增值，每传送一条消息，双方的消息序号都会自动加一。这个消息序号在通道中用 SEQNUM 属性表示，在双方连接通道的时候会约定一个起始值，以后每传递一条消息各自加一。通道的相关属性 SEQWRAP 表示序号的最大值，缺省最大值为 999,999,999。序列号越界后自动归零，从头开始。通道利用消息序号来标识传送和确认的消息。

通常情况下，通道双方的消息序号计数应该是相同的。然而在某些异常情况下，会出现双方序号不一致的情况，这通常是因为通信故障后，双方对前面的某一条（或一批）消息是否发送成功理解不一致。在解决了不确定（In-doubt）的消息后，可以用 MQSC 命令通过重置消息序号将双方调整到一致。一旦连接断开后，通道重连时双方 MCA 会将消息序号同步。如果通信异常造成序号不一致，可以在通道发送端用 MQSC 命令 RESET CHANNEL SEQNUM 手工将两者同步。

在连接通道的主动方重置消息序号会将双方一起调整，在被动方重置则只设置一端。

```
RESET CHANNEL(ChannelName) [SEQNUM(number)]
```

1.5.4 通道 resolve 命令

发送方和接收方的通道状态中除了 SEQNUM（通道消息序号）参数控制消息传递外，还有 LUWID 参数。LUWID 指的消息批次交易号，对于每一批消息发送方都需要收到接收方的确认信息才认为消息完整无误地送达对方，接着产生下一个 LUWID 并开始下一批消息传送。如果没有收到确认而与接收方失去联系，这时发送方认为这批消息为不确定（In-doubt）状态。

大多数时候，WebSphere MQ 会在通道重连时自动解决不确定状态的问题。当然，我们也可以手工解决。事实上，通道的 LUWID 分成 CURLUWID 和 LSTLUWID 两个参数属性，具体工

作过程如下：

- 发送方产生一个批次交易号，设置在 CURLUID 并通知接收方
- 接收方将其设置在 CURLUID
- 发送方向接收方一条接一条地传送整批消息
- 接收方在完整地收到消息后，将交易号设置在 LSTLUWID，提交整批消息并回送确认信息
- 发送方在接收确认信息后，将交易号设置在 LSTLUWID，提交整批消息
- 重复上述步骤。

所以，在发送方出现不确定状态时，只需要比较一个发送方的 CURLUID 和接收方的 LSTLUWID，就可以知道该批消息在接收端是否已经提交，从而在发送方做出相应的动作即可。具体步骤如下：

1) 比较双方的 LUWID

- 对于不确定 (In-doubt) 状态的发送端：

DISPLAY CHSTATUS(ChannelName) SAVED CURLUID

- 对于接收端：

DISPLAY CHSTATUS(ChannelName) SAVED LSTLUWID

2) 如果两者相同，说明该批消息在接收端已经完整地收到并提交。在发送端执行：

RESOLVE CHANNEL(ChannelName) ACTION(COMMIT)

3) 如果两者不同，说明该批消息在接收端未能完整地收到并提交。在发送端执行：

RESOLVE CHANNEL(ChannelName) ACTION(BACKOUT)

1.5.5 通道 ping 命令

类似于 TCP/IP 中的 ping 命令，MQSC 命令中也有对通道的 PING，格式如下。其中，DATALEN 表示 PING 数据包的大小，可以用 16 字节到 32,768 字节。

PING CHANNEL(ChannelName) [DATALEN(16|integer)]

PING 命令可以检查对方的队列管理器或端口监听器是否启动，也可以检查对方的通道定义是否正确。但不检查通道的通性状态。换句话说，PING CHANNEL 只检查通道能否连通，而不检查目前是否连通。

1.6 消息通道状态

图 7 显示了所有可能的通道状态层次关系。如果通道不是 inactive 状态，则是 current 状态。如果通道不是 retrying, stopped 或 starting 状态，则是 active 状态。如果通道是 active 状态，则通道可能存在 initializing, binding, requesting, running, paused, stopping 状态。当通道处在 active 状态时，系统需要消耗资源，并且会有一个进程或者线程启动。如果通道是 stopped 状态：

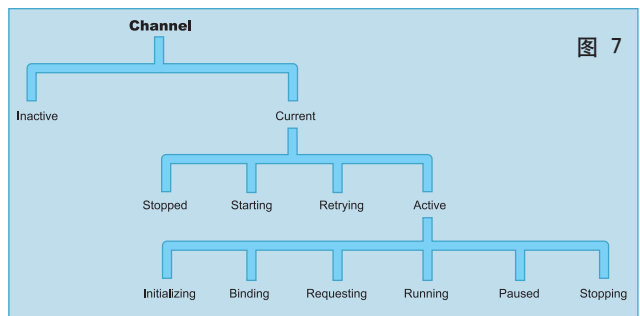
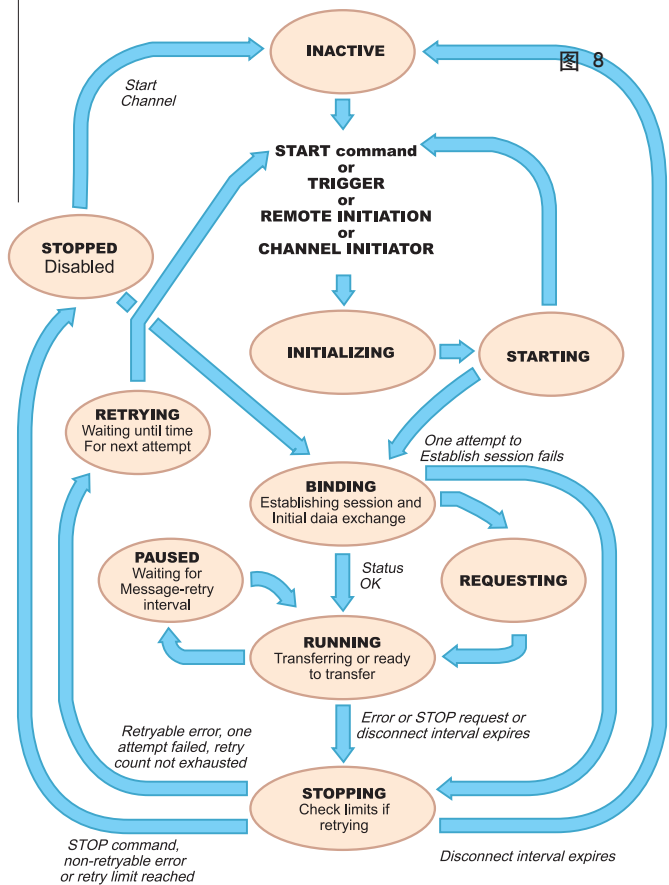


图 8 显示了各种通道状态之间变化的流程。



2. 附录：MQ 对象配置参考

队列管理器 QMA 和队列管理器 QMB 是互为双向通讯，它们都分别有本地接收队列，传输队列，死信队列，远程队列，发送通道，接收通道。

队列管理器 QMA 的对象定义清单：

```
* 定义接收队列 QLA
def ql(QLA) maxdepth(10000) defpsist(yes) maxmsgl
(1048576) replace
* 定义传输队列 TO.QMB
def ql(TO.QMB) usage(xmitq) defpsist(yes) maxdepth
(10000) maxmsgl(1048576) +
trigger trigtype(first) trigdata(QMA.QMB) initq(system.
channel.initq) replace

* 定义发送通道 QMA.QMB
def chl(QMA.QMB) CHLTYPE(SDR) discint(1800) +
XMITQ(TO.QMB) CONNAME('11.193.9.74(1415)') trptype
(tcp) +
shorttrty(10000) shorttmr(30) longtmr(300) longrty
(999999999) +
batchsz(50) replace

* 定义接收通道 QMA.QMB
def chl(QMB.QMA) CHLTYPE(RCVR) replace

* 定义远程队列
def qr(QRB) rname(QLB) rqmname(QMB) xmitq(TO.QMB)
defpsist(yes) replace

* 定义死信队列
def ql(DEADQ) defpsist(yes) maxdepth(20000) replace
alter qmgr deadq(DEADQ)
*
alter ql(system.cics.initiation.queue) defpsist(yes)

* 修改 first 触发间隔时间
```

```
alter qmgr trigint(5000)
```

队列管理器 QMB 的对象定义清单：

```
* 定义接收队列 QLB
def ql(QLB) maxdepth(10000) defpsist(yes) maxmsgl
(1048576) replace
* 定义传输队列 TO.QMA
def ql(TO.QMA) usage(xmitq) defpsist(yes) maxdepth
(10000) maxmsgl(1048576) +
trigger trigtype(first) trigdata(QMB.QMA) initq(system.
channel.initq) replace

* 定义发送通道 QMB.QMA
def chl(QMB.QMA) CHLTYPE(SDR) discint(1800) +
XMITQ(TO.QMA) CONNAME('11.193.9.73(1415)') trptype
(tcp) +
shorttrty(10000) shorttmr(30) longtmr(300) longrty
(999999999) +
batchsz(50) replace

* 定义接收通道 QMA.QMB
def chl(QMA.QMB) CHLTYPE(RCVR) replace

* 定义远程队列
def qr(QRA) rname(QLA) rqmname(QMA) xmitq(TO.QMA)
defpsist(yes) replace

* 定义死信队列
def ql(DEADQ) defpsist(yes) maxdepth(20000) replace
alter qmgr deadq(DEADQ)
*
alter ql(system.cics.initiation.queue) defpsist(yes)

* 修改 first 触发间隔时间
alter qmgr trigint(5000) ■
```

带您走进 WebSphere sMash 的奇妙世界

IBM 中国开发中心 Web 2.0 首席架构师 毛新生 / 文



随着互联网的发展,新的技术和趋势正在改变着开发人员从事 Web 应用开发的方式。面向 Web 的轻量级开发技术在社区中逐渐流行,比如 PHP、Python 以及 Ruby and Rails 等动态脚本语言都拥有大量的用户;新的 Web 2.0 技术和富客户端在不断改进用户体验;Web 和企业中有越来越多的可重用资产以 REST 服务、Feed 和 Widget 等方式提供出来;企业中需要大量的情景应用来满足快速变化的商业需求。

基于这些趋势、变化,在去年的 IBM IMPACT 2008 会议上,IBM 发布了一个创新的产品 WebSphere sMash。它源于 IBM 的一个孵化项目 Project Zero。该项目的目标就是希望使得 Web 应用开发变得更加简单、快速和敏捷。

WebSphere sMash 为新一代 Web 应用提供一个开发和执行平台,并且从根本上简化开发、部署和运作体验。通过引入一系列广受欢迎的实践模式,开发者和使用者不仅可以快速开发、部署和运行 Web 应用程序,同时结合 sMash 本身的诸多特性还可以快捷并简便地与已有 Web 应用程序和企业级系统进行集成。

首先 sMash 提供了一个简化很多,快捷很多的轻量级平台和编程模型,来服务于 Web 应用程序的开发和运行。这个轻量级平台的模块化架构建立在一个非常微小(小于 10M)的核心组件之上,其他功能都以打包的方式提供,只有在需要的时候才加载。而且使用 WebSphere sMash 开发的每个程序都独立地运行在一个 Java 虚拟机进程之中,不需要依赖额外的类似“容器”概念。这意味着程序员可以看到完整的应用堆栈,并控制程序的一切,当面对各种问题的确定,比如性能的优化、内存溢出的原因的界定等,要容易得多。更值得一提的是,这种完整性还是建立在一系列快捷方便的“轻量级”特性之上:启动一个 sMash 应用程序只需要 1 秒左右,运行时的 Java 虚拟机进程的内存占用仅为

30M,空闲时更降至 300K 多一点,良好的容错特性还允许你重新启动 sMash 应用程序而没有任何状态损失,跨平台特性使得在不同的平台环境中,复制打包好的功能组件即可使用。

WebSphere sMash 强调对动态脚本语言(Groovy, PHP 等等)的直接支持。Java 程序员可以使用 Groovy 这一运行在 Java 虚拟机上的动态语言来开发,而习惯开放源代码开发和 LAMP(Linux/Apache/MySQL/PHP、Perl 或 Python)的程序员可以使用 PHP 语言来开发,这让来自不同社区的众多程序员可以很快地开始在 sMash 平台上,利用自己已有的技能开发应用或服务。目前许多使用 PHP 开发的流行 Web 应用程序都可以不加任何修改地在 sMash 平台上运行,与此同时,人们在 sMash 平台上开发出来的服务和应用,自然而然,是 REST 风格的。

WebSphere sMash 专注于根据面向服务架构(SOA)开发敏捷的 Web 2.0 应用程序,它针对 REST-style 服务发布,集成, Mash-ups 和丰富的 Web 界面开发提供了优化,大大简化了动态 Web 应用的开发复杂性。sMash 提供了一种简单编程模型来构建 REST 风格的 Web 2.0 应用。它不仅允许程序员通过一些简单的约定和动态语言来快速创建 RESTful 的资源,并且对于基于数据库的应用,提供了 ZRM(Zero Resource Model)的技术来简化对于 RESTful 的资源的建模和访问。使用 ZRM 的支持,你可以快速创建数据库表并且获得该资源的 JSON(JavaScript? Object Notation)或者 ATOM 的资源表示方式。并且在此之上提供了基于 Dojo(一种流行的 Ajax 开发框架)的 ZRM DataStore 实现,使得 Web 开发工程师通过 JavaScript 代码就可以轻松访问后端的 REST 资源,实现增删改查等基本操作。WebSphere sMash 还提供了一系列对于 Web 应用安全的支持,不仅包括了一些标准的认证方法,还支持第三方的认证方式,例如 OpenID 协议。

除帮助用户快速创建 Web 应用程序之外, WebSphere

sMash 还提供以活动为单位的流程设计功能，来将不同的服务或应用程序集成起来。它以流程方式集成和组装 REST 服务和 RSS 或 Atom 格式的数据，并且提供了一个基于浏览器的图形化流程设计工具，结合已经提供的处理企业中常见的审批、邮件通知等业务流程的基本活动单元，帮助开发人员迅速设计和组装应用。

WebSphere sMash 的另一个有益的尝试就是提供了一个基于 Web 浏览器的集成开发环境，用于快速开发构建 WebSphere sMash 应用程序，她提供的内置编辑器具有语法高亮显示、Java/Groovy/PHP 调试功能、并可进行可视化页面编辑和流程设计等功能，也可启动和停止应用并访问运行时日志，而这一切都只需要通过浏览器来完成。

在 2009 年 2 月，IBM 宣布将在 Amazon EC2 弹性计算云服务上提供一系列软件产品，并根据使用量付费。IBM 提供免费的 Amazon Machine Images (AMIs) 供 WebSphere sMash 开发及测试使用，让软件开发工程师可快速通过 WebSphere sMash 在 Amazon EC2 平台上创建和部署应用程序，通过 AppBuilder，

开发人员可以使用浏览器进行远程的开发和调试。

现在已经有很多的合作伙在 WebSphere sMash 上面积极开发自己的产品和解决方案。比如 SugarCRM, ZSL 等等。您能够通过 Youtube 搜索 WebSphere sMash 相关视频来观看的客户成功案例和演示。

在另外一方面 WebSphere sMash 在软件开发过程本身也进行了有益的探索。WebSphere sMash 采用社区驱动的商业开发模式进行发展。www.projectzero.org 作为它的开放在线社区，提供所有功能组件的下载，并欢迎用户免费注册，并提供反馈意见，建议，批评和交流。目前 WebSphere sMash 的下载量已经超过 90 万份，通过这种交互式的开发方式，用户社区将能参与和影响有关 sMash 的技术决策，能够直接接触到开发团队和源代码。这种技术和沟通透明性可以更好地为用户提供优化的解决方案，提供更加符合客观实际的技术产品。2009 年，sMash 正在围绕开发社区的反馈意见和建议，积极地开发新功能并改进已有特性。■

(感谢我的团队对撰写本文提供支持)

WebSphere sMash 介绍

WebSphere sMash 是 IBM WebSphere 家族提供的一个新应用平台产品，它提供了一个 Web 2.0 应用的平台，覆盖了开发、部署和运行管理的全部过程。WebSphere 能够很好地为企业关键业务提供安全、可靠、可伸缩，和管理的平台，基于业界多年的企业计算经验。

但是，企业中也有很多非关键业务应用，这样的应用数量很大，需求非常多样，也不见得用很长的时间，最好能够快速开发，部署和运行，sMash 提供了一个简化很多，快捷很多的轻量级平台和模型，来服务于这类应用的开发和运行。

在我的心目中，WebSphere sMash 提供了一个很好的下一代 Web 应用的编程模型和平台。利用 sMash，动态脚本语言是基本手段，Java 程序员可以用

Groovy (<http://groovy.codehaus.org/>) 来开发，而习惯开源和 LAMP 程序员可以使用 PHP 语言来开发，这让众不同社区的多程序员可以很快地开始在 sMash 平台上，利用自己已有的技能开发服务。与此同时，人们在 sMash 平台上开发出来的服务和应用，自然而然，是 REST 风格的。

sMash 在 Java 平台上，引入和应用了一些好的实践模式。这里讨论一下，首先是 Application is the Server，应用本身就是服务器，这避免了使用“容器”的概念。容器是一个大管家，有很多好处，可是这些好处建立在容器所规定的各种契约，限制和模式上的，比如 J2EE 规范中的各种 SPI 和规范。通常这些规范代表了企业计算平台的最佳实践，但是需要人们花比较大的力气和长的时间来学习。sMash 则

不同，一个应用独占一个 JVM 进程，程序员看到完整的应用堆栈，需要什么能力，调用 sMash 提供的 library，程序员控制程序的一切。这对各种问题的确定，比如性能的优化，内存溢出的原因的界定等，要容易得多，另外一个 convention over configuration，这在各种社区中讨论和使用很多了，很高兴 sMash 将其作为一个设计原则，不过因为各种原因，sMash 的 configuration 似乎多了一些。

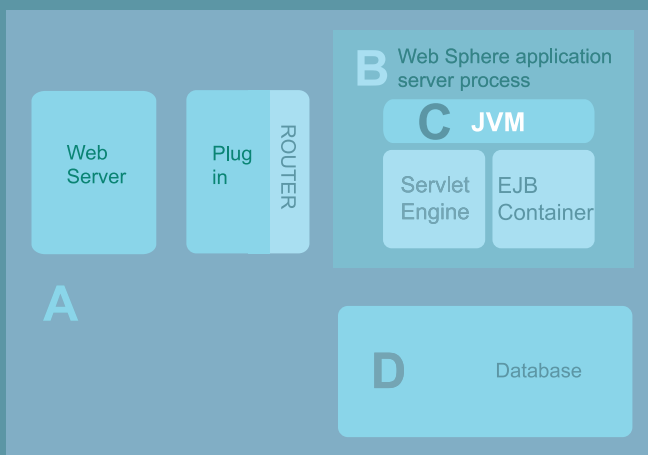
WebSphere sMash 是一个商业化产品，但是通过开放社区的方式来开发，这个社区就是 Project Zero (<http://www.projectzero.org>)，是 IBM 软件开发的一个尝试，希望在商业软件开发的过程中，随时让人们接触和了解开发团队，了解开发计划和过程，还可以通过社区反映自己的看法。所以这个产品从应用模型，到开发方法，都充满了 Web 2.0 的元素。■



WebSphere Cluster 调优过程

1. WebSphere Application Server 影响性能的因素

下面是应用服务器 WebSphere Application Server 的整体架构图：



如上图所示，影响 WebSphere Application Server 应用服务器的性能主要包括如下几个方面：

1. 系统硬件配置
2. 操作系统配置
3. WEB SERVER
4. WebSphere Application Server 进程
5. Java 虚拟机 (JVM)
6. 数据库

本文主要对 WEB SERVER、WebSphere Application Server 进程、Java 虚拟机 (JVM) 进行调优。另外 WAS 调优分应用调优和参数调优两部分，应用调优主要是在开发应用程序时进行的应用性能优化，这种调优

对性能的优化最明显。本文主要介绍的是在应用调优完成后进行的 WAS 参数调优。

2. WebSphere Application Server 参数调优

在着手进行应用程序服务器的优化之前，首先进行应用程序

和操作系统的优化是一个更好的选择。尤其是应用程序的优化，通过对应用程序中影响性能的部分进行重新的设计和调整，往往能够带来比单纯的参数调优更为巨大的性能提升。对于一般的 J2EE 应用程序而言，WAS 中最重要的优

化参数包括针对 JVM、Web Container 和 Data Source 等的。

● JVM

Heapsize (-Xms 和 -Xmx)：heapsize 的大小依赖于系统平台和具体的应用等多种因素。最大 heapsize 需要小于机器的物理内存，一般来说，设置最大 heapsize 为 512m 是一个常见的起点。同时，在生产环境中，最好将 Xms 设置为小于 Xmx 的值。

GC (Garbage Collection)：一般来说，良好的 GC 状态需要保证相邻两次垃圾回收的平均间隔时间应当是单次垃圾回收所需时间的至少 5-6 倍。GC 的调优是通过在模拟压力的情况下不断调整最大最小 heapsize

来实现的。

Heap Fragmentation：heap 碎片的问题在 JVM 中存在大对象的情况下尤为突出。减少碎片的方法包括调整 pCluster (-Xp) 和 kCluster (-Xk) 参数。

● Web Container

对 Web Container 的调优是通过 Web Container 传输链中各个通道 (TCP、HTTP、WebContainer) 的参数调整进行的。这些参数包括诸如 ThreadPool 的最大最小值，buffer 大小，timeout 时间的大小，keep-alive 的值等等

● Data Source

对 Data Source 的优化包括两个方面。一是 JDBC Driver 的选取，尽可能使用 Type 4 的 JDBC driver，这种 driver 是纯 java 的，适用于 client/server 模式，并提供比 type2 和 legacy/CLI 的 driver 更好的性能。另一方面是 Database 连接池的参数设置，主要包括最大和最小连接以及 timeout 的设置。具体的设置于应用程序的特性和并发用户量相关，一般来说，可设置最小连接为 1 且最大连接为 30，作为一个继续调优的起点。

除了 JVM，Web Container 和 Data Source 之外，WAS 的性能调优还包括很多其他方面的内容，如 JMS、EJB、Session、Dynamic Cache 等等。

3. WAS 性能调优的基本步骤

部署在 WAS 上的 J2EE 应用程序，其性能是由多个因素决定的。例如网络、数据

库、内存分配、WAS 服务器的配置以及应用程序的设计。对于一个标准的 J2EE 应用，一个请求到来时，往往需要经过多次转发：网络 > Web 服务器 Web 容器 > EJB 容器 > 数据库。而每一次转发，都可能造成请求处理的瓶颈，使得应用程序整体性能下降。如果我们把每一次转发的待处理资源都看成一个队列，如下图：

对于 WAS 调优，要记住的一个基本原则就是，使得在队列中等待的请求的数量最小化。在实践中我们发现，为了达到这个目的，最有效的配置方式就是使得队列成为一个“漏斗”。也就是说，越靠近客户端的队列，其容量越大，而后面的队列，其容量要略小于或等于前面的队列。按照这个原则，调优的基本步骤如下：

- 设置的是 Web Server 的最大并发用户：

- ▲ 这个设置是在 conf/httpd.conf 这个文件里面配置的。在 Unix 系统中，对应的属性是 MaxClient；在 Windows 系统中，对应的属性是 ThreadsPerChild。



- 设置 Web Container 的最大、最小并发用户：

- ▲ 在管理控制台中点击应用程序服务器 > server1 > 线程池 > WebContainer，根据观察的性能情况和应用情况输入合适的最小、最大进程数。

- 对象请求代理 (ORB) 的线程池大小：

- ▲ 在管理控制台中点击应用程序服务器 > server1 > ORB 服务 > 线程池，根据观察的性能情况和应用情况输入合适的最小、最大进程数。

- 设置数据库的连接池属性：

- ▲ JDBC 提供者 > 数据库 JDBC 驱动名称 > 数据源 > 数据源名称 > 连接池，根据观察的性能情况和应用情况输入合适的最小、最大连接数。

- JVM 堆参数设置的性能调优：

- ▲ 应用程序服务器 > server1 > 进程定义 > Java 虚拟机，根据硬件物理内存和应用情况输入合适的初始堆大小、最大堆大小。

- ORB 参数调用方式的性能调优：

- ▲ 应用程序服务器 > server1 > ORB 服务 > 选中按引用传递。

- 关闭动态加载开关：

- ▲ 企业应用程序 > 应用名称 > 关闭启动类重新装入开关。

- ▲ 关闭会话序列化，应用程序服务器 > server1 > 会话管理 > 分布式环境设置 > 分布式会话选择无即可。

4. 合理使用缓存机制

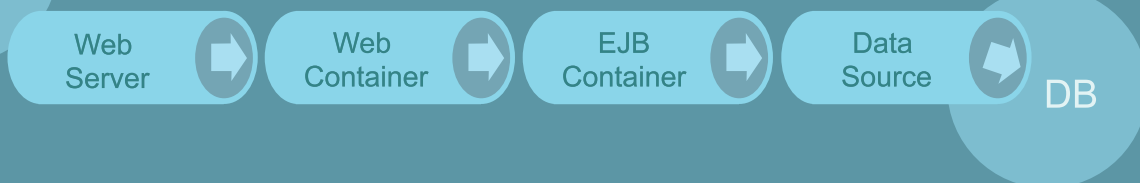
WAS 中可以用很多种手段实现缓存。其中最常见的一种，就是在应用中使用开发的手段缓存一些中间结果。但这种缓存是一把双刃剑。用好了可以很好地提高性能，如果掌握不好，使用过度了，则会对底层 JVM 的 Heap 造成很大的威胁。一般 JVM Heap 出现 OutOfMemory (内存泄漏) 的问题，都是应用中直接或者间接的缓存技术滥用的后果。所以缓存技术拿捏得当非常重要，也比较不容易。

下面主要谈谈 WAS 产品本身提供的缓存功能。

WAS 提供动态缓存机制，可以对 Web 命令、Servlet 输出和 JavaServer Pages (JSP) 文件进行高速缓存，从而提高应用程序的性能。动态高速缓存服务在应用服务器 JVM 中工作，拦截对可高速缓存对象的调用。例如，它通过 servlet 的 service

WebSphere Queuing Network

待处理资源队列





方法或命令执行方法拦截调用，以及将对象的输出存储到高速缓存，或者对来自于动态高速缓存的对象内容进行处理。

在 WAS 中可以通过配置实现常用的高速缓存对象、功能和模块有：

- Servlet 高速缓存
- Portlet 片段高速缓存
- Edge Side Include 高速缓存
- Web 命令高速缓存
- Web Service 高速缓存
- Web Service 客户机高速缓存

对于 WAS 中的缓存机制，和应用开发中的缓存类似，也要掌握好缓存的“度”，避免滥用，原则有如下几点：

1) 不是缓存用得越多越好。只要是 WAS 中提供了缓存机制的环节，统统都用上。这是错误的。

对于这一原则，有一个非常简单的方法可以确定是否采用缓存，采用哪些缓存技术：如果实现缓存本身相对于您的拓扑结构或者您所掌握的技术来说，显得非常的复杂，那么完全可以不用。举例来说，虽然 Edge Component 中提供了很好的缓存技术，但您的生产环境中，原本就没有考虑 Edge Component，现在为了缓存而缓存，非要把 Edge Component 加进去，使得拓扑偏离了原先的设计，变得更复杂化，就很没有必要了。

2) 确认您想缓存的对象是否真的需要缓存。这一点事先比较难以判断。因为生产环境真实的使用情况是千变万化的。最佳的方法就是在真实环境中对缓存进行监控，查看被缓存对象的命中率。如果某些缓存环节利用率极低，或者某个对象命中缓存

的概率非常小，则完全可以取消这样的缓存。对于利用率高的缓存，可以在内存使用比较平稳的前提下适当增大力度。

5. WAS 性能差的表现和解决方法

系统性能差一般的情况下有以下表现

- 1) CPU使用不高，用户感觉交易响应时间很长。
- 2) CPU使用很高，用户感觉交易响应时间很长。

对于第一种情况，可以断定是由于系统的某一小部分造成了瓶颈，导致了所有的请求都在等待。简单举例来说，线程池的数量开的太小，导致所有的请求都在排队等待进入线程池，因为没有可用的线程使用，所以这个交易请求一直在排队，导致交易响应时间很长。如果数据库连接池开的太小，也会有同样的表现。

对于第二种情况，比较复杂。可能的根源之一是硬件资源不够。根源之二是应用系统中产生了多个大对象。根源之三是程序算法有问题。解决思路如下：用性能分析器，如RAD或JProfiler，对运行环境进行分析，分析哪个类甚至于哪个函数消耗了这么多的CPU，并找到相应的解决方案。

6. 应用程序性能监视

WAS 提供了若干工具，可以用于帮助衡量其性能。WAS 中免费提供的 Tivoli Performance Viewer (TPV) 允许客户对关键

资源（如 JVM、Web 容器和 EJB 容器以及远程连接池）进行监视。这款工具使用非常方便，可以用于确定应用程序使用可用资源的方式，还可以提供针对行为不正常的远程资源的信息。例如，如果数据库连接池显示为了获得连接进行了多次尝试，则可能表示远程数据过载，或者需要优化数据库，或者需要向不够大的池中添加更多的连接。与此类似，WAS还提供了 Runtime Performance Advisor 功能，可以就系统中的潜在优化问题向管理员提供反馈。

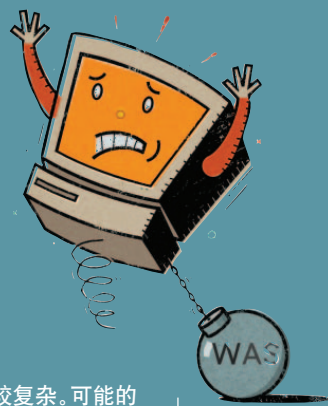
如果出现问题，使用Tivoli Performance Viewer，察看如下最重要的10个监视项目：

Servlets 和 Enterprise JavaBeans

- 1) 平均响应时间
- 2) 请求数（事务）
- 3) 活的 HTTP Session 数
- 线程池
- 4) web 服务器线程
- 5) web容器和EJB容器线程池
- 数据源
- 6) 数据源连接池大小

- Java 虚拟机
- 7) JVM 内存，垃圾收集统计
- Web, 应用程序, 和数据库服务器上的系统资源
- 8) CPU 利用率
- 9) 磁盘和网络 I/O
- 10) Paging 活动

同时，可配合使用 Tivoli Performance Advisor，它们可帮助您调整您的系统，对设置不足给出建议。（网友提供） ■



我和 WebSphere 的故事——在摸索中前进

华北计算技术研究所 陈华雄 / 文

一、初识 WAS

当年, 我们公司的战略发生改变, 成为了 IBM 的合作伙伴。因此, 公司的产品应用也做了一给改变, 计划将我们基于开源平台的 JBoss 产品迁移到 WAS 上运行。当初我们小组前台使用 JSF+Spring+Hibernate3/EJB3+JBpm3 架构来进行开发, 但是却不能直接迁移到 WAS 上运行, 因为 WAS 默认采用了 JSF 的 Apach 实现, 与 SUN 的实现有所差异, 故直接迁移不能运行。几经周折, 我们只好把原来的组件全部迁移到 Apache 的实现, 对 EJB3 的功能进行转换, 才顺利地通过 WAS 上的测试。

对 WAS 的看法:

1) 对其它厂商的组件限制, 对自身的平台组件捆绑;

2) 对 JDK 的最新版本的支持滞后, 当初我们开发的平台已经是应用 EJB3, JDK5, 但是 WAS6 还是停留在 EJB2.x、JDK1.4, 使得我们要修改已经完成的代码来符合平台。而为了迁移, 差不多把原来的项目几乎重写;

3) 那时我们使用的是 WAS 的最新版本 6.0, 补丁非常多, 也经常出现一些莫明其妙的问题。例如, 在开源平台运行没有问题, 但上到 WAS 平台却报错; 在大多数的情况下, 如果没有售后服务, 想解决问题, 困难极大;

4) 性能卓越, 平台稳定。对比开源平台, 处理大量并发, WAS 具有很大的性能优势。从我们测试的数据来看, 结果很令自己吃惊, 十分满意;

二、在 Portal 的折腾中成长

起步于 Portal 6.0, 当时迫于项目的要求, 走上了 Portal 的整合之路。从前台的主题皮肤、导航到后台的内容及数据整合; 从简单的使用到后台的管理维护; 从使用产品提供的模式到扩展产品架构自定义模式开发。随着项目的深入, 从深度得到理解, 从折腾中成长, 成熟, 得到重生。

学习 Portal 是个比较复杂的过程, 因为要学习的点实在太多, 所以前期学习起来会比较麻烦; 在学习的过程中, 除了要了解 Portal 产品本身以外, 还得理解与其它产品的结合; 对于跟 IBM 其他产品的结合, 相对来说还是比较容易, 但如何进行配置就不见得那么简单了。

回想起来, 自己除了加班就是加班, 除了看文档写代码还是看文档写代码。在那段日子没有人可以帮忙, 没有例子可以参照。一切都得自己研究: 看文档, 理解文档, 看产品代码、反编译代码。然而庆幸的是, 在此过程中还是可以解决了很多问题, 基本上能满足用户的需求。说实在的, 虽说那段时间做的项目比较多, 但不见得自己有什么高明之处, 唯独主题皮肤这块自己还有些许成就感。

对 Portal 的看法:

1) 拥抱标准

Portal 对规范标准支持得很好, 对 jsr 168, 280 都有很好的支持, 其容器是双容器统一的特性。

2) 平台灵活, 架构开放

Portal 的平台相当灵活, 你可以很轻易整合业务系统 (IBM 产品或非 IBM 产品) 及你自己定制的整合方式, 它没有其它平台的强约束性 (当然定制是有一定限度的)。

你可以进行简单修改就能实现新的需求, 或者根据它的平台机制重写实现来符合你的需求, 这些灵活都源于其架构开发。平台的架构开发, 它自身只是一种实现, 开发者可以根据自己的需求来开发自己的实现。例如, TAI 机制, 用户可以外挂自己的认证、应用它的 SPI, 用户还可以定制生成的 URL, 内容的访问、搜索等等。

3) 开发工具多样化

对于 Portal 的开发, IBM 提供 RAD 及 WPF 的开发——自定义的编码方式 RAD 和快速开发的工具 WPF。这是两款十分优秀的开发工具, RAD 的开发很强, 支持广泛的 J2EE 开发; WPF 支持快速生成代码, 其架构开放, 用户可以根据它的机制开发自己的组件来重用;

三、稳定的 MQ

MQ 的名声比较好, 得到了广泛的认同。即使在使用的过程中碰到一些问题, 但基本上解决起来比较容易, 个人认为 MQ 是非常稳定的。此外, 支持 MQ 的平台、开发语言也很丰富。相对其它产品, MQ 的学习资源比较丰富。在 EAI 领域, 经常可以看到它的存在, 经常能听到开发人员的称赞。开发人员经常说的一句话: “丢给 MQ 就行了, 都不用你管”。可想而知, 它的稳定性有多高。

对 MQ 的看法:

1) MQ 给我的印象就是稳定性出奇的好, 在项目中, 它是最值得信赖的消息传送员。在 EAI 项目中, 有很多系统的交互, 但极少出现异常;

2) 概念简单, 易学。MQ 中间件概念比较清晰, 代码级应用, 对于开发人员来说比



较容易上手,看看书,学习一些必要的代码,即可以上岗操作,实践性较强;

3)资源丰富:MQ的资源比较丰富,相关论坛也多;

4)得到广泛的认可:放眼望去,在政府,电信,金融,电力各个行业,都很容易看到MQ的身影,应用相当广泛。

四、功能强大的 MB

ESB和MB都是属于总线方面的产品,相对之下,MB开发起来比ESB容易,且兼容的开发语言及能通信协议要比ESB多。MB底层是以MQ作传输工具,上面作逻辑开

发,不需要像MQ那样编码,只需做消息转换,消息转换后再用MQ来传输。

由于MB是面向消息的开发,因此在项目中MB与MQ之间的结合非常好,一般都没有什么问题。但是,对Web Service的消息操作就不那么简单了,虽然逻辑上是非常简单,都是XML文档,但是用ESQL的MRM操作却并不简单。因此,每种报文都得慢慢调试,甚至用XMLNSC或XMLNS的文档格式来操作,需要自己对消息报文重新封装。当然,如果MRM格式能顺利操作则非常容易。

对MB的看法:

1)屏蔽代码级的消息转换,使信息转换

更加直接,逻辑性比较容易维护;相对于MQ来说,没有那多的更编码,简化了编码,同时逻辑清晰。作为总路线来说,MB维护各种消息流,容易进行对流的维护;

2)对消息机制广泛的支持:MB广泛支持消息协议,对非IBM产品支持力度也很大。由于MB依赖MQ,所以它的传输稳定性很好;

3)容易拓展,重用消息的处理逻辑:MB提供开放的消息处理结点机制,用户可以开发自定义结点,可以用C或Java来开发,你可以对项目的消息实现自定义结点,使很多消息的处理机制得到重用。 ■

韦氏紫血三国——DataPower 长板浴血

IBM 黄河 / 文

公元208年初,时建安十三年,曹操挟天子以令诸侯,领八十三万大军挥师南下,先取新野,后取荆A襄,刘备大败。却说玄德引十数万百姓、三千余军马,一程程挨着往江陵进发。赵云保护老小,张飞断后。时秋未冬初,凉风透骨;黄昏将近,哭声遍野。至四更时分,只听得西北喊声震地而来。玄德大惊,急上马引本部精兵二千余人迎敌。曹兵掩至,势不可当。玄德死战。正在危迫之际,幸得张飞引军至,杀开一条血路,救玄德望东而走。

却说赵云自四更时分,与曹军厮杀,往来冲突,杀至天明,寻不见玄德,又失了玄德老小,云自思曰:“主公将甘、糜二夫人与小主人阿斗,托付在我身上;今日军中失散,有何面目去见主人?不如去决一死战,好歹要寻主母与小主人下落!”遂拍马望长坂坡而去。正走之间,见一将手提铁枪,背着一口剑,引十数骑跃马而来。赵云更不打话,直取那将。交马只一合,把那将一枪刺倒,从骑皆走。原来那将背剑之将夏侯恩也。却见其将背后伏宝剑一口,拔出视之,靶上正面有金嵌“青”二字;翻过视之,又见隐隐金光流动,细看却是“DataPower”。云心下大喜,暗吟:“早知世间有宝物曰DataPower者,不想今日却被我得到!”遂插剑提枪,复杀入重围,回顾手下从骑,已没一人,只剩得孤身。云并无半点退心,只顾往来寻觅;终在一残墙下枯井之傍寻得幼主,乃解开勒甲绦,放下掩心镜,将阿斗抱护在怀,绰枪上马。但见曹军一齐拥至。云乃拔DataPower青乱砍,手起处,衣甲平过,血如涌泉。杀退众军将,直透重围。回见剑刃,寒光灼眼,滴血不粘,青锋一线,分毫不缺。

却说曹操在景山顶上,望见一将,所到之处,威不可当,急问左右是谁。曹洪飞马下山大叫曰:“军中战将可留姓名!”云应声曰:“右手鼠白烂银枪,左手DataPower剑,吾乃常山赵子龙也!”曹洪回报曹操。操曰:“真虎将也!吾当生致之。”遂令飞马传报各处:“如见到持DataPower宝剑者,不许放冷箭,只要捉活的。”因此赵云得脱此难;此亦DataPower宝剑护身也。

这一场杀:赵云怀抱后主,直透重围,砍倒大旗两面,夺槊三条;前后枪刺剑砍,杀死曹营名将五十余员。终杀出重围,纵马过桥,行二十余里,见玄德与众人憩于树下。云下马伏地而泣。玄德亦泣。云喘息而言曰:“赵云之罪,万死犹轻!幸得公子无恙!”双手递与玄德。又递上获之青宝剑;玄德视之曰:“此物乃未来神兵,DataPower者,SOA整合龙骨利器。子龙此番浴血奋战,当属DataPower三大功劳也。整合之选,简单易用,快速部署;子龙获此宝剑,既能应之于手,发乎其神,精乎于心,此其一功也;防护攻击,刀箭不入,安全无忧;子龙获此宝剑,出入八十三万曹军如闲庭信步,曹军望剑光而止冷箭,此其二功也;加速处理,超高吞吐,性能无敌;子龙获此宝剑,遂驰骋战场,虽彻夜鏖战不能止将军一息,虽陷马坑绊马索不能缓将军一步,七进七出当阳长坂,最后救我孺子归来,此其三功也。今汝得此宝物乃天数,汝当自配之!”云泣喜拜伏受之。后人诗曰:

血染征袍透甲红,整合谁敢与争锋!冲阵护主防冷箭,只有DataPower赵子龙。 ■

Tomcat 向 WebSphere 迁移的问题及解决方法

清茗 / 文

由于业务需要，公司决定将运行在 Tomcat 上的 Web 项目整体迁移到 WebSphere。在整个迁移过程中，我们遇到了很多问题，同样也学习到了不少知识。为了让更多准备将 Tomcat 迁移到 WebSphere 的用户少走弯路，更好地达到想要达到的目的，特将本次迁移过程中所出现的问题及解决方法整理出来，和大家一起分享。

版本：Tomcat 5.5

迁移至 WebSphere 6.0

开发工具：Eclipse 3.2

1、安装 WebSphere 时，一定要注意不要使用系统默认的 C:\Program Files 路径，而要将 Program Files 那段截掉，放在根目录（具体放在哪个盘的根目录是可以选择的，只要不装在 Program Files 下就行）。否则到后期发布时，会报如下错误：

找不到 java 类：

```
Files\IBM\WebSphere\AppServer\
java\jre\bin\ext\ibmorb\jar;D:\Program
```

原因：Program 与 Files 中间有空格，WebSphere 不能将其识别为一个目录，所以找不到相应的文件。如果已经报该错误，将 WebSphere 卸载后再进行重装。

2、WebSphere 安装完以后，需要将 Eclipse 里面的 JDK 指定为 WebSphere 自带的（路径为 C:\IBM\WebSphere\AppServer\java，即 AppServer 目录下的 java 文件夹），做匹配可知 JDK 版本为 1.4。

由于此前用的 JDK 版本为 6.0，所以版本降下来后，会导致代码全部变红，这时你需要对报错的地方都进行修改。当时我们做迁移的时候，由于考虑到解决这个报错不会涉及到太多的工作量，所以就对报错

代码做了修改。但是，最好的方式是让客户升级 JDK 版本。

3、代码修改完成后，工程处还有个错误：

```
Project facet Java 6.0 is not supported
此时你需要访问到工程所在目录，进入“.settings”文件夹下修改org.eclipse.jst.common.project.facet.core.prefs和org.eclipse.wst.common.project.facet.core.xml文件，将jst.java的版本都更改为1.4。
```

4、此时新增服务器 IBM 牌 WebSphere，填写端口号及相关内容：

Server Profile Directory：为对应概要文件（profiles）路径

Cell：为对应概要文件下，installedApps 目录下的文件夹名称（如：ljpNode02Cell）

Node：为上面 Cell 中填写的前半部分如（ljpNode02）

Server：若无特殊设置，一般采用默认 server1

需要注意的是：Cell 和 Node 填写时，直接是文件名，不能带有中括号，否则会找不到该路径。

5、选择新增的 Server，并进行发布，报如下错误：

```
E:\eclipse\plugins\org.eclipse.jst.server.websphere.core_1.0.101.v200609140551\serverdef\scripts\websphere.6.x.xml:35: Java returned: 103;
```

这个问题是 Web.xml 格式有问题。我的版本是 xsd 2.4，对应的解决办法是：节点顺序按照

```
(icon?, display-name?, description?, distributable?, context-param*, filter*, filter-
```

```
mapping*,
```

```
listener*, servlet*, servlet-mapping*, session-config?, mime-mapping*, welcome-file-list?, error-page*, taglib*, resource-env-ref*, resource-ref*, security-constraint*, login-config?, security-role*, env-entry*, ejb-ref*, ejb-local-ref*)来进行排列。
```

另外，taglib 在 xsd 2.4 中的书写方式是：

```
<jsp-config>
<taglib>
<taglib-uri></taglib-uri>
<taglib-location></taglib-location>
</taglib>
</jsp-config>
```

6、发布成功后，一切正常，但进入主页面后，却又报了个错：

```
Could not be defined due to: basic/DoradoDBManager (Unsupported major.minor version 50.0)
```

```
This is often caused by having a class defined at multiple
```

```
locations within the classloader hierarchy. Other potential causes
```

```
include compiling against an older or newer version of the class
```

```
that has an incompatible method signature.
```

原来是程序编译的 JDK 版本与实际使用的版本不符所造成的。

解决方法：依次选中 Project->Properties->Java Compiler，将 Compiler compliance level 更改为 1.4，然后点击‘确定’即可。■



WebSphere 性能调优全讲解

中软国际 曲建菲 / 文

一、WebSphere 优化概述

- WebSphere 参数调整, 使服务器运行更加稳定, 有效优化服务器;
- WebSphere 实时监控, 并通过日志等多途径分析程序 bug 和性能瓶颈;
- WebSphere 监测工具, 利用第三方监测工具, 减少工作量, 提高监控效率;
- WebSphere 配置开发, 开发特定配置文件, 减小服务器开销, 增强用户体验性;
- WebSphere 架构设计, 利用集群机制, 形成稳定高效的系统环境。

1、参数调整

WebSphere 的控制台, 在以往的项目中最常用的就是 war 包的部署以及程序的启停等, 对于 WebSphere 的配置参数没有过多的涉及, 使我们无法合理有效的利用应用服务器, 造成部分资源的浪费。通过业务分析及监控, 我们可以设置合理的 WAS 配置参数, 使其维护机制等更为合理, 使其提供的内部机制充分得到利用, 还可以通过其提供的 API 接口, 进行有针对性的程序优化。

2、监控优化

WebSphere 自身提供了一个功能相当强大的监控程序——TPV (Tivoli Performance Viewer), 通过开启监控的开关, 设置对各关注指标的计数器, 来进行有针对性的监控, 监控过程中可以控制数据的刷新速度, 日志保留情况, 来观察各计数器的数据变化。其他的监控软件还包括 LR, WASMon, jwebap 等等, 在监控的过程中, 由于实时监控造成很大的人力开销, 所以第三方软件的诞生, 使我们监控的工作事半功倍, 弥补了 TPV 的很多不足。第三方监控软件包括预警通知, 报表输出, 全局监控, 历史查询等重要功能。

3、文件分析

在监控的过程中, 以及日常运维中, 会产生很多的日志文件,

包括了垃圾回收日志、应用程序日志、服务器错误日志, 以及堆转储文件和线程池使用文件等, 这些文件的产生可以使我们快速的定位错误位置, 找出问题根源。IBM 提供了这些日志文件的分析工具和错误代码库, 方便我们分析和查询, 真正做到了监控——分析——优化一条龙服务。

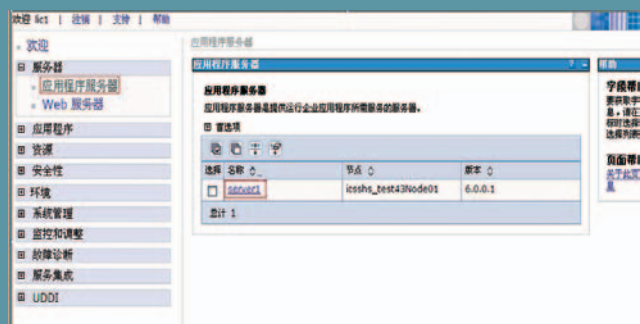
4、架构设计

WebSphere 在应对高负载时提供了一套机制——集群, 集群可以帮助我们进行负载均衡以及保证系统的稳定性, 在单机进行垂直部署或者多机的横向部署, 通过 DM 进行对节点的管理。集群的方式很多, 以应对不同的架构设计, 但这些都的前提是了解集群的架构和原理, 了解项目自身特点, 这样才能有针对性的进行方案设计, 以发挥集群的最大作用。

二、WebSphere 指标监控

1、前期准备工作

1) 用自己的用户名简称登录到 WebSphere 应用平台, 登录地址为 <http://xxx.xxx.xxx.xxx:9060/ibm/console/>, 进入页面后选择“服务器”->“应用程序服务器”, 点击页面服务名称 server1。



2) 进入 server1 页面后点击页面“性能”下的“性能监控基础结构 (PMI)”。



3) 进入“性能监控基础结构 (PMI)”页面后，勾选启用性能监控基础结构 (PMI)，在“当前监控的统计信息集”下选择“定制”方式，点击应用，应用完后，点击“定制”链接。



4) 进入“定制监控级别”页面后，依次启用“动态高速缓存、JDBC 连接池、JVM 运行下的线程、系统数据、线程池、Web 应用程序”下的一些计数器，具体启用哪些根据个人需求，举例如下：

> 动态高速缓存 (maxmemorycachesize, desc)

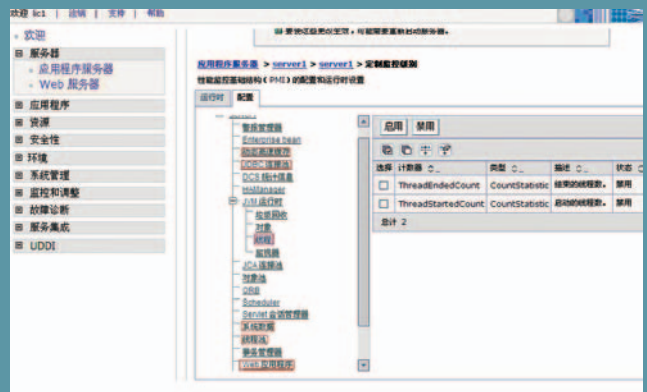
> JDBC 连接池 (AllocateCount, CloseCount, CreateCount, FaultCount, FreePoolSize, PercentUsed, PoolSize, PrepStmtCacheDiscardCount, WaitTime)

> JVM 运行时的线程 (ThreadEndedCount, ThreadStartedCount)

> 系统数据 (CPUUsageSinceLastMeasurement)

> 线程池 (ActiveCount, ClearedThreadHangCount, CreateCount, DestroyCount, PercentMaxed, PoolSize)

> Web 应用程序 (ConcurrentRequests, ErrorCount, LoadedServletCount, RequestCount, ServiceTime)



监控的主要参数包括线程池、连接池、Java 虚拟机等。常用计数器有：

JVM 运行时

计数器名称	描述	推荐
Heap Size	Java 虚拟机运行时的总内存	是
Used Memory	Java 虚拟机运行时的使用中的内存容量	是
Process CPU Usage	Java 虚拟机的 CPU 使用情况	是
Free Memory	Java 虚拟机运行时的空闲内存	
Up Time	Java 虚拟机已经运行的时间数	

JDBC 连接池

计数器名称	描述	推荐
Create Count	创建的连接数	是
Allocate Count	已分配的连接数	是
Fault Count	池中的连接超时数	是
Pool Size	连接池大小	是
Percent Used	正在使用的池的百分率 (基于设定的连接总数)	是
Close Count	关闭的连接数	
Return Count	返回到池的连接数	
Percent Maxd	所有连接的平均使用百分比	
Free Pool Size	池中的空闲连接数	
Waiting Thread Count	等待连接的平均并发进程数	
Use Time	使用连接的平均时间 (分配和返回连接的时间差)	



Wait Time	在允许连接之前的平均等待时间
Managed	正在用于连接池的 Managed Connection 对象数
Connection Count	
Connection	正在用于特定连接的 Connection 对象数
Handle Count	
Prep Stmt Cache	因为高速缓存已满而废弃的语句数
Discard Count	
JDBC Time	运行 JDBC 平均调用时间

线程池

计数器名称	描述	推荐
Create Count	— 创建的线程总数	是
Active Count	并发活动的线程数	是
Pool Size	池中的线程平均数	是
Percent Maxd	所有线程的平均使用百分比	是
Destroy Count	销毁的线程总数	
Active Time	线程处于活动状态的平均时间	
Declared	声明停止的线程数	
Thread Hung Count		
Cleared Thread Hung Count	清除停止的线程数	
Concurrent Hung Thread Count	并发停止的线程数	

5) 接下来, server1 页面下点击 Runtime Performance Advisor 配置。



6) 在打开页面, 勾选启用 Runtime Performance Advisor 设置时间间隔为 4 分钟, 处理器数设置为服务器对应的数量。



7) 设置完成后, 选择监控和调整下的性能查看器, 点击当前活动链接, 在页面点击 server1 链接。

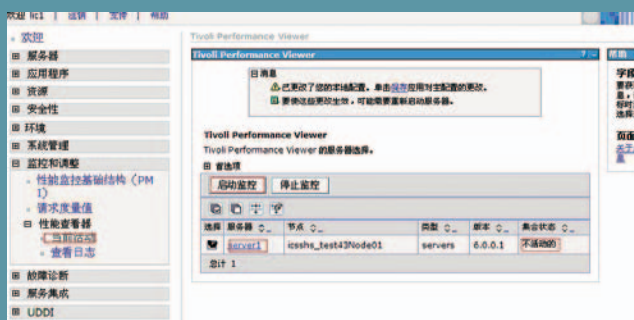


8) 进入页面后, 选择设置下的用户, 修改刷新率为 60 秒, 缓冲大小为 100, 点击应用。这个的作用是将各指标数据以列表的形式进行展示, 所设置的时间为列表的刷新时间, 定期对各指标进行数据反馈, 不能太快, 否则影响应用服务器的速度。缓冲区大小则显示表示为列表所能罗列的数据。

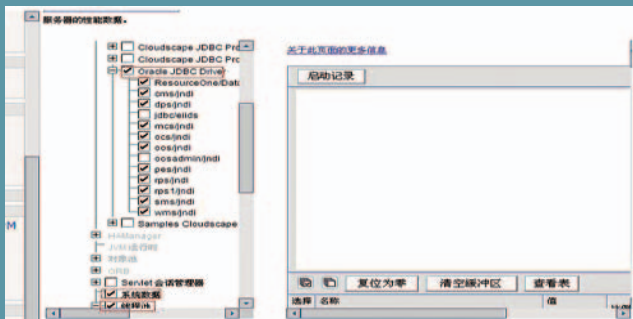


2、监测步骤

1) 在开始监测前，先选中服务，点击启动监测将集合状态改为活动的。(如果想做日志记录，这里需要点击启动监测，将server1设置为活动的)



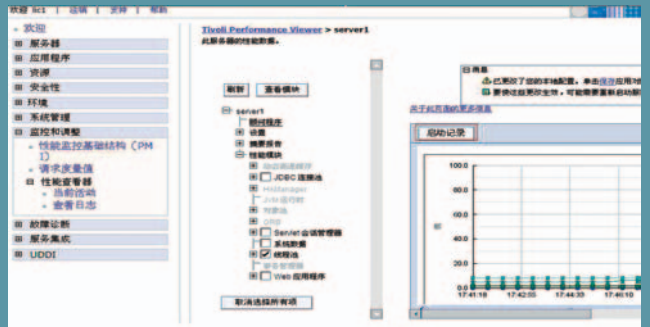
2) 进入页面后，在页面勾选性能模块下需要观测的指标，通常情况下，我们主要观测 JDBC 连接池、JVM 和线程池。



3) 设置完成后，在右边显示表中，先选择“全选”标记，再点击“清空缓冲区”按钮。勾选观测指标后，WAS 不会将所有设置的计数器全部显示，需要人为手动勾选，当选择全选后，计数器全部显示。清空缓冲区表示将观测前的垃圾数据清除。



4) 完成上面步骤后即可点击“启动记录”按钮，开始监测了。启动记录主要是将数据记录在日志文件中，需要设置日志配置。点击查看表，可以进行表格的方式进行查看（推荐），这样的方式刷新速度快，占用资源少。观测过程中，同时观测的计数器数量被 WAS 做了限制，所以我们需要手动来选择哪些需要观测，哪些暂时不考虑，随时切换观察各指标。



三、WebSphere 参数讲解

1、连接池

应用程序读取数据库有 2 种方式，一种是直接连接数据库，一种是调用连接池。

1) 直连是程序直接创建物理连接，调用数据库进行数据读取。直连的创建会带来很大的系统开销，若程序中多处频繁使用直连，会造成应用服务器资源消耗过多，影响程序的性能。

2) 连接池是创建和管理一个物理连接的缓冲池，其中会保留一定数量创建的物理连接不关闭，当有客户端请求时，调用连接池，可以有效减少物理连接的创建次数，降低直连所带来的系统开销，缓解应用服务器压力，提高程序性能。

下面我对连接池各参数意义进行讲解：

> 连接超时

概述：

连接超时是指，当对指定连接池进行请求时，池中并没有可用连接（连接全部被使用，或者数据库请求超时），当请求时间到达指定之间时未响应，那么这个时候就会产生超时异常，通过日志可以发现。

意义：

连接超时的设置,是对我们应用响应速度的一种把关,客户往往要求我们的产品在多长时间必须有响应,所以连接超时的设置,可以让我们发现哪些程序点有响应速度问题,可能是数据库查询语句问题,也有可能是程序逻辑死循环,再有可能就是数据库表结构需要优化,还有可能是最大连接数到达最大值。

> 最大连接数

概述:

最大连接数是指当前连接池中允许创建的最大物理连接数,当到达指定值后,将不允许创建物理连接。和连接超时相对应,当达到最大值后,连接请求将等待,直到池中有空闲连接为止,否则报连接超时错误。当使用集群机制时,会同时存在多个相同连接池,这个时候需要考虑最大数量的设置。

意义:

最大连接数可以有效控制创建物理连接的数量,连接池的大小影响着服务器资源的占用情况,若连接池过大,则会长期占用服务器可利用资源,若连接池过小,无法满足现场环境应用高负载使用压力。最大连接数的设置应根据 TPV 观测数据进行合理配置。

> 最小连接数

概述:

最小连接数是指当前连接池要保留的最小物理连接,其决定未使用超时维护机制的下限,连接池的创建不是根据最小连接数而特意创建,而是根据用户请求而创建,系统会一直维护最小的连接数目。

意义:

最小连接数使应用服务器保持一定数量的物理连接,利用应用服务器维护机制,合理分配服务器资源。当应用程序访问频繁,但访问人数少的情况下,最小连接数的合理配置,可以将有效的资源进行充分利用,满足特定应用需求。

> 收集时间

概述:

收集时间是连接池维护机制的核心,是指每次维护连接池的时间间隔。其有两个维护指标,分别为未使用超时和时效超时,其值应该小于两个指标中的任何一个。每一次维护周期中,连接池都会

将连接池中超时的物理连接关闭,以减少系统占用资源。

意义:

合理的收集时间设置,是帮助我们关闭不必要的连接,节省系统资源占用的有效途径。收集时间设置不易过大,因为时间间隔过长,会使很多未被使用的物理连接持续占用资源。若收集时间过小,则频繁的维护会带来很多系统开销,连接池的主要精力都放到了维护上。

> 未使用的超时

概述:

未使用的超时指池中的物理连接空闲未使用的时间间隔,每隔指定时间,系统会为连接标记,帮助收集时间在维护过程中进行关闭。未使用的超时应该小于实效超时时间,并且其以最小连接数为标准,当连接数超过最小连接数时,其才起作用。

意义:

未使用超时的设置,帮助我们关闭不必要的空闲连接,释放系统资源,并且减少数据库开销。根据现场环境使用情况,我们可以根据系统访问频繁程度,来定制合理的未使用超时,如果过小,当访问频繁程度大时,总需要重新创建,如果过大,当访问频繁程度不大时,连接池又空闲占用过多。

> 时效超时

概述:

实效超时指关闭物理连接的时间间隔,这个值是指到达指定的时间后,关闭满足时间条件的物理连接,若这个物理连接未使用,则直接关闭,若这个连接正在使用,则当前事务结束后,关闭此连接。这个值不受最小连接数的影响,若没有新创建的连接,此机制会关闭连接直到为 0。

意义:

时效超时的设置,是为了方式应用程序或者数据库造成的数据库连接持续占用,可能导致的原因包括程序逻辑错误,数据库宕机导致的错误等。还有一种情况为人为导致,就是若某个用户持续占用一个资源不放,会导致其他用户无法访问。所以时效超时的设置,是对不合理使用应用,或者链接错误等进行强行关闭,保证程序的稳定性和持久性。■



实现承诺 缔造价值

在中国已超过100间银行使用中联银行解决方案



中联银行解决方案客户

- 中国建设银行吉林省分行 • 中国建设银行宁波分行 • 渤海银行 • 徽商银行 • 威海市商业银行
- 武汉市商业银行 • 合肥市商业银行 • 乌鲁木齐市商业银行 • 西安市商业银行 • 昆明市商业银行
- 潍坊市商业银行 • 邯郸市商业银行 • 陕西省农村信用社 • 江苏省农村信用社 • 辽宁省农村信用社
- 福建省农村信用社 • 浙江省农村信用社 • 吉林省农村信用社 • 宁波国际银行 • 中国进出口银行



中联银行解决方案系列

整体解决方案
核心业务系统
卡业务系统
中间业务平台系统
多渠道接入系统

Web-Teller柜员系统
短信服务系统
网上银行系统
电话银行系统
客户经理系统

资金管理系统
信贷管理系统
业务分析系统
稽核监控系统
银行自动报表系统

办公自动化系统
公积金系统
现代化支付系统
个人征信系统
商业智能分析系统

中联集团简介

中联集团("中联"成立于一九八二年),为和记黄埔有限公司"和黄"之全资附属公司。中联集团是亚洲区首屈一指的信息科技服务供应商,专门从事系统基建及应用解决方案业务。中联的业务遍及中国大陆、香港及澳门。中联于中国大陆,为多家银行金融机构提供系统集成与软件应用服务,在该市场稳占领导地位。详情请浏览: www.vandagroup.com



和记黄埔集团成员

中联集团总部地址:
香港总部
香港九龙红磡德丰街22号
海滨广场二座11楼1107-1111室

Tel: (852) 21972888 Fax: (852) 21972333

中国区总部
中国北京西城区阜成门外大街2号万通
新世界广场A座9层

Tel: (010) 62270888 Fax: (010) 62270666

一、WebSphere 端口配置问题



新装的 WebSphere 和 IHS，配置了多个虚拟主机，分别设定不同的访问端口，但目前发现只有 80 和 9080 可以访问，其它的监听都没有启动，请问如何解决？

(1): 在 WebSphere 控制台里要添加虚拟主机端口，在 IHS 的 httpd.conf 里要加入相应的 Listen 节点，还要重新生成一下 plugin 配置文件。

(2): IHS 虚拟主机 SSL 配置，由于 WebSphere 的经典模式是将静态页面，图片与动态页面分离，所以经常会遇到虚拟主机配置的问题。

常规的虚拟主机配置与 apache 的虚拟主机配置一样只需配置 virtual host 标签，但是如果遇到两个虚拟主机都需要 SSL 通讯该如何配置呢？

下面给出配置方案：

其实没有大家想得那么复杂，只需要针对每个虚拟主机对应得域名配置一个 443 的虚拟主机即可。

- 1、用 ikeyman 生成密钥
 - 2、在 httpd.conf 中加入
- ```
NameVirtualHost 10.4.33.1:80
<VirtualHost 10.4.33.1:80>
DocumentRoot path1
ServerName domain1.cn
</VirtualHost>
<VirtualHost 10.4.33.51:80>
```

```
DocumentRoot ??path2
ServerName domain2.cn
</VirtualHost>
<IfModule mod_ibm_ssl.c>
Listen 443
NameVirtualHost 10.4.33.51:443
<VirtualHost 10.4.33.51:443>
DocumentRoot path1
??ServerName domain1.cn
SSLEnable
</VirtualHost>
<VirtualHost 10.4.33.51:443>
DocumentRoot path2
??ServerName domain2.cn
SSLEnable
</VirtualHost>
</IfModule>
SSLDisable
KeyFile "/opt/IBMIHS/bin/key.
kdb" ◆
```

## 二、WAS 安装完以后，是否还需配环境变量？



WAS 安装完以后，是否还需配环境变量？

为什么会出现本地服务开不起，系统找不到指定路径的情况？（之前把它装在 C 盘，指定的路径却是 E 盘）。以前在 E 盘上装过，但都卸载了的。请问本地服务的指定路径是否可以更改吗？

项目在 WAS 部署之后 访问地址不是 9080？

(1): 可能和以前装过但没卸载干净有关系，卸载的时候把所有 WAS HOME 里的所有文件全部删除；然后重启机器。

关于访问端口的问题，在安装的过程中如果没有更改，那么就是默认的 9080。

(2): 如果是 WAS6 的话，请到 Info Center 上搜索 vpd.properties 文件的相关介绍；如果是 WAS6.1，或之后版本的话，请到 Info Center 上搜索 .nifregistry 文件的相关介绍。

关于本地服务的问题：这是由于原来在 E 盘安装的程序没有清除干净所导致。解决方法：到注册表里去，按照 E 盘安装路径为关键字，进行搜索，然后修改相关内容。另：此法切勿在生产环境上使用。◆

## 三、WebSphere 节点问题



在 crt 上查询 WebSphere 的节点进程，用的是 `ps -ef|grep nodeagent`，可是只能查询出一个节点进程，查询不出所有的节点进程，用 `ps -ef|grep java` 查询进程，但查询出来的结果显示不全，也看不出哪个是要杀的进程，请问如何查询所有节点进程？

(1): 首先，要确定 WebSphere 的结构，如果没有配置 DM 的话，应该没有 nodeagent 对应的 java 进程。这样的话，就只有 Application Server 对应的 java 进程——`ps -ef | grep java | grep <servername>`。

如果配置了 DM，首先会出现一个 dmgr 对应的 java 进程，如果 nodeagent 服务启动，还有 nodeagent 对应的 java 进程，然后才是 server 对应的 java 进程。

(2): 节点，应该就是 profile。运行 `ps -ef | grep <profilename>` 就能查到当前节点所对应的 nodeagent 和 server 的进程信息。◆



## 四、如何监测 WebSphere 6.1 的性能?



Tivoli Performance Viewer (TPV), WAS 自带的, 在控制台上可以直接用。  
为什么使用 Tivoli Performance Viewer?

管理员和程序员可通过使用 Tivoli Performance Viewer (TPV) 来从管理控制台监视 WebSphere Application Server 的整体运行状况。

从 TPV 中, 您可以查看当前活动或记录以下内容的性能监控基础结构 (PMI) 性能数据:

- 1、系统资源 (如 CPU 使用率);
- 2、WebSphere 池和队列 (如数据库连接池);
- 3、客户应用程序数据 (如平均 servlet 响应时间);
- 4、还可查看通过使用 TPV 实现定制 PMI 的其他产品或客户应用程序的数据。

通过查看 PMI 数据, 管理员可以确定需要改变应用程序和配置设置的哪一部分以改进性能。例如, 为了确定要关注的应用程序部分, 可查看 servlet 摘要报告、企业 bean 和 Enterprise JavaBeans (EJB) 方法, 并确定哪些资源的响应时间最长。然后, 可以将重点放在改进响应时间最长的那些应用程序资源的配置上。

通过查看各种图或使用 Tivoli Performance Advisor, 使用 Tivoli Performance Viewer 来帮助管理配置设置。例如, 通过查看线程池的摘要图表, 您可以通过监视使用量百分比 (%) 来确定是否需要增加或减小线程池的大小。根据提供的数据更改配置 设置之后, 可确定更改的效果。要帮助进行配置设置, 请使用 Tivoli Performance Viewer。当应用程序正在运行时, 顾问程序查看各种数据, 并提供配置设置建议来提高性能。◆

## 六、数据源链接问题



**问题描述:** 两台 WAS 做集群, 版本是 6.1, 操作系统是 Windows。数据源已经配置好, 测试是成功的, 应用部署后, 登录页面也能访问, 但是输入用户名和密码, 登录没有反应。

systemerr.log 有报错信息如下:

```
[09-1-17 3:37:04:921 CST]
00000045 SystemErr? ???R??at com.
ibm.ws.util.ThreadPool$Worker.run
(ThreadPool.java:1473)
```

```
[09-1-19 9:29:13:656 CST]
00000045 SystemErr? ???R org.
```

```
springframework.jdbc.
CannotGetJdbcConnectionException:
Could not get JDBC Connection; nested
exception is java.sql.SQLException: 调用
中无效的参数DSRA0010E: SQL 状态
= null, 错误代码 = 17,433
```

```
java.sql.SQLException: 调用中无效的
参数DSRA0010E: SQL 状态 = null,
错误代码 = 17,433
```

```
at oracle.jdbc.dbaccess.DBError.
throwSQLException(DBError.java:134)
at oracle.jdbc.dbaccess.DBError.
throwSQLException(DBError.java:179)
```

Spring 框集中有一个 jdbc 配置文件, 应该查看一下这个配置文件是否跟 was 中

## 五、创建 Cluster 的错误



环境: Windows 平台  
产品: WAS 6.1

之前已经创建了一个名为 MyCluster 的 Cluster, 删除后, 再重新创建一个同名的 Cluster, 出现如下错误:

```
The new cluster could not be
created: ADMG9201E: Replication
domain already exists for cluster
MyCluster.
```

应该是未彻底删除旧的那个 MyCluster 所造成的原因, 想问接下来如何进行操作?

(1): 确认删除集群的操作是不是已经保存并应用修改了。

(2): 创建集群的时候, 一般都会生成一个和它同名的复制域(就是集群节点之间会话复制的配置)。在不记得是否已经删除集群的情况下, 也需要确认这个复制域会不会自动删除。◆

## 七、WebSphere 6.1 中的乱码问题



在控制台上, 选中 Server > Java 和进程管理 > 进程定义 > Java 虚拟机, 设置通用 JVM 参数:

```
-Dfile.encoding=GBK
-Dclient.encoding.override=GBK
-Ddefault.client.encoding=GBK ◆
```



# 读者反馈

www.WebSpherechina.net



我们希望了解你的需求，  
请简单的完成下面问题并传真给我们。

传真：010-68867107-805

印象最深的栏目：\_\_\_\_\_

希望了解更多信息的栏目：\_\_\_\_\_

最想从本刊中了解的内容：\_\_\_\_\_

对 WebSphere 这一技术的哪些侧重点感兴趣：\_\_\_\_\_

本期喜爱的文章：\_\_\_\_\_

本期不喜爱的文章：\_\_\_\_\_

请剪下并传真此页给我们。欢迎您成为 WebSphere 开发与应用网站会员，在收到您的信息并审核通过后，我们将免费为您邮寄杂志。同时您可以登录 WebSphere 开发与应用网站：[www.WebSpherechina.net](http://www.WebSpherechina.net)，与更多 WebSphere 专家在线互动交流。

姓名：\_\_\_\_\_ 性别：男 女 电子邮件：\_\_\_\_\_ 电话：\_\_\_\_\_ 传真：\_\_\_\_\_

单位：\_\_\_\_\_ 部门：\_\_\_\_\_ 职位：\_\_\_\_\_

地址：\_\_\_\_\_ 邮编：\_\_\_\_\_

地址：北京市石景山区石景山路23号中融大厦530室 (100043) 广告：010-68867107-806 邮件：[chao\\_huang@twtgrou.com.cn](mailto:chao_huang@twtgrou.com.cn)





SunGard是世界领先的软件和IT服务企业，服务全球70多个国家，客户超过25,000位，包括世界上最大的25家金融服务企业。

SunGard为金融服务、高等教育和公共部门提供软件和流程解决方案，亦提供灾难恢复服务、IT管理服务、信息连续性咨询服务和业务连续性管理软件。

SunGard年收入超过50亿美元，位列财富500强第472位，在福布斯私有企业排行榜中位列私有软件和服务企业第一名。《Continuity, Insurance & Risk》杂志已史无前例的第五次将SunGard评为年度服务供应商。

更多信息，请登陆[www.sungard.com](http://www.sungard.com)。

- 全球第三大应用软件企业 (\*Datamonitor, 2009)
- 经纪商及其他金融机构每天利用SunGard系统处理的交易达500万笔
- SunGard系统目前为25万亿美元的投资资产提供日常管理服务
- 资本市场领域供应商第一名 (\*Source: American Banker/FinTech 100)

### 公司历史

SunGard成立于1982年，由Sun Oil Company以杠杆收购方式收购。公司于1986年在纽约股票交易所上市。2005年8月，银湖合伙公司 (Silver Lake Partners) 控制的私募股权基金公司通过杠杆收购以115亿美元代价将该公司转为私有。

# 一个空间，多种交流

## WebSphere开发与应用社区

<http://www.webspherechina.net>

WebSphere开发与应用社区  
是一个非盈利的用户组织，  
为大家提供交流学习、  
职业发展、  
商务合作的中文协作平台。

