

研究、探讨、传播

www.WebSphereChina.net 总第4期

WebSphere 中国

IBM WebSphere行业解决方案的专业纸质读物

WebSphere让世界互联互通



免费读物 审核赠阅



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

金融危机未能弱化医疗 IT 市场

——医疗 IT 风景独好



2009 年是我国医药卫生界充满震撼也是令医疗 IT 业十分兴奋的一年。4 月 6 日国家正式发布了《中共中央 国务院关于深化医药卫生体制改革的意见》，紧接着卫生部又陆续出台了诸如扩大社会医保范围、加大医保投入、推行基本药物制度、医保费用异地转移支付、公益性改革试点等改革措施。在这样的政策驱使下，医疗 IT 业出现的热词是“社区卫生”、“区域医疗”和“电子病历”。

在全球性金融危机的冲击之下，我国的医疗 IT 市场依然旺盛，PACS、LIS、HIS 等传统产品继续热卖，而临床信息系统（CIS），特别是电子病历、手术麻醉、心电和监护信息管理系统进入快速增长期，其中电子病历系统用户群体，在去年的基础上又增长了 30%—40%。虽然国内经济发展也受到全球金融危机的影响，但医疗 IT 仍然风景独好。

预计未来一年，随着新医改政策的推进，以“医院管理和电子病历为重点”的系统建设所带动的适应性服务需求将不断增加；各级政府将加快建设“以社区卫生服务中心为主体的城市社区卫生服务网络”；区域医疗将扩大实验范围，因此，社区卫生信息系统和区域医疗信息系统建设将渐入增长期，由此引起的各级各类数据中心建设和发展将会逐步提出外包服务的要求。

值得商家注意的是，用户在选择医疗 IT 产品时将更多的关注商家的品牌、业绩、信誉、知识含量和服务质量。

应用系统研发取得重大进展

医药卫生领域的信息系统研发，一般是从三大层面：社区卫生服务信息系统、医院信息系统和区域医疗信息系统的进展来考察。社区卫生服务信息系统建设开始大面积展开。这类信息系统研发和推广应用可说是今年最为显著的成绩。从事医疗 IT 的商家纷纷研制出各具特色的社区卫生服务系统，并在一些地区陆续投入使用。除了起步较早的上海市的长宁区、龙湾区、松江区之外，地处西南的成都地区、地处东北的大连市和北京市都有较大的研发和推广力度。

就整个区一级的社区卫生服务系统来看，功能最全、运行最好、作用发挥最为充分的要属上海长宁区的系统。在这个具有十个社区中心、约 70 万人口的一个区，已经为每一位居民建立了健康档案，实现了全区的所有医疗和公共卫生机构的包括 HIS、LIS、PACS 以及家庭病床管理在内的四大类十六个系统的整合，实现了全区的互联互通、信息共享，已经大大方便居民的保健就医，也为进一步降低医疗费用创造了条件。

新时期医院信息系统研发进入萌发期。我国目前广为使用的医院信息系统大多为上个世纪 90 年代中期陆续推出的，它的架构和功能已经不能满足新时期的内部和外部不断扩充的要求。在这种需求的驱使下，具有研发实力的医院或独立地或与企业联合决定研制新的版本。这个举动对于我国今后相当一段时间的医院数字化事业的发展将会产生重大影响。

值得业界深入思考的问题

提出问题，讨论问题，得出共识，本身也是前进。我认为有两个问比较突出，需要业内人士深入研究探讨。

第一个问题是社区卫生和区域医疗信息系统是否马上大面积铺开？

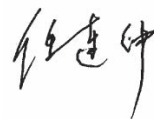
社区卫生信息系统（以下简称社区系统）和区域医疗信息系统（以下简称区域系统）是必须要建设的，

据说目前全国已经有六十余处正在建设区域系统，建设社区系统的就更多了，但是需要看到，这两件事都是新生事物，涉及部门多，涉及新的价值链，很多问题需进一步研究，不宜匆忙从事。我们随便举出一些问题：1) 很多地方建立了社区系统，但没有与中高层医疗机构互联，基层社区几乎没能享受到高层医疗机构的任何资源，很多功能放在那里用不上，辛辛苦苦建立起来的健康档案几乎成为“死档”，个别系统的运维消耗远远超出自身负担的能力。2) 区域系统建设建立了，医疗文档共享开通了，可是调阅率很低；“双向转诊”功能建立了，可是病员没能上下流动起来的；3) 还有的，看到区域系统热，于是各个业务部门便纷纷建立各自的区域业务系统，根本不知道区域系统本身是一个综合服务平台，它可以容纳很多种服务。4) 这两类系统都有运维如何落实问题，等等。这些问题如不认真组织研究并拿出解决办法，马上大面积铺开，势必造成新的麻烦和浪费。为此，提出建议如下：1) 集中力量加强模式研究，使系统真正可扩展、可伸缩、易管理，解决区域系统发展中若干根本性问题，如：系统的运行模式、可伸缩的架构模式、数据存储模式、各种服务流程和服务模式等都需加紧研究。2) 在已经具备条件的地区，要使社区卫生服务信息系统与区域医疗信息系统同步建设，借助区域医疗信息系统提供的各项服务，让底层的社区医疗机构享受到高层医院的人力和设备资源，提高基层社区的服务水平，增加服务内容，分担高层医院的负担。3) 结合本地区的实际情况做好整体建设规划，坚持在一个综合平台上开展多种应用，避免类似系统多头展开，依据实际需要和协调的可能安排出建设步骤和建设顺序，使系统建设有序有效地进行，避免“折腾”和浪费；4) 采取从小到大、逐步试点、逐步展开的发展策略。

第二个问题是我国新时期医院信息系统该由谁主导？

前已提到，目前我国各级医院广为使用的信息系统架构及核心功能大都是上世纪90年代中期设计的，现已过去十年，医院内部需求不断扩充、外部需求不断增加，新医改“意见”又提出了很多新的要求。因此原有系统的架构需要新版本的诞生。新版本的主题绝不仅是新技术的应用和简单的功能扩充，它不仅要解决当前渴望解决的若干问题，还要满足今后相当一段时间可持续发展的要求，诸如：在内部功能不断扩充和外部需求不断增加的情况下采用怎样的集成方案，且在这一方案支持下系统可方便的扩充而不影响系统的整体业务流程和系统的管理和使用；在数字化程度已经大大提升情况下业务流程优化成什么模样，使效率达到最高；建成什么样的电子病历系统，不仅能合理归档、长久保存，且还能方便地支持医疗、教学、科研以及公安政法等多方面的应用，并为无纸化保存创造条件；怎样让日益积累起来的宝贵数据随时产生出有价值的信息，等等。就是说，这一新的系统一定是既能满足当前需要，又能支持长远发展。谁来完成这一重任？跨国公司？他们眼界开阔、实力强、经验丰富，容易产生较为先进的理念，但是还应该看到，信息管理与国家政策和措施联系极为紧密，而且功能设计甚至操作方法还必须融于民族文化之中，由外乡人做这样的系统风险较大。本国公司？他们的经历较短，独立承担尚感能力不足。我本人看好的是既有丰厚经验又有设计能力的事业单位团队与实力较强的企业联合的团队，相信他们能够较好的完成新时期医院信息系统设计和推广应用这一历史使命。

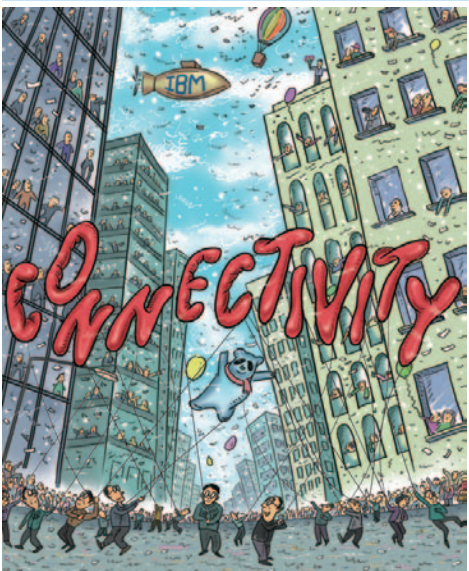
任连仲，解放军总医院原信息中心主任、医疗信息化专家，参与并组织了“军字一号工程”（军总医院信息系统）的设计和推广应用，与傅证共同主编了《医院信息系统建设与应用》一书。



www.WebSpherechina.net

p6 COVER STORY
封面报道

WebSphere 让世界互联互通



ESB 的建设,是企业建立 SOA 体系的重要环节,为企业各应用系统构建了一个统一、灵活、完整的整合平台。它很好的解决了企业内外新老应用系统之间互操作和信息交互的问题,最大限度的减少了应用系统互联所面临的复杂性,为企业开发后续应用及信息共享、数据整合打下了良好的基础,能够使企业适应未来在业务功能、处理流程、信息技术以及与外部异构系统通信等方面的发展需求。

IBM WebSphere MQ 备份和恢复



对于生产系统,备份和恢复是非常重要的。在出现紧急故障,例如磁盘损坏,文件以外删除,电源故障等,我们需要制定一个完善的系统恢复方案。IBM WebSphereMQ 产品架构某些方面很类似于数据库产品,例如 WebSphereMQ 的队列和数据库的表的功能很类似。在一个应用系统中,数据库的备份和恢复是非常重要的。

专家专栏 P36

卷首语

- 1 金融危机未能弱化医疗 IT 市场——医疗 IT 风景独好

新闻

- 4 IBM 完成收购软件公司 Lombardi 交易

封面报道

- 7 WebSphere DataPower XI50: 硬件化的 ESB
- 12 IBM WMB 概述及其 V7 的功能简介
- 17 IBM WESB 概述及其 V6.2 的功能简介

解决方案

- 21 中软国际基于 SOA 体系结构的卷烟营销总体解决方案套件解决方案
- 23 北京移动 BOSS 网管项目案例介绍
- 24 中科软银保通系统解决方案
- 26 中科软银行信贷风险管理解决方案

- 28 上海华腾个人信贷管理解决方案

· 前沿技术

- 30 WebSphere 应用服务器 Hypervisor 产品简介
- 31 WebSphere 家族新成员 CloudBurst
- 33 浅谈 WebSphere eXtreme Scale
- 35 WebSphere 应用基础平台总揽及最新技术介绍

· 专家专栏

- 36 IBM WebSphere MQ 备份和恢复
- 39 IBM WAS 6X 内存泄漏分析
- 44 使用 Websphere Business Monitor 监控 Message Broker 中的自动流

- 49 Websphere 性能优化之页面静态方案

· SMARTWORK 人行

- 54 SmartWork 三人行: 南京浦口区政府数据交换项目经验交流

· 博客

- 56 J2EE 应用性能问题分析
- 57 小型机设置 WebSphere 的默认编码全解密
- 57 关于 MB 集群的构架心得

· 有问必答

- 58 集群环境 WAS 负载均衡问题

WebSphere 中国

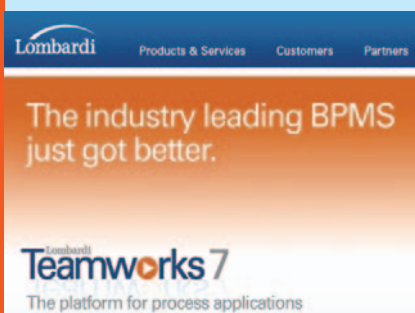
IBM WebSphere 行业解决方案的专业纸质读物

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: 黎素珍

名誉总编: 寇卫东
主 编: 黄 超
专家顾问: 毛新生 甘 荃
黄 河 张东焕
设 计: 肖 晔 (杞人)

《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 完成收购软件公司 Lombardi 交易



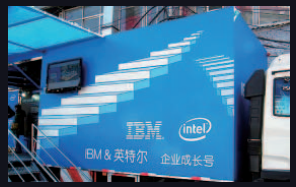
的自动化和一体化，以提高效率并降低成本。

收购完成后，Lombardi 将成为 IBM 应用一体化软件产品组合

IBM 近日宣布已经完成收购软件公司 Lombardi 的交易。Lombardi 是美国德克萨斯州一家私人持有软件公司，是业务流程管理软件和服务的领先提供商；该公司帮助企业实现业务流程

的一部分，这是 IBM 软件业务的一个重要增长领域，去年第四季度营收同比增长了 13%。IBM 预计，Lombardi 客户和合作伙伴对 IBM 及 Lombardi 技术的投资将得以保留，用户可以无需更换当前系统就可以享受到更广泛的功能。业务流程是每个组织的基础，保障了从保险索赔到银行贷款再到医院入院等各个流程的顺畅和及时完成。Lombardi 的产品使得组织能够迅速调整业务流程，应对突发和不断变化的需求，尤其是严重依赖于协作以完成任务或项目的组织。Lombardi 将给 IBM 的企业业务流程管理能力带来新的能力，增强了 IBM 实现流程自动化的能力，同时使得部门级别的经理能够及时改变已在运行的业务流程，降低了复杂和耗时技术干预的需要。

IBM：后金融危机时代信息质量决定企业成败



IBM 近日在北京发布报告称，后金融危机时代，企业已将经营重点从降低成本转移到创造收入上。海量信息

有可能成为企业寻找商机的负担，而高质量的信息才是确保企业收入增长的关键。

IBM 公司在《利用业务分析与优化实现突破》白皮书中指出，通过对全球近 400 名企业高层管理人员的调查，发现企业对于新商机的关注度明显提高。企业获得海量信息后需要对这些信息进行分析 and 优化，才能获得有效的市场信息。IBM 公司在调查中还发现，目前有 80% 的高管在做重大决策时都会面临信息不足或者信息不可靠的困境。目前，IBM 在公司内部使用云计算收集来自近 100 个不同的信息仓库和数据存储器的信息，并对超过 1000TB 的数据进行分析。

IBM 建 LotusLive 实验室旨在加速云协作

近日，在一年一度的美国 Lotusphere 大会上，IBM 发布基于创新技术拓展的云协作方案，并推出 LotusLive 实验室，旨在让用户体验和评估最前沿技术，以此推动云计算在协作服务领域的创新。

IBM 同时发布的云计算合作伙伴项目和云邮件协作项目，将为云计算发展注入全新动力。IBM 建立的 LotusLive



实验室是全新的研究实验平台，旨在加快 IBM 云协作领域的科技创新。LotusLive 实验室将发挥门户的作用，汇集新思路，进行基于用户的 LotusLive 应用界面的云协作技术的模拟和创新。IBM 遍及全球的知名研究科学家们都将致力于此的推广和创新。

目前，LotusLive 实验室已具备一批涵盖领域广泛的创新成果，包括能够协作完成并分享幻灯片的图库；能够记录并快速提供会议展示文字与音频材料的协作记录会议服务；以互动可视化方式显示会议日程安排的要务浏览图；借助 LotusLive 各项技术服务建立 LotusLive mashups 的组合技术等。

IBM 2009年第四财季净利润48亿美元

IBM近日发布第四季度财报时称,第四季度公司实现营收272亿美元,同比增长1%;实现净利润48亿美元,同比增长9%,每股盈利3.59美元,同比增长10%。据悉,第四财季净利润增

长主要受当季利润率上升及收入小幅增长提振。2009年全年实现营收958亿美元,与上年相比下降8%;实现净利润134亿美元,与上年相比增长9%,每股盈利10.01美元,与上年相比增长13%。IBM同时表示,预计2010财政年度每股收益至少达到11美元,公司对达到长期目标区间高端的能力抱有信心。

IBM董事长兼首席执行官彭明盛表示,第四季度业绩强劲,利润率、利润和每股盈利都再次增长,使2009年圆满结束;公司继续受益于战略转型和我们客户在这种经济环境中看重的产品,以及我们致力于世界发展中国家。



美国获专利最多企业 IBM连续17年排名第一

IBM日前宣布,公司2009年在美国获得4914项专利,与惠普相比几乎翻了两番,超过了微软、惠普、甲骨文、苹果、埃森哲和谷歌的专利数总和,连续第17年名列世界最有创新性公司首位。同时,IBM还宣布,将首次对外提供专业知识和专利组合管理软件,帮助以创新性成长为战略方向的公司增强知识产权

(IP)管理能力。此套软件可以帮助创新性企业有效地评估专利组合,寻找新的专利机会,更加紧密地将业务和知识产权战略结合起来。



IBM发布客户端应用进军移动领域

IBM近日宣布了其在苹果iPhone应用商店的第一个iPhone应用程序NotesTraveler。NotesTraveler是Lotuspushmail的客户端。通过Domino, iPhone的应用程序支持提供访问加密的电子邮件发送。Lotus产品管理总监埃德布利尔表示: Lotus软件也将为iPhone的Traveler应用加入电子邮件和日历增强功能。

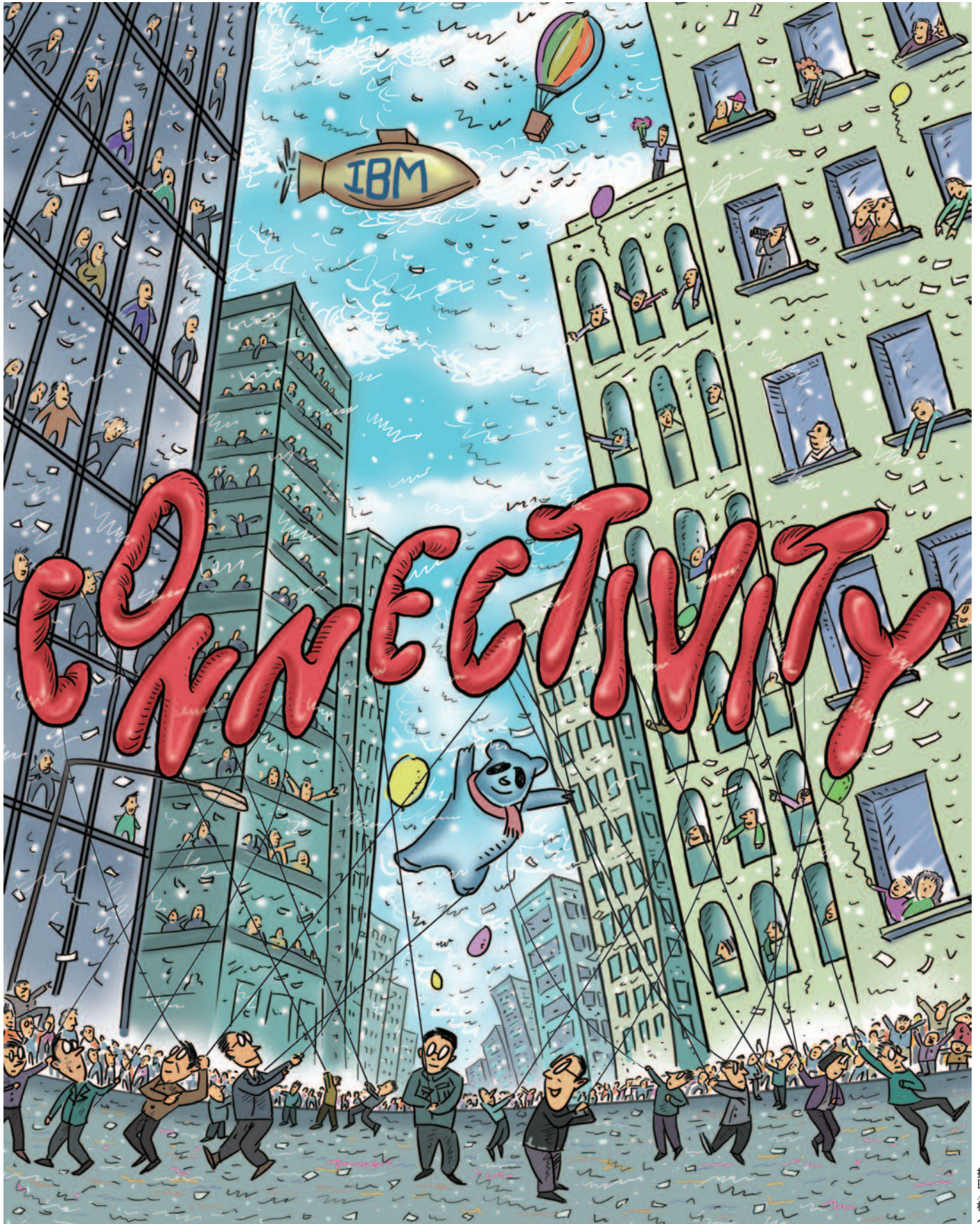
IBM将进军移动电话业务,计划再销售RIM的黑莓移动设备,并欲加载IBM的协作软件。IBM公司在其年度Lotusphere会议上透露,这项努力是2010年IBM移动战略的一部分。黑莓设备将提供IBM公司的LotusQuickr黑莓客户端。客户端提供了基于Quickr的内容共享和协作,它是一个内容仓库。此外,在黑莓的设备上它将是—一个为IBM的Lotus提供连接新的黑莓客户端,这是一种社会软件套件的RIM产品。



IBM将在日本成立“云计算事业”部门

IBM日前发布,2010年将在日本新设立“云计算事业”部门,由吉崎敏文先生负责。吉崎敏文表示,美国IBM有鉴于日本云计算市场的重要性,率先于日本设立直属于总部的云计算组织。IBM从去年1月开始,针对78国、2500位CIO进行调查(日本企业CIO占50%),内容为强化竞争力、云端运算与SaaS领域的计划。IBM认为云计算是“受网络服务影响,IT服务新型态”,将以虚拟化、标准化、自动化技术进行计划。IBM累积四十年的虚拟化技术实力,目前全球有10所云计算验证中心,9所云端信息中心,以IBM强项,进行与其他公司不同的云计算事业,预计发布Public Desktop Cloud、Public Test Cloud、Compute Cloud、Smart Analytics Cloud、Private Storage Cloud、Information Archive等服务。





插图/杞人

WebSphere DataPower XI50：硬件化的 ESB

寇卫东 崔鹏 / 文

IBM WebSphere DataPower Integration Appliance XI50 (以下简称 XI50)

是新一代的硬件化的 ESB 产品，它为 ESB 用户提供了新的选择。作为硬件化的 ESB，XI50 的各种功能通过硬件，固件及部分软件协同完成，充分满足了 ESB 的功能需求，从而简化了 ESB 解决方案的优势，在很多应用场景中，XI50 实现了“开箱即用”的 ESB，用户仅需接通电源，为其配置并联通网络，并通过 XI50 提供的 Web 管理控制台配置所需的服务，即可完成 ESB 的构建工作。本文将介绍 IBM WebSphere DataPower SOA Appliances 产品家族，并且特别介绍 XI50。我们将通过不同的场景示例，介绍如何用 XI50 构建 ESB。同时，我们也将讨论 XI50 的安全性功能，以及对 XML/Web Services 处理能力。XI50 可以为用户提供了一个安全的运行操作环境，其对数据处理能力可达到传统软件处理的 5 倍到 10 倍。

一、ESB 及其面临的新挑战

面向服务的架构 (Service Oriented Architecture, SOA) 已经在业界得到了广泛认同和应用。作为 SOA 的核心组件，企业服务总线 (Enterprise Service Bus, ESB) 有效地实现了分布式应用系统之间的“松耦合”式的整合，如图 1 所示，ESB 消除了采用异构技术带来整合壁垒，

并能够提供标准化的整合接口，成为连接企业各个应用系统的中枢。



图 1：ESB 原理示意图

从功能层面看，它主要提供了如下四方面能力：

- 路由：提供透明的服务请求路由；
- 转化：支持不同数据格式的转换；
- 协调：实现异种传输协议的桥接；
- 中介：处理来自不同业务源的事件信息。

IBM 陆续推出了 WebSphere Message Broker 和 WebSphere ESB 产品，在不同程度上满足了上述 ESB 的功能需求，并得到了实际应用。随着整合相关技术的不断发展和业界对 ESB 应用的不断深入，ESB 已经日益成为众多企业 IT 基础设施的核心组件，例如，在某些大型企业中，60% - 80% 的分布式应用系统间交互都通过 ESB 完成。由此，ESB 对于企业 IT 系统的重要性逐步凸显出来，进而对 ESB 的非功能性需求在不断提高，成为 ESB 面临的新挑战，其中，以下三方面的需求尤为突出：

● 提高 ESB 的开发和运维效率，以适应不断变化的业务需求，并降低相关的成本；

● 增强 ESB 可靠性和安全性，以保障核心业务的稳定运行，提升 IT 系统服务水平；

● 提升 ESB 的处理能力，特别是针对 XML/Web Services 的处理能力，以满足日益增长的业务需求；

在应对上述挑战的策略中，有这样一种解决思路：如果类比于网络层，ESB 可以看作是一种应用或服务的路由器，在相对成熟和标准化的环境中，是否可以用类似网络层的硬件设备来实现 ESB 呢？事实上，如图 2 所示，这个思路符合 IT 技术的发展的一种趋势：通用标准的功能实现通常将逐步由软件转向硬件。

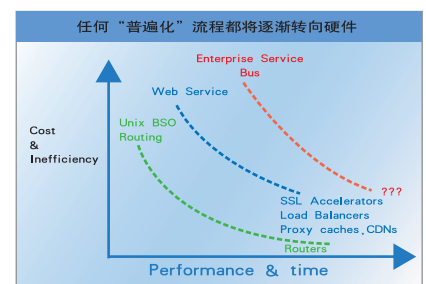


图 2：IT 产品硬件化趋势示意图

如果采用硬件设备实现 ESB，相对于软件解决方案，它将带给用户更加简化的开发管理体验，更高的稳定性，更好的性能以及更低的成本。IBM 作为业界



领先的中间件提供商，率先推出了新一代的硬件化的 ESB 产品——IBM WebSphere DataPower Integration Appliance XI50（以下简称 XI50），为 ESB 用户提供了新的选择。

二、WebSphere DataPower SOA Appliances 及其 XI50 简介

XI50 是 IBM WebSphere 品牌下 DataPower SOA Appliances 产品家族的成员之一。DataPower 产品家族在 IT 市场已经有超过 10 年的发展历程，诞生之初，其市场定位为处理 XML/Web Services 的硬件加速设备。在不断发展过程中，衍生出了面向应用处理加速、应用/服务安全、应用/服务整合、B2B 以及低延迟传输等 5 款硬件化的中间件产品，它们是：

- XML Accelerator Appliance XA35: XML/Web Services 处理加速设备；
- XML Security Gateway XS40: 面向应用/服务安全控制的安全网关；
- Integration Appliance XI50: 面向应用/服务整合的硬件 ESB；
- B2B Appliance XB60: 面向 B2B 的交易网关；
- Low Latency Appliance XM70: 支持低延迟传输的高效传输中介。

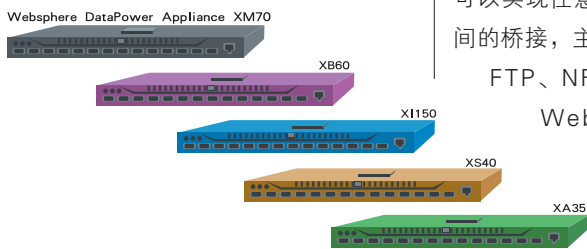


图 3: IBM WebSphere DataPower SOA Appliances 产品家族

如图 3 所示，每款产品都设计独特，外形规格一致，易于部署（1U，可机架式安装），通过以太网（提供 4 个 1Gb 以太网接口）与外部系统进行连接，通过简单的配置（几乎无需编写程序）即可无缝地嵌入到现有的 IT 基础架构中。

XI50 作为硬件化的 ESB，其各种功能通过硬件，固件及部分软件协同完成，充分满足了 ESB 的功能需求，它提供的应用/服务整合能力包括：

- Any-to-Any 的数据格式转化: XI50 可以支持 XML、Flat file 以及 binary 数据之间的任意格式转换，例如不同格式的 XML 数据之间转换，或 XML 到 CSV 或 ISO8583 之间的转换。XI50 内置了“DataGlue”数据转换引擎，该引擎可以支持用户自定义的映射规则，从而完成各种数据格式之间的转换。
- 基于内容的路由: XI50 可以基于接收到的消息的任意部分内容进行路由，例如，对于 XML 格式的消息，可以基于消息体中的特定字段的内容，对于 JMS 消息，可以基于消息头中的字段。同时，XI50 还可以基于消息携带的用户身份信息进行路由，例如 Web Services 消息中的用户 token。此外，XI50 还可以支持一点对多点的路由。
- 多种传输协议的桥接: XI50 支持多种常用应用层协议和消息中间件，并可以实现任意两种协议或消息中间件之间的桥接，主要包括：HTTP、HTTPS、FTP、NFS、WebSphere JMS、WebSphere MQ 和 TIBCO EMS 等。同时 XI50 支持对主流数据库访问，可以基于 SQL 直接操作数据库中的数据，包括调用存储

过程。

此外，XI50 还提供了众多的针对应用/服务层的安全功能，支持多种安全标准，并支持基于流量控制的服务级别管理，支持对应用/服务的负载均衡。

XI50 支持 Active/Standby 和 Active/Active 两种模式的高可用工作模式，确保 ESB 方案的可靠性；XI50 具备良好的可扩展性，它支持水平扩展，可以通过集群方式对外提供服务。

XI50 提供基于 Web 的管理控制台，也支持基于 SSH 或 telnet 的命令行的管理接口，提供图形化的监控方式，也支持基于 SNMP 的监控。这些功能都为实现 ESB 的解决方案提供有力的支持。

三、XI50：“开箱即用”的 ESB

作为硬件化的 ESB，XI50 天然具备了简化 ESB 解决方案的优势。在很多应用场景中，XI50 实现了“开箱即用”的 ESB，用户只需接通电源，为其配置并联网，并通过 XI50 提供的 Web 管理控制台配置所需的服务，即可完成 ESB 的构建工作。以下，我们通过两个例子来进一步说明。

协议桥接及消息路由的场景示例

在图 4 的场景中，XI50 需要接收 WMQ App 基于 WMQ 发送的 XML 消息，并根据消息内容，基于 HTTP 将 XML 消息路由到后端的 Web App A 或 B。

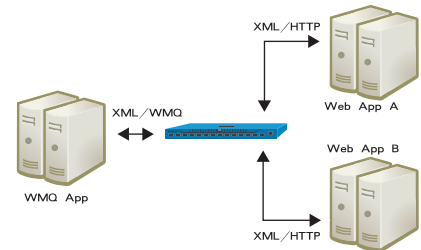


图 4 XI50 的协议桥接及路由应用场景



寇卫东 寇卫东教授，IBM 软件集团两岸三地大中华区总工程师（总架构师），国际知名专家，美国马里兰大学兼职教授，中国通信

学科第一位博士、第一位博士后，有接近 30 年国内外 IT 行业经验，他的客户遍及中央政府及其部委、省、市、区等地方政府，以及石油化工、汽车、制造、交通运输、零售、银行、保险、电信、中小企业等行业。寇教授 IBM 工作近 15 年；曾在北美西门子、美国 AT&T、加拿大 IBM、瑞典林彻平大学、加拿大滑铁卢大学、香港大学等境外的公司和大学工作 17 年，设计和领导开发大型软件系统，教授研究生课程，曾任首席工程师、首席研究员、常务副总监等职，曾 6 次担任国际会议大会主席，出版了 7 部英文专著、拥有 20 余项美国、加拿大、中国等国家专利、发表了 100 余篇论文；回国后，曾任国家重点实验室主任、计算机学院院长；1992 年当选为纽约科学院院士；2004 年荣获中国政府授予国际专家的最高荣誉，并获温家宝总理等国家领导人接见、颁奖、合影、宴请。

为实现上述场景，首先，用户需要登录到 XI50 的 Web 管理控制台，并创建 Multi-Protocol Gateway（以下简称 MPGW）服务（如图 5 所示）。

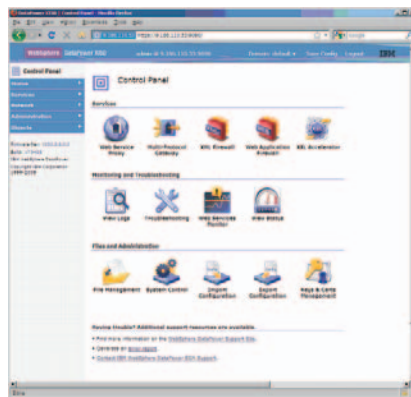


图 5: XI50 Web 管理控制台

在 MPGW 的配置中，用户仅需要完成如下三个步骤的配置。

(1) 配置连接 WMQ App 的对象 MQ Front Side Handler（以下简称 MFSH）：如图 6 所示，在 MFSH 中，仅需提供 WMQ 队列管理器的信息和相关的输入、输出队列即可。

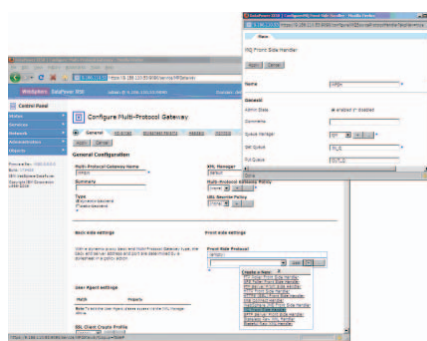


图 6: 配置 MQ Front Side Handler

(2) 配置消息处理策略：如图 7 所示，在处理策略中，需要配置处理 request 消息的 Client-to-Server 规则和处理 response 消息的 Server-to-Client 规则。每个规则，由一组处理节点组成，并以消

息匹配（Match）节点为起始，对于 Client-to-Server 规则需要配置路由（Route）节点，在路由节点中需要配置相应的路由策略，而对 Server-to-Client 规则，无需对消息进行处理，仅需要配置结果（Result）节点。配置过程仅需通过“拖拉点选”即可完成。

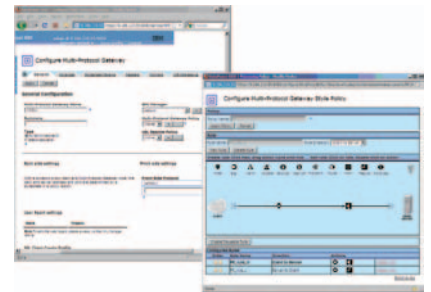


图 7 配置 MPGW 消息处理策略

(3) 配置路由节点中的路由策略：如图 8 所示，MPGW 会依据 XML 消息中的 Service 字段的内容，确定路由的目标。在路由节点中，采用 XPATH 的方式指定路由的条件，同时指定对应的路由目的端的主机和端口。

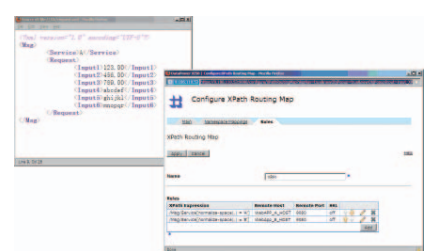


图 8 XML 消息样例和路由节点中基于 XPATH 的路由策略

通过以上三个简单的配置步骤一个具备协议桥接和内容路由功能的 ESB 就构建完成。

异构数据格式转换场景示例

在如图 9 所示的场景中，XI50 需要实



现HTTP到WMQ的协议桥接,以及XML到COBOL Copybook的数据格式转换。本示例将着重探讨数据格式转换的实现。



图9 XML到COBOL Copybook的数据格式转换场景

为实现XML到COBOL Copybook的数据格式转换,用户需要完成三个主要步骤:

(1) 定义不同格式数据之间的映射规则: XI50支持基于IBM WebSphere Transformation Extender Design Studio (WTX DS) 订制的映射规则描述文件。在本场景中,用户需要把描述XML数据格式的Schema文件和描述COBOL Copybook数据格式的文件导入WTX DS中,如图10所示,通过图形化的操作进行匹配,并生成 dpa 格式的映射文件。

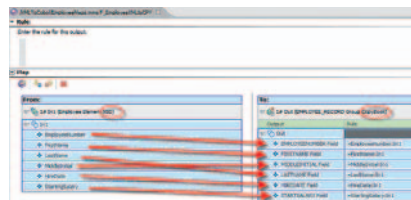


图10 基于WTX DS订制数据映射规则

(2) 增加格式转换 (Transform) 节点: 在XI50的服务配置过程中,用户需要在消息处理策略的 Client-to-Server 的规则中增加格式转换 (Transform) 节点,如图11所示。

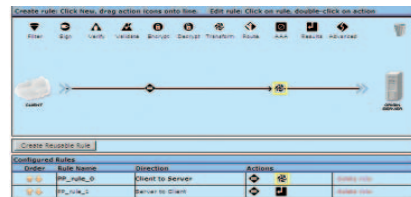
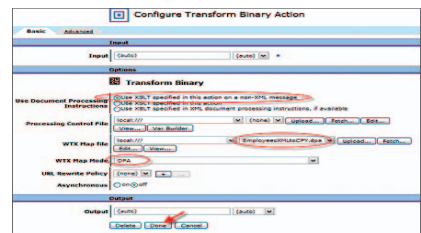


图11 消息处理策略中的格式转换节点

(3) 配置转换节点: 用户仅需要将 dpa 格式的数据转换文件上传到XI50的本地目录中,并如图12所示,在转换节点中指定该文件即可。



通过以上简单的配置步骤,基于XI50的ESB就具备了支持特定格式的XML到COBOL Copybook的数据格式的转换。

场景示例小结

通过以上示例,不难看出,相对软件的ESB解决方案而言,XI50以“开箱即用”的使用方式大大简化了开发和运维工作。

就开发方面而言,用户几乎不需要做任何开发工作,仅通过Web管理控制台进行配置工作,即可实现大多数ESB的功能(XI50也支持基于XSLT的开发),大大简化了开发工作,降低了开发成本,提升了投产速度。

就运维方面而言,对于基于XI50的ESB解决方案,用户无需分别维护服务器、操作系统、中间件以及应用。同时,作为硬件化的ESB专用设备,XI50提供了简单易用的管理维护方式,大大降低了运维成本,提高了运维效率。

四、XI50: 安全、高效的ESB

XI50在实现ESB基本功能的基础上,增强了应用/服务的安全性方面的功能,并继承了DataPower产品家族高效



崔鹏 IBM(中国)有限公司软件部,高级信息工程师(Certified IT Specialist),北京航空航天大学计算机学院硕士,2004年加入IBM公司软件部。主要负责支持WebSphere Infrastructure及Cloud Computing相关产品的技术支持,曾参与银行、电信、政府,交通及能源等行业多个大型IT项目建设和IBM重要客户IT软件架构规划。

处理XML/Web Services的专利技术。以下分别详细介绍。

安全的专用设备和面向应用/服务层的安全功能

XI50 作为专用的硬件化 ESB，自身提供很好的安全性，为用户提供了一个安全的运行操作环境：

- XI50 是一个封闭的设备，对外接口仅为用于安装初始化的串口和以太网接口，避免不必要的干扰；

消息、部分消息以及字段级别的加密及解密，支持 SSL，其加密/解密算法和密钥均遵循开放标准。

- 数字签名功能：XI50 支持对完整消息、部分消息以及字段级别的数字签名及验证功能，其签名/验证算法和密钥均遵循开放标准。

- 访问控制功能：XI50 实现了一个灵活有效的认证，授权及审计的框架。任何消息在到达 XI50 之后，该框架会分别抽取客户端所要访问的资源及认证标识信息。对于取得的用户标识，XI50 可以通过内部定制访问控制策略或者外部的访问控制服务器（例如 Tivoli Access Manager、Tivoli Federated Identity Manager、RSA ClearTrust 及 CA eTrust 等），进行用户认证。对于通过认证的用户，XI50 将进一步通过访问控制文件或者外部的访问控制服务器确认该用户是否对所请求的资源具有访问权限，如通过授权验证，XI50 还将提供审计功能，再将该消息发送给服务提供者。

- 应用/服务层攻击防护功能：XI50 内置了多种应用/服务层攻击的防护机制，主要可以防护 XML 实体膨胀和递归攻击，XML 文档大小和深度攻击，XML Flood 攻击，恶意包含外部实体攻击，

Public Key DoS，SQL 注入攻击和跨站点脚本攻击等。

以上安全功能的实现都可以通过 Web 管理控制台配置实现，并可以方便地集成于消息处理流程中，与其他功能进行灵活的组合。同时通过 XI50 可实现应用/服务的安全网关，实现对应用/服务安全的集中管控和灵活的管理，使安全解决方案与应用/服务的实现“松耦合”。

图 13 给出了 XI50 的一个典型应用场景，该场景中，XI50 以集群方式提供服务，它既作为 ESB 提供路由、数据转换和协议桥接的核心功能，又提供的应用/服务安全网关所具有的访问控制、数据加密、数字签名、数据校验、数据过滤和攻击防护的功能，因此，在此场景中，XI50 有效实现了网关模式（Gateway Pattern）的 ESB。

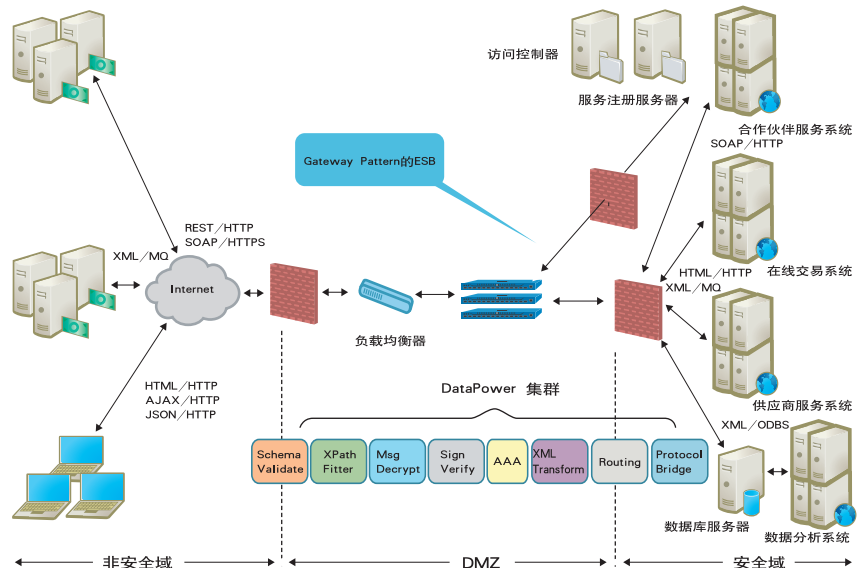


图 13: XI50 的典型应用场景

“线速”数据处理能力

高性能是 XI50 作为硬件化的 ESB 的

又一显著优势。相对于软件 ESB，在 XI50 内部，很多原来由软件实现的功能均由硬件或固件替代完成，因而大大提升了处理性能，特别是 XI50 内置了具有专利技术的 XML 协处理器，大幅提升了对 XML/Web Services 的处理能力。通常，XI50 的对 XML/Web Services 处理能力可达到传统软件解决方案的 5 倍以上。

图 14 给出了 XI50 和软件解决方案对某 XML 安全任务处理结果对比，该任务为 XML 的加密任务，可以分解为 XML 的处理任务和加密任务两部分，在 XML 加密任务中，又包括部分 XML 处理任何和纯加密处理任务。传统的软件方法不能对这些任务实现有效的加速，对于具有加密加速特性的软件解决方案，也仅能提升纯加密处理任务的处理速度，因而性能空间提升有限。而对于 XI50，不仅可以提升纯加密处理任务，更重要的是

它可以大幅提升 XML 的处理速度，所以在该测试中，XI50 对 XML 加密任务的处



理能力达到了软件方案的 10 倍。

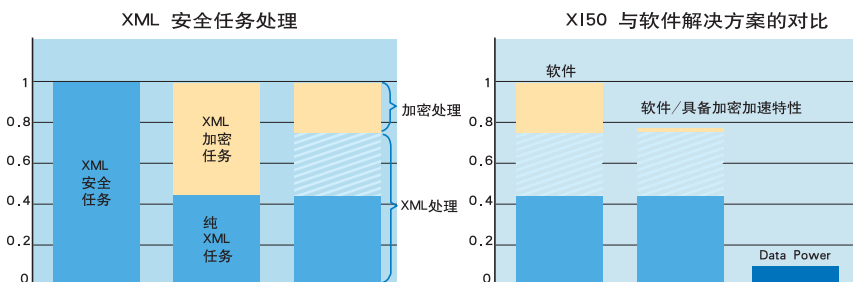


图 14 XI50 和软件解决方案对某 XML 安全任务处理结果对比

在 XI50 众多的应用案例中，XI50 曾成功解析并处理了数据量为 1GB 的 XML 消息，单一 XI50 也支撑过每秒 25000 个交易，其高效的处理能力因而赢得的广大用户的信赖。

五、总结

目前，IBM WebSphere DataPower Integration Appliance XI50 作为 IBM 新一代的硬件化企业服务总线，已经广泛地服务于 IBM 全球的关键客户，并发挥

着重要作用。XI50 提供了完整的 ESB 功能，并针对提高开发运维效率，提升 ESB 的稳定性和安全性，加速 ESB 的处理效率等非功能性需求进行了全面增强。

XI50 的“开箱即用”特性使其多次在与软件 ESB 解决方案的竞争测试中取得明显优势，基于 XI50 构建 ESB 实现相同测试场景所需的时间通常小于软件方案用时的 50%。在实际的生产环境中，基于 XI50 的构建的 ESB 的部署时间也远远小于基于软件 ESB 的方案。图 15 给出了

某项目中，基于 XI50 的 ESB 方案与传统软件 ESB 方案对比，通过采用 XI50，用户将以更低的成本，实现更短的生产时间和更低的风险。同时，XI50 提供的应用/服务层安全功能将有效保护 ESB 以及后台的应用系统，确保业务系统的稳定运行；XI50 提供的高效的处理能力也将帮助用户更好地支撑业务运行。

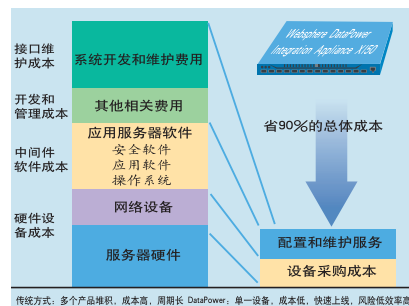


图 15：基于 XI50 的 ESB 方案与传统软件 ESB 方案对比

总之，XI50 为构建 ESB 方案提供了有价值的新思路，为简化、保护并加速 ESB 解决方案提供了坚实基础。

IBM WMB 概述及其 V7 的功能简介

寇卫东、严国军

IBM WebSphere Message Broker (简称 WMB) 是 IBM 的高级 ESB 产品。它能够以智能的、实时的方式，从多种设备类型、业务单元和位置向几乎遍布企业 (甚至企业之外) 的全部系统和应用，路由和转换消息与数据。WMB 提供了图形化的开发工具 Message Broker Toolkit。在 Eclipse 的基础上，使用 WMB 能够让用户方便快捷的开发出消息流应用程序。本文将介绍 WMB，包括它的架构、开发与运行环境、以及它的特性。

一、企业服务总线概述

企业服务总线 (Enterprise Service Bus, ESB) 作为面向

服务的架构 (SOA) 的核心组件，是实现服务间智能化集成和管理的中介。它是逻辑上与 SOA 所遵循的基本原则保持一致的服务集成基础架构，提供了服务管理的方法和在分布式异构环境中进行服务交互的功能。

企业服务总线所提供的服务

一般来讲，ESB 应该提供如下三类服务：

1. 传输服务

(1) 安全可靠的数据传输

数据在传输时，ESB 必须保证其从发送方到接收方的过程

是安全可靠的，不丢失，不复传。

(2) 持久性/非持久性

ESB支持数据的持久性。持久性适用于可靠程度高而性能要求一般的数据，即使宕机也不允许数据丢失，如银行的批量对帐报文。非持久性适用于性能要求高而可靠性一般的数据，即使数据丢失也能通过重发来完成，如银行的实时交易报文。

(3) 同步/异步

ESB支持系统间的同步和异步通信。同步模式是请求/应答方式，请求方在发出请求的同时，需要等待接收方的回复，ESB依次传输请求和应答两类消息。在异步模式下，发送方只要送出数据即可认为传输完成，ESB保证数据可靠传达。

2. 仲裁服务

(1) 路由

ESB是为不同的系统提供连接服务的，各系统之间的数据传输，由ESB从发送方路由正确的接收方。这种路由的实现可以是基于传输数据的内容，也可以是基于外部数据源定义，或者可以基于用户自己的算法规则。其支持的路由方式也可以有多种形式，如单向传递、请求/响应、发布/订阅、服务的分解与组装等。

(2) 格式转换

ESB所连接的系统是多种多样的，每个系统可能都有自己的数据格式。在系统之间进行数据交换时，需要ESB把源端数据格式转换成目的端数据格式。

3. 事件服务

(1) 事件的发现和发布

ESB处理的外部事件，既可以包括系统间简单的消息交换，也可以是对外系统的事件探测。同时，ESB支持通过时间点来触发新事件，或者从当前的处理过程中生成新的时间事件。

(2) 发布和订阅(Publish/Subscribe, Pub/Sub)

信息的发布者发布特定主题的信息，信息的订阅者订阅自己感兴趣主题的信息，两者无须关心对方的存在。当连接ESB的系统作为发布者发布某一特定主题的信息时，所有连接到ESB并订阅了该主题的系统都能收到ESB发送过来的该信息。发布系统和订阅系统是互相透明的。

企业服务总线的功能

ESB对外提供的服务由其自身的功能保障，一般说来，

ESB至少具有如下功能：

(1) 多通信协议的支持

ESB是接入多个系统的连接层，各个系统可能使用不同的通信协议，ESB必须提供多种通信协议的接入。

(2) 灵活的数据模型

ESB要完成各个系统间不同数据格式的转换，就需要有自己灵活的数据模型，能够为不同系统的数据结构建模，定义不同的数据格式。

(3) 流程处理

ESB需要有自己的流程处理来完成通信协议的变换、消息格式的转换、消息的路由等工作。

(4) 服务质量的控制。

作为SOA架构的核心组件，ESB需要稳定可靠地运行、性能优异、灵活且易于扩展。

(5) 系统的安全控制及监控

维护人员对ESB系统的访问需要进行安全级别控制，能提供友好的工具监控整个系统的运行。

企业服务总线实施产品介绍

ESB是逻辑架构的一个组件，市场上能够实现ESB的产品很多，如IBM公司的WebSphere Message Broker、WebSphere ESB、WebSphere DataPower，Oracle公司的Oracle Service Bus(OSB)，微软公司的BizTalk Server，TIBCO公司的BusinessWorks，金蝶公司的Apusic ESB，等等。

WebSphere Message Broker提供了一个能满足客户的应用集成和信息调解需求的解决方案。相对于其它的ESB产品，WebSphere Message Broker以其强健的设计、丰富的功能、可扩展的架构、优异的性能、便捷的操作，能够轻松帮助客户生成一个先进的企业服务总线，逐步实现企业范围内的面向服务的架构，能够连接企业内部和外部合作伙伴的各个系统和应用，使发生在企业基础架构内所有业务事件具有更快更大的可视化。

二、WebSphere Message Broker 概述

WMB 产品概述

WebSphere Message Broker V7能够以智能的、实时的方式，从多种设备类型、业务单元和位置向几乎遍布企业（甚至企业之外）的全部系统和应用，路由和转换消息与数据。

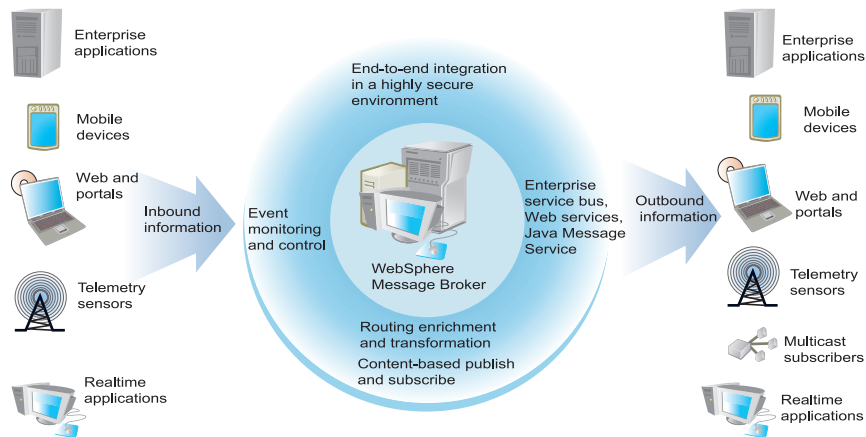


图1: IBM WMB图例



严国军 IBM(中国)有限公司 软件部,2006年6月加入IBM,主要从事IBM业务整合中间件的技术支持工作,拥有丰富的产品经验,支持的产品包括WebSphere MQ、WebSphere Message Broker、WebSphere MQ File Transfer Edition等,同时,具有很多大型项目的实施经验,曾参与中国人民银行国库系统、建设银行EAIH项目、辽宁网通BOSS系统、江苏电信BSS EAI等项目的实施。

支持的产品包括WebSphere MQ、WebSphere Message Broker、WebSphere MQ File Transfer Edition等,同时,具有很多大型项目的实施经验,曾参与中国人民银行国库系统、建设银行EAIH项目、辽宁网通BOSS系统、江苏电信BSS EAI等项目的实施。

作为涵盖流程集成、人力管理和企业应用连通性的业务集成解决方案,WebSphere Message Broker能够帮助客户:

- 生成并部署新的业务程序;
- 同步多个系统中的业务事件;
- 整合不同平台上的应用;
- 在应用之间转换信息格式并且实现路由;

同时,随着业务的扩展和系统整合的深入,WebSphere Message Broker能够帮助客户在原有的基础上轻松实现

- 在不增加复杂度的情况下扩展企业的IT基础架构;
- 保护企业在应用和数据上已有的和正在进行中的投资;
- 无缝的扩展ESB的连接能力。

WMB 产品架构

WebSphere Message Broker 的产品架构如下图(图2)所示:

WebSphere Message Broker 主要包括两个部分:开发环境和运行环境。

1. 开发环境

WebSphere Message Broker 提供

图形化的开发工具 Message Broker Toolkit,它是建立在Eclipse的基础上,能够让用户方便快捷的开发出消息流应用程序,用于将逻辑提供给代理。代理使用该逻辑在运行时处理来自业务应用的消息。在Message Brokers Toolkit中,用户可以实现消息流应用的开发、管理、调试和部署等一系列功能。用户开发的消息流应用主要包括两类:消息流(Message Flow)和消息集(Message Set)。

- 消息流是提供逻辑的程序,代理用消息流处理来自业务应用的消息;
- 消息集定义了代理中处理的消

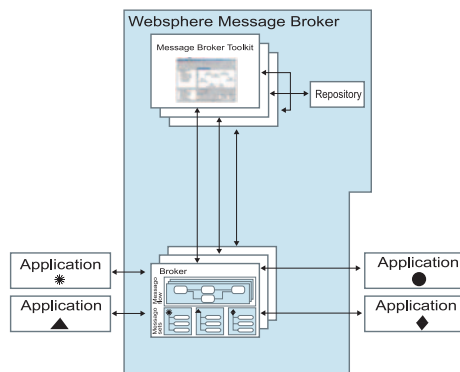


图2: IBM WMB架构示意图

息流的消息结构。

2. 运行环境

运行环境是一个组件集，用于在代理中部署、运行消息流应用程序。组件包括：

- 代理 (Broker)：是 WebSphere Message Broker 的消息处理引擎，它提供 WebSphere Message Broker 的所有运行时服务，在 Windows 系统上它是一个系统服务，在 Unix 平台上它表现为一个后台进程。代理根据消息集和消息流的定义，来路由每个消息，并且对消息进行各种处理，必要时同时按照接收端需要的消息格式进行格式转换。

- 执行组 (Execution groups)：执行组将代理中的消息流分组，是若干消息流的组合，是消息流运行引擎。每个执行组是一个独立的进程，这样，在不同执行组中的消息流就可以做到在运行时相互独立；在一个执行组内部，消息流在不同的线程池内运行，为了提高性能，我们可以通过设置每个消息流的运行实例的个数来指定每个消息流的线程池的大小。

- 消息流：在 WebSphere Message Broker 中对消息的运算处理、格式转换和路由等功能是通过消息流实现的，每个消息从输入 Broker 到从 Broker 中输出，都将被一个消息流处理，然后发往目的应用系统。消息流由各种消息处理节点 (Message Processing Node) 组成，消息处理节点可对消息进行各种处理操作，节点与节点相连，便组成了一个消息流。

- 消息集：由于流经代理程序的消息格式的多样性，必须能够通过某种途径区分不同类型的消息。每种消息类型的定义，或具有相关性的一组消息类型的定义，被描述为一个消息集 (Message Set)，每个消息集拥有一个不同的字典。当接收到一个消息时，消息头中的信息可以帮助确定将被载入的正确字典，以对消息进行解析。为将其用于代理程序，一个消息集中所有相关的消息都会被指定给一个或多个代理程序，这些代理程序从指定的消息流中接收消息并对其进行处理。

消息格式使用消息模版 (Message Template) 来定义并被存储在消息存储库中，消息存储库由消息库管理器 (Message Repository Manager，MRM) 来管理，MRM 是 WebSphere Message Broker 中定制消息的主要工具。MRM 中的消息模板由以下四个数值来定义：

- 消息域 (Message Domain)，它描述了消息定义的来源，

消息域有如下几种：MRM，预定制格式；XML，自定义格式；BLOB，无需解析的格式；JMS，JMS 消息；等等。

- 消息集或项目 (Message Set)，它将某一特定域内的消息、单元和类型集聚集在一起，创建一个与某一特定消息流或商业操作相关的完整的消息定义。

- 消息类型 (Message Type)，它精确地定义了消息内部的数据结构，可以提供像字符串的数量和位置这样的细节。

- 消息格式 (Message Format)，它确定了消息的线格式。

三、WebSphere Message Broker 的功能特性

1. 广泛的连通性

ESB 作为企业内外各系统交互的中枢，其与各系统的连接能力，是考核 ESB 功能的一个重要指标。WebSphere Message Broker 集成了多种通讯方式，涵盖了企业应用、移动手持设备、互联网门户、实时系统和遥感终端，扩展了企业集成总线的边界、范围和规模，具体表现在：

(1) 广泛的标准协议的支持

WebSphere Message Broker 提供众多的内置节点，全面支持目前市场上常见的连接协议，包括：HTTP/HTTPS、WebSphere MQ、JMS、Flat File/FTP/SFTP、TCP/IP、SMTP、SCA、等等。

(2) 支持各种 Web Services 和 Web 2.0

不仅支持基于 SOAP 协议的各类 Web Services，如 SOAP/HTTP、SOAP/JMS、SOAP/MQ 等，而且还支持其它的 Web Services 标准，如 HTTP REST、XML-RPC 等。同时，WebSphere Message Broker 还支持多种 Web Services 的标准，如：WS-Security、WS-Addressing 等。

(3) 无缝集成商业标准的数据库

支持 SQL V3 标准，可以直接通过 SQL 语句对数据库表进行增、删、改、查等操作，还可以通过图形化工具对数据库表配置。用户可以使用多种内置节点与 ODBC/JDBC 数据源进行交互。全面支持市场上主流的商业数据库，如：DB2、Oracle、Sybase、Informix、SQL Server 等。

(4) 与多种企业信息系统 (EIS) 的整合

以内置节点的方式提供 WebSphere Adapter，支持与 SAP、Siebel、PeopleSoft 等多种企业信息系统的轻松整合。同时，WebSphere Message Broker 还提供企业元数据发现



(EMD)工具，快速、简捷的查找并构建关键的企业数据结构。

2. 灵活的格式转换

由于ESB连接的系统是在不同时期，采用不同的架构、技术建立起来的，它们的数据所展现的消息格式都不尽相同，所以，解析、转换这些异构的消息格式成了ESB的又一重要功能。

WebSphere Message Broker 提供各种内置的消息解析器，无需用户编程，即可实现对大部分消息格式的解析，如：

- XML 格式，包括普通XML 报文、XML DTD、XML Schema 等。

- 面向记录的消息格式，如：C 的头文件、COBOL 的 copybook、JMS 消息、分隔符消息等。

- 各种行业标准或大型系统数据标准，如：EDIFACT、SWIFT、HL7、ACORD、X12、TLOG、IDOC 等。

对一些非标准的、用户自定义的数据格式，WebSphere Message Broker 提供了图形化的开发工具，帮助客户预定义消息格式。

在实际运行过程中，WebSphere Message Broker 支持对数据进行格式和内容的双重校验，有效防止垃圾数据的侵入。

同时，WebSphere Message Broker 提供多种方式实现消息格式的转换：

- Mapping 节点：提供图形化的工具支持消息格式的映射。

- XSLTransform 节点：以XSL 样式表实现XML 数据格式之间的互相转换。

- Compute 节点：以ESQL 语言让用户灵活的编程实现数据格式的转换。

- JavaCompute 节点：以Java 语言让用户灵活的编程实现数据格式的转换。

- PHPCompute 节点：以PHP 语言让用户灵活的编程实现数据格式的转换。

3. 智能路由

ESB 的一个重要特点就是服务的虚拟化，服务的请求者和服务的提供者之间互相透明，无需关心对方的所在，这就需要ESB 具备动态的智能路由能力。WebSphere Message Broker 支持多种规则的动态路由：

- 基于消息的内容。

- 基于外部的数据源，如：数据库中的路由表、路由配置文件、服务注册库（如WSRR）。

- 基于用户的业务逻辑或算法。

同时，WebSphere Message Broker 支持在不同的路由规则下对消息的各种处理，如消息的过滤，包括基于某个字段的条件过滤，或组合条件的满足等；对消息的组合处理，如把一个消息拆分成多个消息，或者把多个消息合成一个消息等。

4. 简易的安装配置

相对于以前的版本，WebSphere Message Broker V7 提供了更为简易的安装和配置过程。在新版中，数据库的依赖已不复存在，被部署的资源信息和配置信息将被存放在文件系统中，中间状态信息将被存放在MQ 中；WebSphere MQ 成为唯一需要预先安装的产品；组件配置管理器（Configuration Manager）和用户名称服务器（User Name Server）都已被取消，代理（Broker）是唯一需要创建的组件。

5. 组件化的开发

WebSphere Message Broker 提供基于Eclipse 的图形化开发工具Toolkit，使熟悉Eclipse Java 开发的用户非常容易上手。Toolkit 不仅为用户提供图形化的组件开发模式，还集开发、管理、部署、调试等功能于一身，极大的简化了用户的开发进程，尤其是对远程调试的支持，更是Toolkit 的一大亮点。

在WebSphere Message Broker V7 中，更是推出了基于模式（pattern）的快速开发模式。“模式”可以被认为是一种在特定的情景下的、被充分验证过的、具备通用的架构和设计的“半成品”的解决方案，可以被无限重用。用户可以根据需要，从现有的模式中生成当前的解决方案框架并逐步完善，这极大的简化了用户开发的工作量，提升了用户开发的效率。

6. 管理和监控

WebSphere Message Broker V7 新推出的MB 资源管理器，与MQ 资源管理器紧密结合，为用户提供了一个单一的管理控制界面，可以同时管理MQ 和MB 的各种对象，实现了管理方式的统一。

MB 资源管理器具备丰富的管理特色，为管理和维护人员

提供完全的功能，可以启动和停止代理、执行组和消息流，可以创建、修改、删除本地执行组，可以部署消息流到多个执行组，可以启动/停止跟踪节点，设置代理跟踪级别，等等。

同时，MB 资源管理器提供图形化的监控功能，允许维护和管理人员清楚的了解当前系统的运行状况，以图形化的指标展示系统资源的使用情况，包括 CPU、I/O 等的度量值，消息流、处理节点等的处理性能，并可根据代理、执行组、消息流随意选择监控统计范围。允许用户将日志数据以 CVS/Excel 等可读格式导出到文件，为系统的后续优化提供分析依据。

WebSphere Message Broker 的处理节点可以通过配置监控标签在运行时产生监控事件，与 WebSphere Business Monitor 的结合，可以监控和分析各种 KPI 指标，实现企业业务监控、业务分析以及业务审计等多种场景。

四、总结

ESB 的建设，是企业建立 SOA 体系的重要环节，为企业各应用系统构建了一个统一、灵活、完整的整合平台。它很好的解决了企业内外新老应用系统之间互操作和信息交互的问题，最大限度的减少了应用系统互联所面临的复杂性，为企业开发后续应用及信息共享、数据整合打下了良好的基础，能够使企业适应未来在业务功能、处理流程、信息技术以及与外部异构系统通信等方面的发展需求。这大大促进企业业务流程优化、提高了企业运行效率。

WebSphere Message Broker 作为 IBM 的高级 ESB 产品，在银行、保险、电信、医疗卫生、交通运输、能源、烟草等多个领域都得到了广泛的应用。可以肯定的说，WebSphere Message Broker 在企业 ESB 建设的路上将发挥越来越大的作用。■

IBM WESB 概述及其 V6.2 的功能简介

寇卫东、范迪 / 文



范迪 现任 IBM 公司软件部高级技术顾问。多年来一直从事 WebSphere 中间件及其相关产品的技术支持和软件架构工作。通过 IBM SOA 架构师认证，是国内最早接触和从事 SOA 相关技术支持和推广工作的技术专家之一。在 SOA、BPM、EAI 等领域的设计和开发方面有着丰富的经验，并致力于将其在 IBM 的众多合作伙伴及企业客户中推广普及。曾参与过中国石油、中国网通、中国电信、中国移动、人民银行、交通银行、国家发改委、中国国际航空公司、中国远洋进出口公司、国家税务总局、华为公司等众多大型项目的技术支持和方案设计工作。

IBM WebSphere Enterprise Service Bus (以下简称 Websphere ESB 或者 WESB) 是一个灵活的，用于集成服务和应用程序之间连接的基础架构，它的设计目标就是用于 SOA 的开发。WebSphere ESB 位于 SOA 的核心部分，通过减少接口的数量和复杂性来方便 SOA 实施。它使客户集中精力于其核心业务而不是 IT。本文将对 Websphere ESB 作一个概述性的介绍，并且着重讨论 WebSphere ESB V6.2 的功能特性及其优势。

一、IBM WESB 概述

WebSphere ESB 提供了一个基于标准的集成平台，用于方便服务之间的连接。通过 Web Services 的连接，Java 消息服务 (JMS) 和基于服务的集成，客户

可以通过使用一个企业总线来增强灵活性，最小化故障率，来处理集成的逻辑。WESB 允许在业务实践和端点之间的智能互操作。

1. WESB 技术方案

面向服务的体系架构 (Service Oriented Architecture, 简称 SOA) 提供了动态开发和更改集成应用的能力，使客户能够集成现有应用和新应用，透明地在一起工作。

WebSphere ESB 能够通过减少接口的数量和复杂程度来增强 SOA 的能力，因此，客户能够更加关注其核心业务，而不是其 IT 系统。

WebSphere ESB 具备如下特性能够在集成环境中最大化灵活性：

- 在服务间路由消息



- 在服务请求者和服务之间转变传输协议
- 在服务请求者和服务之间转换消息格式
- 处理来自不同来源的业务事件
- IBM WebSphere Integration Developer提供了一个交互式的、可视的开发环境，能够降低对编程技能的要求，IBM WebSphere ESB通过与该开发环境的无缝集成，能够提高时间价值

2. WESB 的技术优势

WebSphere ESB提供了一个企业服务总线，通过基于标准的接口连接应用程序以加速客户的 SOA 进程。WESB 产品的关键优势包括：

- 提供了 Web 服务的连接方式，JMS 消息机制和面向服务的集成

WebSphere ESB 提供了智能的集成，通过面向服务的接口来连接客户的资产。

- 使用简单

WebSphere 集成开发工具是一个可以应用于 WebSphere ESB 的基于 Eclipse 的有力开发工具。这个工具使用简单，仅仅需要最小的编程技巧。使用这个工具，客户并不需要了解 JAVA 以建立集成解决方案 - 它是集成的，互操作的并且提供了可视化的开发体验。当客户使用 WebSphere ESB 来开发其集成解决方案时，WebSphere 集成开发工具易于开发，建立，测试，部署和管理服务组件。WebSphere 集成开发工具同时是用 WebSphere Process Server 来配置和建立解决方案的开发工具，它提供了一种直接的方式来提高客户的集成能力，包括流程级别的集成。

- 节省时间成本

通过使用适配器支持数以百计的独

立软件开发商的解决方案，比如 SAP，Siebel，PeopleSoft，JD Edwards 和 Oracle。这就允许客户通过使用预先设置的中间方式，包括 XML 的转换，基于内容的路由和消息记录方式节约时间和开发成本。

- 与 WebSphere 平台的无缝集成

WebSphere ESB 是基于 WebSphere Application Server 的，WAS 是一个世界级的 J2EE 基础平台，提供了业界领先的可用性，可扩展性和高性能。通过使用构建于 WebSphere ESB 的 WebSphere Process Server，IBM 同时提供了

解决更复杂的业务问题的能力。所以当业务需要的时候，客户能够轻松地进行扩展来利用 WebSphere Process Server。更进一步，WebSphere ESB 能够与 IBM Tivoli

安全和系统管理等能力紧密集成进行更全面的并且监控在客户业务之中的信息流。

3. WESB 的特色总结

WebSphere ESB 是支持 SOA 的核心（连通性）的主要产品之一，通过减少接口的数量和复杂性来加速其实施，因此可以帮助客户最大化 SOA 的灵活性。WebSphere ESB 的特色包括：

- 提供了用来管理 Web 服务连接，互操作，服务托管和服务中间环境的完整解决方案
- 允许客户安全可靠地在应用程序之间共享信息，这种方法从开发和维护

的角度来看都是节约成本的。

- 帮助客户轻松地增加新的服务或者改变现有服务而不影响当前的 IT 环境。
- 通过卸载消息处理和将复杂的中间任务独立到一个专门的集成服务器中以增强系统的性能和可用性

二、WESB V6.2 的功能简介

1. WESB 与面向服务架构

IBM 面向服务的体系结构 (Service-Oriented Architecture, SOA) 是个组件模型，能够相互关联各种被称为服务的应用的不同功能单位，相互关

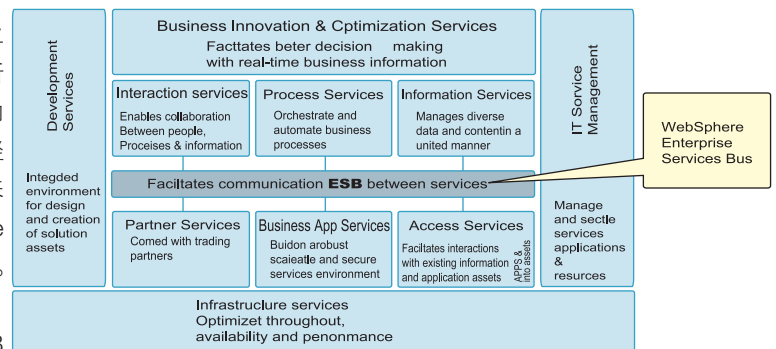


图 1: IBM SOA 的参考架构

联是通过这些服务间的明确定义的接口和契约进行的。

服务接口的定义是中立的，独立于服务实现的硬件平台、操作系统和编程语言，所以构建于不同系统上的服务能够以一种统一通用的方式进行交互。

IBM 面向服务的体系结构的参考架构概述了广泛的、企业范围的 SOA 解决方案所需要的关键能力。客户可以基于其所希望的来实现这些能力，从而可以根据时间的推进和新需求的出现，方便地添加功能和项目级别的解决方案。

如图 1 所示，IBM SOA 参考架构的核心是企业服务总线 (Enterprise Service

Bus, ESB), 它提供了利用跨整个架构实现的服务所需要的所有互相连接的能力, 传输服务、事件服务和调解服务都是通过 ESB 提供的。ESB 为整合应用和服务提供了一个灵活的连接基础架构, 从而最大化了 SOA 的灵活性。

WebSphere ESB 就是企业服务总线的一种主要的实现产品。

2. WESB V6.2 的功能特性

IBM WebSphere Enterprise Service Bus (ESB) V6.2 是一个灵活的连接基础架构, 旨在帮助客户将其应用和服务整合到面向服务的体系架构 (SOA) 中。

WebSphere ESB 是构建于 WebSphere 平台的消息中枢基础之上的服务总线。WebSphere ESB 能够:

- 提供 Web service 连接, MQ 和 MQ/JMS 消息, 以及使用 WebSphere Adapter 的面向服务的整合;
- 针对改进的 IT 操作的与业界领先的 WebSphere 平台进行无缝整合;
- 提供基于 WebSphere Application Server 网络部署版的适用于关键业务的服务质量;
- 容易地为服务编排和商业流程管理 (BPM) 扩展到 WebSphere Process Server;
- 提供服务调解和服务托管的整合方案。

WebSphere Integration Developer 是针对 WebSphere ESB 的开发工具, 提供仅需最少编程技能的集成的、交互式的和可视化的开发体验。

WebSphere Integration Developer 通过对调解功能进行可视化编辑简化了开发工作, 从而支持智能的消息路由、扩充和转换。这个工具可以帮助简化描述服务和定义接口的工作。

WebSphere ESB 推动了服务交互的动态重新配置以满足变化的业务需求。广泛的 WebSphere Adapter 支持提供了对数以百计的 ISV 解决方案的连接。预置的调解功能能够节约时间和开发成本。

具体来说, WebSphere ESB V6.2 包括如下功能:

- 支持多种消息协议, 包括 JMS 1.1 (通过 WebSphere Application Server 和 WebSphere MQ 支持)、WebSphere MQ、TCP/IP、SSL、HTTP(S), 及为最佳灵活性和提高资产复用的多点广播;
- 帮助满足需求的广泛的交互模式 (请求/响应 request/reply, 点对点 point-to-point, 发布/订阅 publish/subscribe, 和多点广播 multicast);
- 高级的 Web 服务, 包括 SOAP/HTTP、SOAP/JMS、WSDL 1.1、Web Services Gateway。WebSphere ESB 支持 WS-* 标准, 包括 WS-Security、WS-Atomic Transactions, 并且包含一个 UDDI 3.0 Registry, 能够用来发布和管理服务的端元数据;
- 支持在 WebSphere Service Registry and Repository 中查询服务端点信息, 使用动态服务端点选择和绑定, 不需重新开发和部署。
- 支持 WSDL/XSD, 扩展跨企业标准模式的互操作能力;
- 内置 WebSphere MQ 绑定支持, 使客户能够更方便快捷地整合 WebSphere MQ 资产; 并提供丰富的 MQ 消息头设置原件, 方便存取 MQ 消息头参数。
- 支持子消息流的定义, 使消息流更加清晰。
- 提供 ServiceGateway 功能, 为不同类型的服务统一了入口。
- 广泛支持 WebSphere Adapter 解

决方案, 包括基于 Java 2 Platform, Enterprise Edition (J2EE) Connector architecture (JCA) 技术的适配器;

- 支持 C/C++ 和 Microsoft? .NET 的消息客户机, 使得非 Java 应用可以连接 ESB;
- 整合发布 / 订阅 Web 服务、RESTful 客户机 (Ajax)、传统 Web 服务, 和第三方消息系统;
- 与 WebSphere Transformation Extender 整合, 为复杂的数据格式和企业标准提供统一转换。

WebSphere ESB V6.2 构建于标准的 WebSphere Application Server (WAS) Network Deployment V6.1 之上, 因此, 继承了所有 WAS 平台的能力, 包括集群 (clustering)、失效转移 (fail-over)、可扩展性 (scalability) 和安全性 (security)。

此外, 基于 WebSphere Application Server V6.1, WebSphere ESB V6.2 能够提供与其它 WebSphere 家族产品的平滑的协同工作能力, 包括 IBM WebSphere MQ 和 IBM WebSphere Message Broker。还能够使用 IBM WebSphere Adapter 解决方案来利用现有应用资产, 并且捕获分散的业务事件。

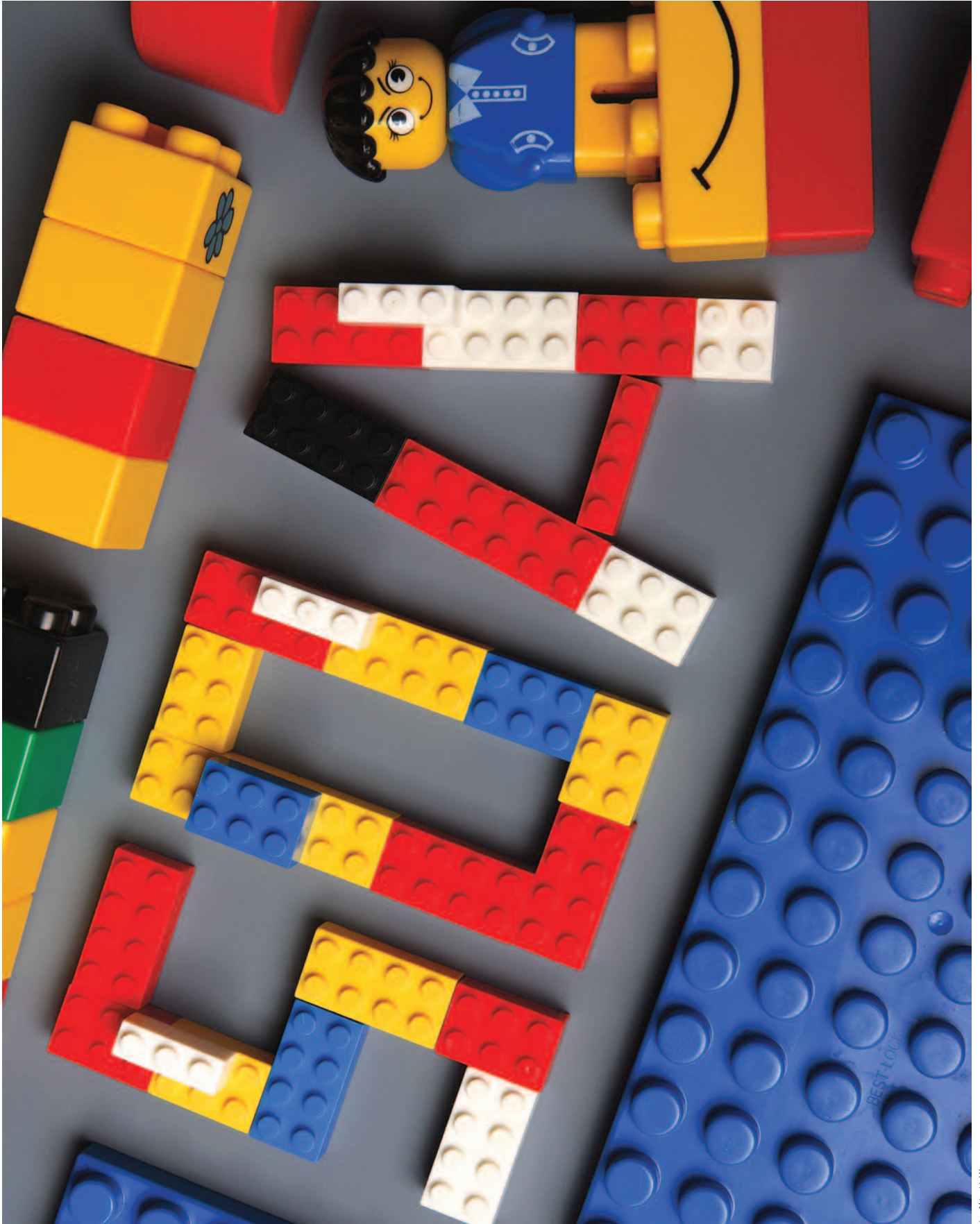
附录: IBM WESB 产品支持信息

1. 全球化支持

IBM WebSphere ESB 提供对所有地区的基本支持, 提供下列语言的翻译: 葡萄牙语、捷克语、法语、德语、匈牙利语、意大利语、日语、朝鲜语、波兰语、俄语、西班牙语、简体中文 (GB18030)、繁体中文。

2. 软硬件支持

IBM WebSphere ESB 支持下列操作系统: AIX、HP-UX、i5/OS、Linux、Linux on Intel、Linux on System z、Solaris、Microsoft Windows、z/OS。■



摄影/杞人

中软国际基于 SOA 体系结构的卷烟营销总体解决方案

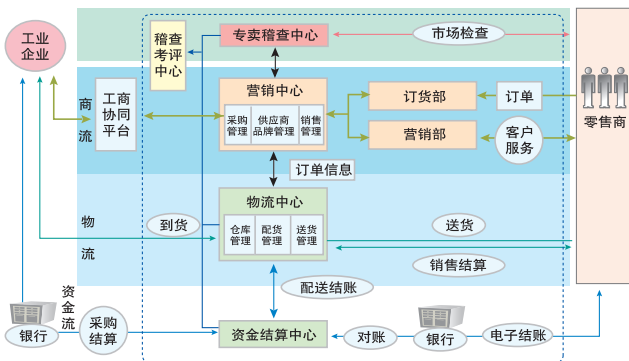
中软国际 / 文

方案概述

根据多年来烟草行业信息化的成功经验,中软国际提出了基于面向服务的体系结构SOA的卷烟营销总体解决方案,该方案按照SOA的组件开发、服务封装等的理念,通过对卷烟营销的管理过程和业务流程进行分析,把卷烟营销归结为品牌策划、资源计划、仓储管理、订单采集、配送管理、客户服务、客户测评等既相互独立又相互联系的7个业务实体(也可以称为业务服务),并通过企业服务总线(ESB)提供的数据共享机制和统一的数据接口服务实现各个业务实体的集成。

方案详情

一、组织架构和业务职能



烟草商业企业的业务流程是由商流、物流、资金流和信息流四个相互关联的子流程共同构成的,涵盖品牌规划、资源计划、卷烟存储、订单采集、卷烟配送和客户服务,是供应链上环环相扣的业务环节和流程。

商流: 涵盖了卷烟销售网络从了解需求、引导消费、订单采集、卷烟采购到卷烟销售的全过程,通过客户服务和品牌培育实现其顺利、高效运行。

物流: 物流中心的主要管理是仓储管理、配货管理、送货管理。与烟厂供应商进行到货协商和到货确认,对零售户进行

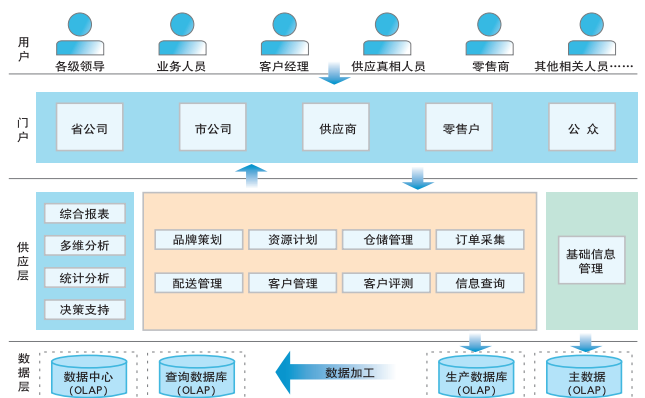
上门送货和销售结算,以及卷烟的在库管理。

资金流: 统一资金结算中心,与供应商进行采购结算,与零售户进行实时和批量的电子销售结算。

信息流: 辅助企业管理整体运作,指示物流的方向,监督着资金流的流量,为卷烟销售网络运行提供统一信息平台,提高运作效率。商流、物流、资金流通过信息流的融合,形成完整的环路。

烟草商业企业通过统一订单、统一结算、统一配送、统一信息平台,将卷烟营销、物流配送、市场管理有机结合起来,缩短供应链,增强价值链管理,提高管理效率,实现企业规模经济和核心竞争力的提升。

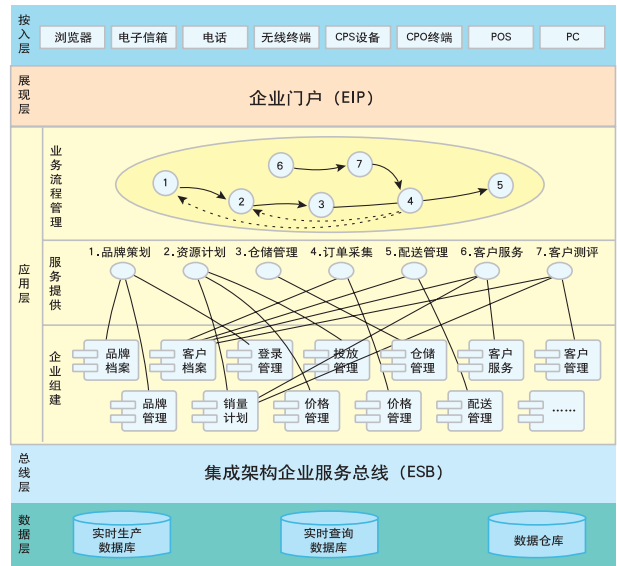
二、应用架构



应用结构分为数据层、应用层、门户层、用户层。

数据层: 包括数据中心、大集业务数据、主数据。对应用层提供数据。

应用层: 包括基础信息管理、综合报表、多维分析、统计分析、决策支持、供应商管理、品牌管理、价格管理、需求预测、销售计划、采购计划、采购交易、货源供应、订单采集、仓储管理、配送管理、资金结算、客户管理、工商协同等。基



基础信息管理是其它应用功能的基础，存放在主数据内。数据中心数据来源是业务数据和主数据，是综合报表、多维分析、统计分析、决策支持的数据支撑。

门户层：提供省公司、地市公司、工业企业、零售户、公众等用户统一的门户。

三、技术架构

该方案按照 SOA 的组件开发、服务封装等的理念进行设计，有以下几个特点：

(1) 统一访问门户

通过引进统一门户平台，实现各业务系统的统一门户和单点登录，要求系统必须具有人性化设计，能够根据不同员工的需要进行功能组合，将合适的功能放在合适的位置给合适的员工访问，实现真正的以人为本的管理。

(2) 统一信息传输渠道和信息交换方式

实现互连互通的重要手段，就是统一建设信息传输与交换系统。建设此系统的目的，就是按照统一的标准，在省公司和下级各单位之间进行电子公文、业务信息的传输及交换，并保证信息在传递过程中的安全性和有效性。

(3) 统一信息配置管理

配置和管理工具主要针对用户级的系统管理员使用。平台

提供统一的配置信息，系统信息集中管理维护，节约维护成本。

(4) 统一解决应用安全

应用集成平台作为信息流的单一出入口，用户获取信息的门户，需要采取统一的登录方式和认证方式确定用户的身份、对用户的访问进行控制、保证传输的数据不被泄漏和篡改。

(5) 统一标准与规范

编制及统一的开发和集成规范，该开发和集成规范不能成为应用系统开发商的羁绊，而是为了快速、优质地开发设计出更可维护、更可集成的应用系统。

四、客户价值

符合《数字烟草发展纲要》、《烟草行业信息化建设统一技术平台要求》等行业规范。

系统采用 B/S 多层结构设计，以 J2EE 技术架构构建应用平台，适应多种不同的操作系统。

系统采用面向服务的 SAO 架构构建，符合面向服务的集成、灵活地绑定服务等理念。

统一访问门户：通过统一门户平台，实现各业务系统的统一门户和单点登录。

统一信息配置管理：配置和管理工具主要针对用户级的系统管理员使用。■

北京移动 BOSS 网管项目案例介绍

北京神州泰岳软件股份有限公司 / 文

北京移动通信有限责任公司是中国移动（香港）有限公司全资间接拥有的外商独资企业，主要经营移动电话通信业务（包括语音、数据、多媒体等），IP 电话及互联网接入服务；具有移动通信、IP 电话和互联网网络设计、投资和建设资格。

北京移动拥有超过 1000 多万客户和强大网络覆盖，是世界上网络规模最大、网络结构最复杂的移动通信城域网之一。目前，北京移动通信有限责任公司的通信网络实现了与全国各省、市、自治区的国内漫游及世界上五大洲的 100 多个国家和地区的漫游。

客户面临的问题和挑战

当今通信市场正由“传统的以通信网和市场为中心”的竞争转变为“以客户为中心的服务质量”的竞争。在中国移动通信集团公司的统一规划和领导下，各省、自治区、直辖市移动通信公司业务支撑系统的集中化改造工作已基本完成。为在新形势下，为充分利用现有资源、提高现有的维护效率，移动集团公司提出了运行维护工作的“集中监视，集中维护，集中管理”的三集中思路。围绕三集中思路的统筹规划已得到广泛的应用和实施。面对改造完成后系统的复杂多样性和不断扩充的业务需求，如何保障业务支撑系统的正常稳定运行，确保并逐步提升北京移动的服务质量？这就迫切要求北京移动建立一个能够对业务支撑系统进行集中监控、集中维护、集中管理的网管系统。

BOSS 网管系统应该不仅能够及时监测并发现业务支撑网中存在的、潜在的各类问题，保证系统的稳定运行和业务的正常开展；还要能够对服务和运行维护工作进行规范化、流程化管理。通过发现、总结和挖掘所存在问题，不断明确管理重点并优化管理流程，从而加强服务和运行维护管理能力、提高服务和运行维护工作效率、改善服务和运行维护工作质量，进而保证北京移动业务支撑网全网服务质量和运行维护水平的可持续性提升。

解决方案

方案设计

根据北京移动 BOSS 系统管理的实际情况和《中国移动业务支撑网网管系统规范》的要求，我们认为本期 BOSS 网管项目主要从如下三个方面进行建设。这三个方面管理对象不同，功能也相对独立；但是在技术实现上又有许多共同和相通之处，存在着紧密的联系。

BOSS 系统平台监控：对线路、网络、系统、数据库、存储、备份等 IT 基础设施的配置、故障和性能进行监控管理。这是保障整个业务正常运作的基础和前提。

BOSS 业务监控：从业务角度对采集、计费、结算、帐务、营业、拨测、客服等业务系统的配置、故障和性能进行管理。这可以最直接地反映业务运行情况，针对业务系统的监控将是本期 BOSS 网管监控平台部分的核心。

BOSS 系统运维管理：从事件管理、问题管理、配置管理、变更管理、统计考核等方面对各个运行维护支持岗位进行规范管理。这是充分调动和发挥人的主观能动性和积极性的根本方式，必将对服务和运行维护工作水平的提升起到关键作用。服务和运行维护管理是本期建设的又一重点。

监控管理平台的管理对象主要是 IT 系统，而服务管理平台主要面向运维支持人员，两者之间通过事件自动报障建单、配置自动同步等接口紧密结合起来，形成统一的整体。

北京移动 BOSS 网管系统由监控管理平台、服务管理平台、内/外部接口、用户和系统自管理和统一统计报表管理等几部分组成。

服务管理平台：提供了流程支撑引擎和基于其上的事件管理、问题管理、配置管理和变更管理，并提供日常运维和知识共享库功能。

监控管理平台：用于实现对 BOSS 系统支撑设备（主机、网络、数据库、中间件、存储、备份）以及业务应用（采集、计

费、营帐、结算、客服、拨测以及一级 BOSS) 的监控管理。

数据分析: 将监控管理平台的各种数据(配置信息、历史性能数据、告警信息)和服务管理平台的各种数据(各类工单、处理过程数据等)进行统一的处理和分析,并通过统一的界面组织和展示。

统一用户登录与认证: 监控管理平台和服务管理平台构成的 BOSS 网管系统,通过内置的或外部已有的 LDAP 服务,实现用户登录和认证的统一管理。

方案特点

1、面向企业核心业务的全面运维: 针对北京移动的核心计费系统,实现了全方位的深入实时监控和运维管理,将业务和 IT 运维相结合,促进了业务发展;

2、良好的统一性: 实现了统一的 CMDB,统一的用户认证和单点登录,统一的数据报表统一分析,统一的平台和应用监控;

3、充分利用现有多种主流 IT 管理产品的优势,并实现有



效集成,形成了实用、高效、综合的统一管理平台;

4、对核心业务关键应用指标(KBI)的深入监控和业务多个分支环节的连续跟踪,保证应用系统监控管理的细化和综合的统一。

应用效果与收益

系统实现对遍布北京全市各营业厅以及美惠、望京、菜市口的 BOSS 网络和客服网络的管理;实现了对 150 台各类服务器、30 多套数据库(包含 Oracle、DB2、SQL Server)、以及对 Siebel、Exchange、WAS、MQ 等商业通用软件的管理。系统完全遵循 BOSS 网管规范,实现了各种平台类监控 KPI 指标的监控,建设了完善的 CMDB 数据库,实现了对 10 个 IT 系统的统一监控管理,整个网管系统能够完全匹配北京移动现有的运维体制。

北京移动计费业务中心的系统维护中心、业务支撑中心、信息技术中心、综合部等三级部门通过该项目的建设,成功实现了有效的核心业务系统监控和运维,取得了良好的经济效益和社会效益。■

中科软银保通系统解决方案

中科软科技股份有限公司 / 文

一、银保通系统概述

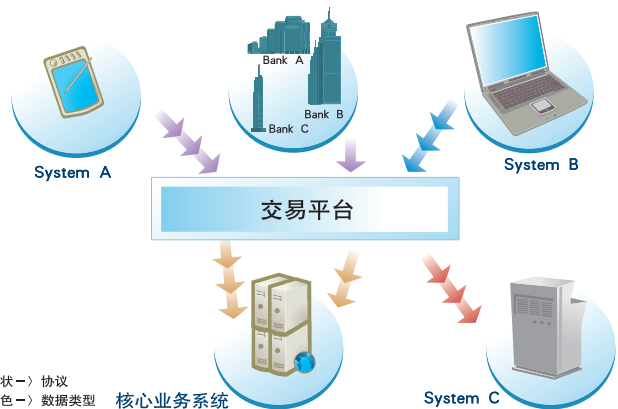
随着市场的发展,保险公司为了适应激烈的竞争,需要利用更多的方式、渠道来拓展其业务。银行、邮储等金融机构,因其具备良好的信誉,庞大的网点机构体系以及广泛稳定的客户源,成为保险公司首选的合作伙伴。于是,与银行方进行实时联机交易的银保通系统应运而生,并在保险公司和银行之间迅速推广。为了适应市场的需求和业务发展的需要,我们需要建立一套达到如下目标的银保通系统:

与保险公司的业务系统有效的整合

支持多种交易流程

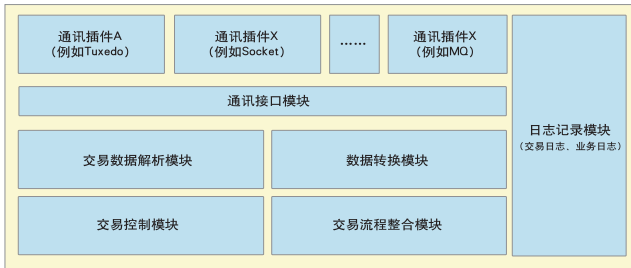
支持多种交易数据标准和格式,包括保监会下发的《保险数据交换标准》。

支持多种通讯协议
保证系统的可拓展性



二、银保通系统软件架构

整个银保通系统按照功能划分，包括通讯模块，数据处理模块，交易处理模块和日志记录模块。如下图所示：



通讯接口模块和通讯插件：

由于系统面对的通讯方式多种多样,为了保证系统的可扩展性,通讯部分的功能划分为通用的通讯接口模块和针对不同的通讯方式、通讯对象而开发的通讯插件。通讯 插件负责接受交易方的信息或向交易方发送信息。通讯接口模块负责提供统一的接口供通讯插件调用,接受通讯插件获得的信息或向通讯插件传递信息。

交易数据解析模块和数据转换模块：

该部分是银保通系统的“翻译员”，负责屏蔽银行和各个银保通系统间的数据差异。交易数据解析模块负责按照配置,将不同的交易数据按照一定的含义进行拆分(组 装)。数据转换模块负责将原始的交易数据按照业务含义的配置,转换成银保通系统能够识别的业务数据,或进行反向的操作。

交易控制模块和交易流程整合模块：

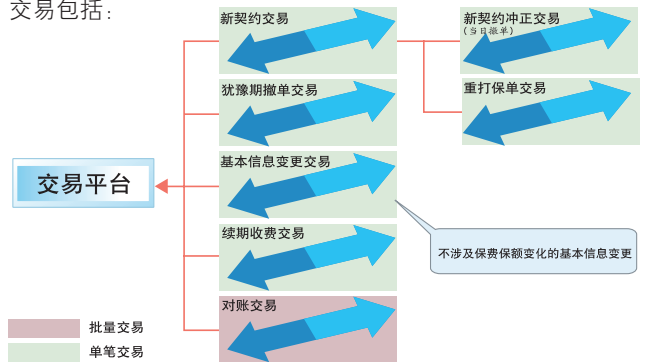
交易控制模块负责控制交易数据的流向和选择处理模块。而交易流程整合功能负责屏蔽各个银行和保险公司间的交易流程的差异。能够与不同类型或厂商的核心业务系统对接。

日志记录模块：

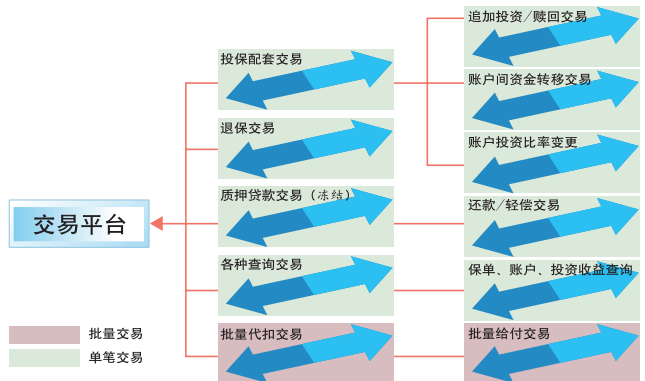
记录交易日志、业务日志和各种系统运行日志。

三、银保通系统交易功能

银保通系统作为一个银行和保险公司的业务交易系统,各项交易功能是其业务能力的核心。目前银保通系统具备的常用交易包括：

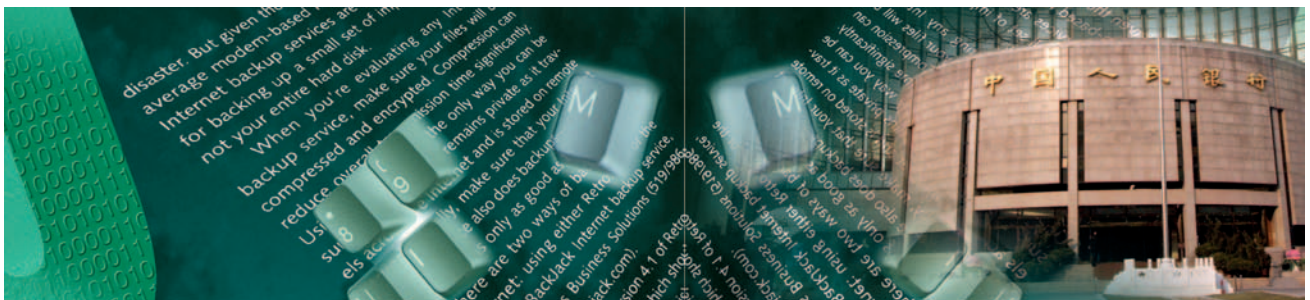


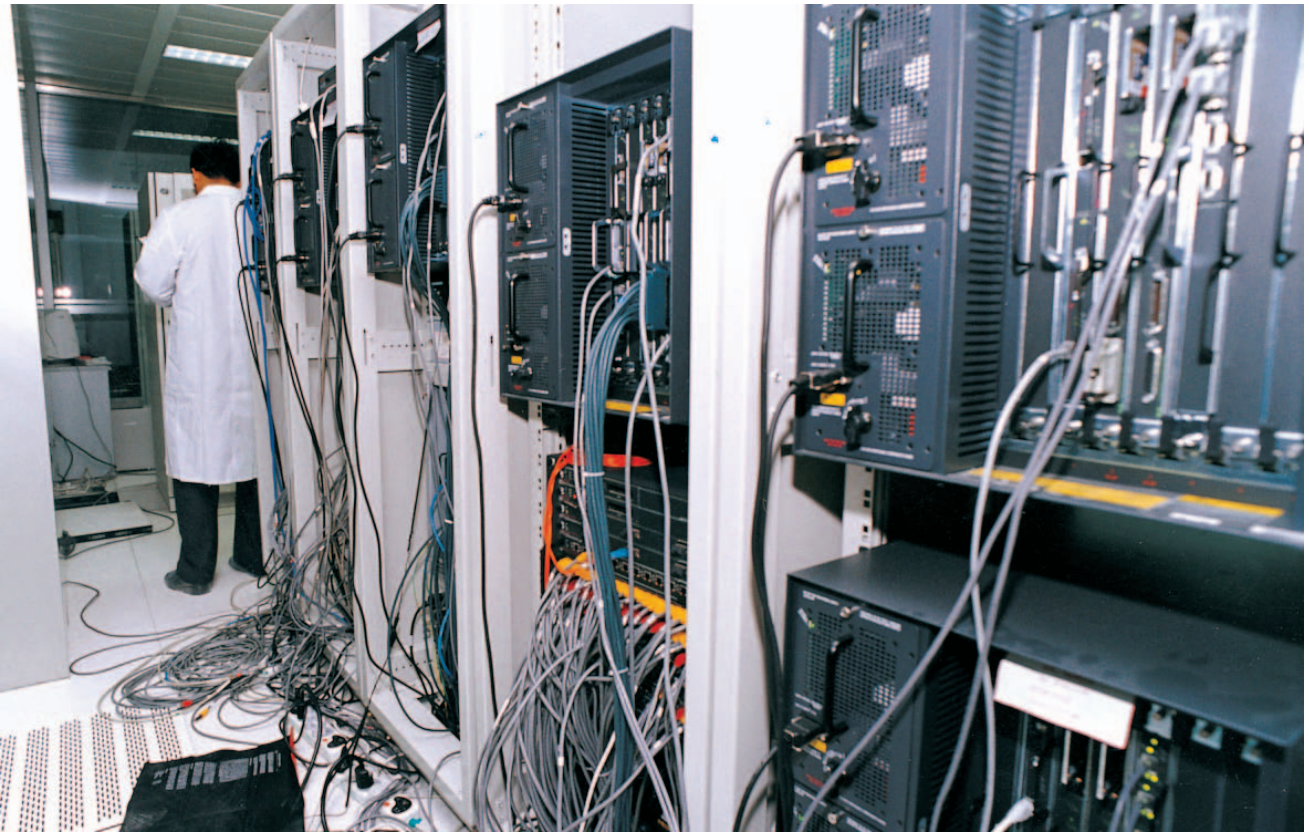
目前银保通系统具备的其他交易包括：



四、银保通系统的外围功能说明

为了支持银保通系统的稳定运行,方便用户监控系统运行状况以及业务状况,银保通系统除了拥有强大的交易功能外,还拥有丰富的外围功能,包括:对帐情况管理、结算银行管理、交易信息的查询及统计、交易权限管理功能、交易报文解析描述、交易流程描述等。■





中科软银行信贷风险管理解决方案

中科软科技股份有限公司 / 文

Risk@Banking 介绍

风险是指在某一特定环境下,在某一特定时间段内,某种损失发生的可能性。风险既是一种损害,也是一种机会。风险管理,是指如何在一个有风险的环境里把风险减至最低的管理过程。花旗银行前董事长瑞斯顿曾说过:“所谓银行,就是基于风险处理能力而盈利的组织。”这句话一语道破了风险管理能力就是银行的核心能力。银行的经营目的在本质上就是在可控的风险内获得最大的利润。信贷风险是商业银行贷款的信用风险,是商业银行面临的最古老的金融风险。银行自诞生的一日起,就在孜孜寻求规避信贷风险的方法和手段,以最大限度地控制信贷风险。银行进行信贷风险管理,不仅可以有效地提高银行的资产质量,改善银行资本充足率偏低状况,维护国家金融系统的安全和稳定,而且可以提升银行的盈利能力,增强银行的竞争力;此外,做好信贷风险管理有助于降低银行的风险资产含量,改进银行在同业中的形象,提升银行在市场上的竞争地位。

Risk@Banking 以风险控制、决策为核心,运用先进的商

业智能技术和数据挖掘技术,从商业风险、信用风险、法律风险、决策性风险和操作性风险等多个方面,将信贷风险管理和银行的盈利情况进行联动分析,使银行的管理者能够借助智能分析手段找出信贷风险产生的规律,从而提高银行的风险管理水平,达到有效地预测风险、防范风险、提高盈利的目的。

Risk@Banking 是一套具有决策支持分析和能力的系统,主要面向银行决策部门及信贷管理部门。主要数据来源为银行综合业务系统、OA系统、外汇系统、信贷管理系统和部分手工数据。

Risk@Banking 功能



该系统由以下四个功能子系统构成：客户信息管理子系统；信用评级子系统；信用风险识别、评估子系统；信用风险控制、决策子系统。

（一）、客户信息管理子系统

客户信息管理数据库可以提供强大的、多样性的数据支持。这些数据主要从内部业务系统数据库、外部业务系统数据库和 Excel 文档中自动导入，少量的基础资料由业务人员手工录入。该子系统为信用评级子系统提供数据，并奠定风险分析的基础。

（二）、信用评级子系统

信用评级子系统能够建立科学的、实用的、精确的和权威的信用评级模型。该子系统对原始客户信息数据进行适当处理，给出客户的信用评级表，输出包括信用评估报告、信用评估结果表单等信息。该子系统有着灵活的权限管理，智能的评估报告管理和输出，批量数据导入，灵活的数据查询与统计分析等功能。其先进的规则引擎帮助您灵活定义信用评级模型，使得信用评级模型可以在用户的使用过程中随着需求的不同而不断变化，不断调整。信用评级模型建立过程，如右图所示：



（三）、信用风险识别、评估子系统

主要依据客户的信用等级，结合风险分析的一种古老方法——专家方法，同时考虑国家金融政策以及国家经济形势，市场环境等，建立完善的风险模型库和量化分析器，可以根据用户的需要不断新增、调整风险模型。根据风险模型对风险进行识别和度量，对风险管理全过程进行全面监控和控制，做出风险管理评估报告，为风险控制决策子系统提供支持。

（四）、信用风险控制、决策子系统

主要基于对客户数据的分析和数据挖掘，并依据评估子系统的结果，运用一定的数学模型求得预测值。在实际运用中，预测值与已发生的实际值进行比较，或与经验估计值相比较，有出入较大，难以真正实用化运行，因此，仅靠现有的历史数据和数据挖掘数学模型，难以提高风险预测的精度，需要有经验的预测人员进行人工干预，按一定权值进行综合预测，才能比

较客观地反映实际，同时，确立、行使信用风险管理原则，制定风险指标以及避险策略等职能，对风险进行再分配或转移。

特点

- * 同时从外部风险和内部风险两个层面全面进行监测和管理
- * 将信贷风险管理和银行的盈利情况进行联动分析
- * 全面结合了客户关系管理、电子商务、企业门户，形成了独特的应用模式

- * 个性化服务完全满足用户多变的需求

- * 强大的 OLAP 分析能力，帮助决策者多方面分析问题

Risk@Banking 效果

- * 增加客户信息量，增强信息处理能力

- * 规范银行信贷业务管理

- * 防范信贷风险，优化信贷资产结构

- * 加强信贷机构和人员的管理

运行环境

- * JDK 版本要求：JDK 5.0 以上

- * 数据库产品：Oracle DB /DB2 /SQL Server /MySQL

- * 中间件产品：IBM Websphere/Jboss/Tomcat 5. ■





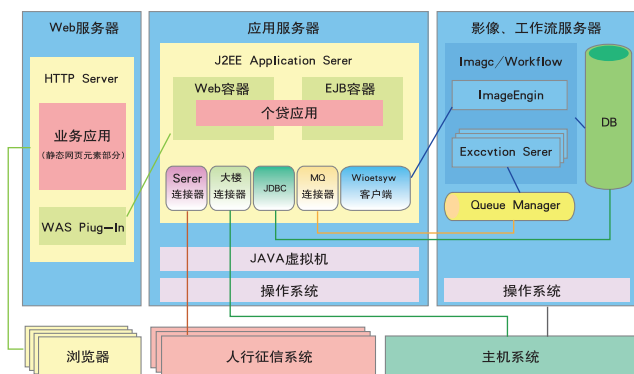
上海华腾个人信贷管理解决方案

上海华腾 / 文

个人信贷管理是银行业务系统中的一个重要业务系统。上海华腾软件系统所开发的个贷（信贷）管理系统，由贷款管理和贷款分析两部分组成；按最新核算制度的规定，实现个贷管理各项业务功能。并能充分发挥银行核心业务系统数据集中，信息共享的优势，实现以客户为中心，提供多渠道服务，同时满足银行内部管理和风险控制需要。

系统结构图

系统架构于 J2EE 技术结构体系之上：



系统技术特点

参数化设计：系统采用了大量的参数化设置，从贷款产品、风险控制、系统运行 3 个维度描述整个个人贷款管理系统。参数的形式包括存放在数据库中的数据 and 以 xml 配置的文件。

层次化设计：层次化设计最主要的目的是使个贷系统适应系统级的移植或修改。整个系统从纵向划分为 5 个层次，每一层的软件都向其上一层软件提供标准访问接口。

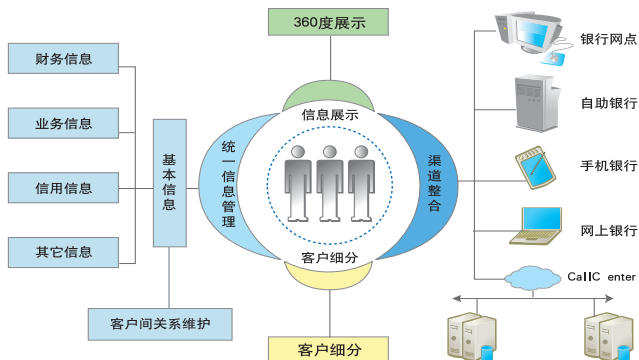
面向服务设计：与传统的端到端的企业应用系统不同，本系统采用 SOA 架构，SOA 提供了一系列的“服务”模块，这些服务模块具有定义良好的输入/输出接口以及功能完善的处理模块。传统的端到端的系统可以通过使用这些服务，方便的将这些服务集成在一起，使所构建的各种服务模块不再仅仅为某一个系统服务，而是可为整个企业内部大多数的系统所用。

组件化设计：高效、快速的系统开发是华腾个贷系统所追求的设计目标之一。为达到这一目标，我们采用了组件化设计的模式。整个系统具有高度的灵活性与可拓展性。

流程化设计：在个贷系统中我们采用工作流产品把业务处理的功能完全从业务流程中剥离出来，形成完全独立的功能模块。这种通用性使得数据流在各个模块间高效运行成为可能，同时各种业务的高度独立，使得业务具备了可插拔性、可扩展性和可维护性。

系统业务特点

以客户为中心：系统以客户为中心进行设计和部署，经营管理体制也都是围绕为客户提供最优质的服务来安排和设计的。



面向市场与营销：系统建立了一整套符合中国国情的市场与营销的管理体系。整个体系基于“以客户为中心”的基础上，一共由 6 部分组成：客户经理制、产品细分、价格管理、市场营销活动、主动销售、盈利分析。

良好的数据展现方式：系统的设计具有多种数据展现方式，系统不仅可以提供二维报表和复杂报表功能，并还可以把报表存储成为 Html、Excel、Pdf 等各式，而且还能提供层次化的灵活分析和丰富的数据展现功能。

强化风险控制：风险管理功能在系统中主要由贷款预警、操作风险、信用风险、额度与权限控制、贷款审批、流程审计、五级清分、资产保全 8 个模块构成。■

WebSphere 应用服务器 Hypervisor 产品简介

黄怡文 / 文

WebSphere 应用服务器 Hypervisor 版本是 IBM WebSphere 应用服务器的虚拟映像版本。基于时下越来越热门的虚拟化技术，WebSphere 应用服务器 Hypervisor 实现了对 WebSphere 应用服务器环境的快速搭建和部署的需求，满足了客户希望在虚拟环境下运行和使用 WebSphere 应用服务器的心愿。

一般而言，传统的 WebSphere 应用服务器环境的搭建过程通常包括了(1)操作系统的安装与配置，(2)WebSphere 应用服务器的安装，(3)WebSphere 应用服务器概要文件的创建，以及(3) WebSphere 拓扑搭建和配置。相对一个大环境的建立而言，第(1)和第(2)个步骤往往是漫长而单调的重复劳动。

虚拟映像的一个突出优点是环境系统的预先安装和配置，比如说，WebSphere 应用服务器 Hypervisor 的虚拟映像中已经事先安装了操作系统与 WebSphere 应用服务器。因此，在虚拟环境中搭建 WebSphere 应用服务器节省了从操作系统到 WebSphere 应用服务器的安装过程，取而代之的是对虚拟映像的激活过程。具体来说，在激活过程中，只需要提供操作系统的网络和用户密码信息，选择 WebSphere 应用服务器概要文件种类，就可以方便快捷地建立起需要的 WebSphere 应用服务器环境。这大大缩短了搭建 WebSphere 应用服务器环境中的重复劳动和漫长时间，

同时也避免了安装和配置中的人为错误操作的可能。

作为 IBM 自己制作和发布的 WebSphere 应用服务器虚拟映像，它还拥有这些方便之处：

- 操作系统和 WebSphere 应用服务器的自动调优

WebSphere 应用服务器 Hypervisor 的一个主要特性是在激活过程中对操作系统和 WebSphere 应用服务器的配置的自动调优。例如根据 virtual machine 的大小，对 JVM heap 以及 threads pools 进行自动化的调优。这些基于十多年 WebSphere 应用服务器的发展与使用基础上经验总结的自动配置调优免去了用户的手动操作，确保了激活后的虚拟映像的高性能。具体详情请参看 WebSphere CloudBurst Appliance 中的“特殊原料”(http://www.ibm.com/developerworks/cn/websphere/techjournal/0909_col_willenborg/0909_col_willenborg.html)

- 一站式的產品支持

作为虚拟映像发布和使用的 WebSphere 应用服务器 Hypervisor 版本，其产品支持包涵了映像中的操作系统与 WebSphere 应用服务器。例如说，在为 VMWare ESX 制作的虚拟映像中，IBM 会提供对 SLES 操作系统的支持与维护。用户不需要另去寻求 Novel 的技术支持。

- 虚拟映像的定期升级

在虚拟映像的维护方面，IBM 会定期发布升级版本。升级版本中包涵 WebSphere 应用服务器以及操作系统的升级补丁，由 IBM 集成测试后为用户发布，节省了用户在系统环境维护方面的工作。

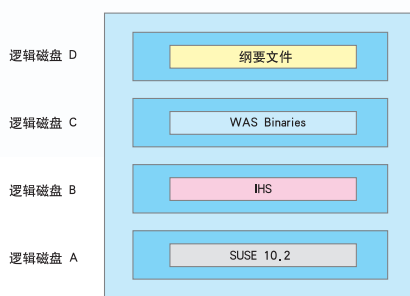
WebSphere 应用服务器 Hypervisor 版本是可以单独购买和使用一个软件产品，同时也可以与一个硬件产品，WebSphere CloudBurst Appliance，一起购买和使用。WebSphere CloudBurst Appliance 是一个快速部署和全面管理企业私有 WebSphere 云环境的硬件产品，其中收纳了 WebSphere 应用服务器 Hypervisor 的虚拟映像。两个产品可谓珠联璧合，携手创建以及智能管理省时省力的 WebSphere 云环境。(详情请见本章 WebSphere CloudBurst Appliance 产品的介绍)

[WebSphere Application Server Hypervisor 细节浏览:](#)

WebSphere 应用服务器 Hypervisor 的虚拟映像由以下四个虚拟逻辑磁盘组成:

- 操作系统磁盘
- WebSphere 应用服务器磁盘
- Web 服务器(即 WebSphere IHS)磁盘
- WebSphere 应用服务器概要文件(Profiles)磁盘

WebSphere 应用服务器Hypervisor



目前提供了对 VMWare ESX 与 PowerVM这两种虚拟环境的支持。基于

VMWare ESX的虚拟映像中的操作系统是SLES 10.2。基于PowerVM的虚拟映像中的操作系统 AIX 6.1.3 64bit。

WebSphere 应用服务器的版本有三种, 分别是7.0, 6.1, 以及装有功能部件包(feature pack)的6.1。(7.0的版本中可自选是否安装功能部件包, 所以7.0只有一个版本。)

WebSphere 应用服务器Hypervisor支持Open Virtualization Format (OVF)。OVF是一个定义包装和发布虚拟映像的

开放式规范。对 O V F 的支持使得 WebSphere 应用服务器 Hypervisor 版本适用于不同虚拟技术实现上。

相关参考资料

- 下载地址: http://www.ibm.com/developerworks/downloads/ws/wasnd/S_TACT=105AGX28&S_CMP=DLMAIN
- 信息中心: <http://www-01.ibm.com/software/webservers/appserv/was/library/v70/hy-ed/info-center.html>

WebSphere 家族新成员 CloudBurst

崔学翔 / 文

IBM 于 2009 年 6 月初发布了一款 WebSphere 家族新产品 - IBM WebSphere CloudBurst。这款产品跟以往 WebSphere 家族产品最大的不同是它是一个硬件设备 (如图 Figure-1), 主要用于在 Hypervisor 设备上 (比如 VMWare ESX server) 简便、快速、可重复性的部署一个复杂的 WebSphere Application Server (后面简称为 WAS) 运行环境。



Figure -1

如果您在公司从事的是基于 WebSphere Application Server 产品开发或者测试的工程师, 您也许会经常碰到下面的问题:

正当项目紧张进行的时候, 您需要几台机器和 WAS 介质 (包括 licenses) 建立一个比较复杂的 WAS 环境去重现和解决一个测试部门提交的 bug, 但是发现在本部门内无法很快找到空余的所需软硬件资源。由于目前大多软件公司的硬件和软件

管理还比较松散, 各个部门 (设计、开发、测试以及其它部门) 基本独立拥有和管理本部门的硬件和软件资源, 这样就对跨部门的资源共享带来很大的麻烦。当然, 除了软硬件资源共享问题, 配置一个比较复杂的 WAS 环境也不是一件轻而易举的事情, 不但需要花费大量的时间和人力在各个机器上重新安装 WAS 产品 (包括所需的大量的补丁和功能包), 还需要对 WAS 进行配置和部署所需的应用。通常对于不是很熟练的技术人员, 手工部署一个复杂的 WAS 环境需要几天甚至一周以上的时间。更麻烦的是, 由于软硬件资源的缺乏, 无法比较长时间的保留一个部署好的 WAS 应用环境, 从而不得不在以后需要的时候重新进行部署。

其实您可以不必为这种事情烦恼, 因为 WebSphere CloudBurst 可以帮助我们轻松解决此类问题。我们先来看一下 WebSphere CloudBurst 提供了哪些好的功能, 能够为我们做什么:

- WebSphere CloudBurst 里面存储了 WAS 各个版本 (6.1.0.x 和 7.0.0.x, 包括最新的补丁包和功能包) 的虚拟镜像介质, 这些经过调优过的介质会最终部署到你的 WAS 环境中去
- 根据 10 年来客户使用 WAS 的经验, WebSphere

CloudBurst里预先放置了很多常用的WAS拓扑定义，使用者可以基于这种拓扑定义快速部署自己的企业级应用。

- WebSphere CloudBurst利用虚拟化的系统资源池可以最大化重用现有硬件软件资源，从而节省企业软硬件的投入资金。

- WebSphere CloudBurst 提供了有效的页面管理台，所有的事情都可以轻松在管理控制台上点击几下鼠标，敲几个键来完成。

- WebSphere CloudBurst 提供了很好的角色划分功能，企业用户可以根据自身的管理需求，对不同的人分配不同的执行权限，从而更有效安全的管理企业内部系统和产品部署

- WebSphere CloudBurst 对自身系统和 WAS 部署都提供了监视和详尽的日志功能，从而可以有效及时的发现问题并对调试解决问题提供帮助。

- WebSphere CloudBurst 提供了完善的软硬件安全机制，确保里面存储的所有数据都是经过加密的，不会通过网络或盗取造成数据泄露。

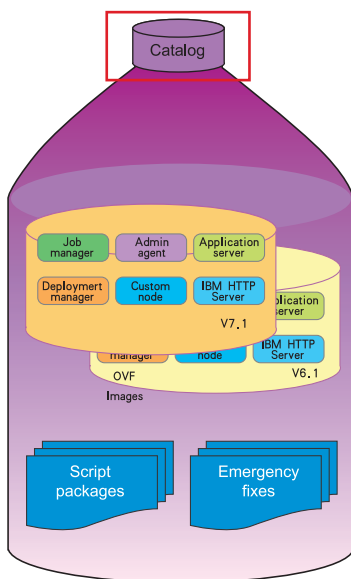
- WebSphere CloudBurst 提供了丰富的编程接口，可以很容易地继承到其他产品中去，比如 IBM 自己的 Rational 和 Tivoli 产品。

- WebSphere CloudBurst可以很好的工作在一个基于虚拟化技术的企业私有云，从而更好的融合到企业内部的云架构中。

了解到 WebSphere CloudBurst 能够提供什么样的功能，我们再来看看它是如何来提供这些功的。WebSphere CloudBurst内部划分几个组建，我们逐一介绍一下。

第一个就是“目录”(catalog)，目录是用来存储介质，补丁安装包，以及用户自己的脚本和应用安装包(如图 Figure-2)。

Figure-2



因为 WebSphere CloudBurst 目前主要支持的是在 Hypervisor 系统上进行分化和部署 WAS 产品，所以目录中存储的 WAS 介质都是基于各种 Hypervisor 所特质的虚拟镜像文件。这种使用 OVF 标准格式的虚拟镜像内嵌了操作了系统，WAS 产品和 IBM HTTP Server，用户不用考虑如何从无到有的搭建一个 WAS 环境，WebSphere CloudBurst 可以按需把这种介质分发到远程 Hypervisor 机上自动展开成为一个配置好的虚拟系统。

第二个重要组件就是“模式”(pattern)，模式里预先放置了一些常用的 WAS 拓扑结构的定义，WebSphere CloudBurst 会按照用户所选定的拓扑定义从 Hypervisor 资源池中自动找到空闲的系统将 WAS 安装部署好。当然，用户也可以从无到有的(或者基于已有的拓扑)定制一个符合自己需求的拓扑结构，存储在模式组件中，供以后重复使用。

我们来看一下 WebSphere CloudBurst 的整体框架图(如图 Figure-3)

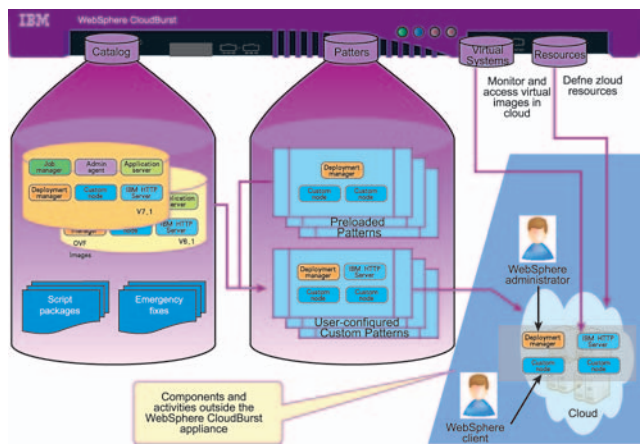


Figure-3

从这张图中，我可以看到 WebSphere CloudBurst 基于所选定的模式，从“目录”中选择合适的 WAS 虚拟介质，把他们部署到企业的私有云上，这个所谓的私有云可以简单理解为一组 hypervisor 系统(目前支持的有 VMWare 的 ESX server, IBM PowerVM 和 zVM)。

当企业有这样一些 hypervisor 系统后，可以按照图 Figure-4 所描述的步骤去具体实施

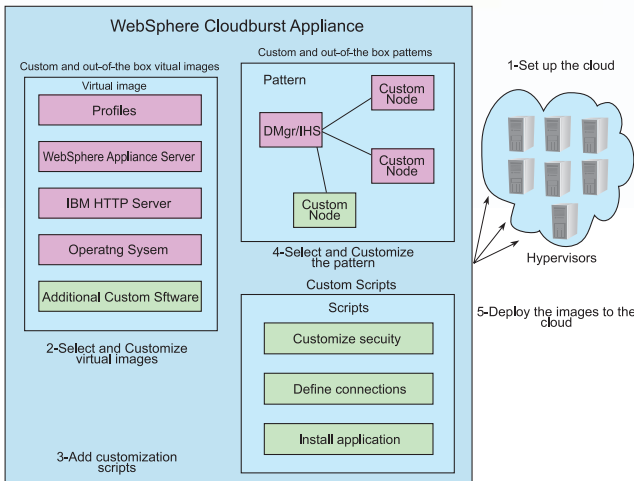


Figure-4

1. 确定已有的hypervisor系统和网络资源, 建立一个私有云;
2. 选择和定制 WAS 的虚拟介质;
3. 加入要按照的企业应用和脚本包 (用来安装部署应用)

4. 选择和定制一个 WAS 环境的拓扑结构;
5. 在虚拟系统上施行真正的部署。

最后我们小结一下, WebSphere CloudBurst 通过预定义的 WAS 虚拟介质和拓扑结构, 能够把 WAS 和企业应用自动部署到 hypervisor 系统环境中去, 显著地减少 WAS 环境的部署时间和人力成本, 有效的利用企业内部软硬件资源, 从而大大地降低企业在软硬件资源和人力资源上面的投入。

参考:

- WebSphere CloudBurst 主页

<http://ibm.com/cloudburst>

- WebSphere Cloud Computing Community (开发者园地)

<http://www.ibm.com/developerworks/spaces/websphereclouds>

- WebSphere CloudBurst 信息中心

<http://publib.boulder.ibm.com/infocenter/wscloudb/v1r0/index.jsp> ■

浅谈 WebSphere eXtreme Scale

兰灵 / 文

WebSphere eXtreme Scale 能够以出色的效率和线性可伸缩性处理海量数据。WebSphere eXtreme Scale 以内存数据网格的形式运行, 动态缓存、分区、复制和管理多个服务器上的应用程序数据与业务逻辑。它提供事务完整性和透明的故障转移, 确保高可用性、高可靠性和一致的响应时间。WebSphere eXtreme Scale 主要具有以下特性:

- **内存管理:** 包含内存管理功能, 支持跨数千种商业服务器的 TB 级数据。

- **提高可用性和容错能力:** 如果主服务器发生故障, 则自动将一个副本服务器提升为主服务器。支持多个副本服

务器。

- **更高的应用程序性能:** 数据存储在内存中, 因而应用程序访问数据的速度极快 (毫秒)。

- **支持线性可伸缩性:** 随着数据量的增长或事务量的增加, 可添加额外的服务器来存储多出的数据并确保应用程序一致地访问数据。

- **应用程序编程模型:** 提供多种编程模型, 用于访问和存储数据。

- **自动地复制数据:** 同步或异步复制数据以确保数据可用性。

- **灵活地定义数据的存放位置:** 指定数据应存储在哪个“区域”, 如服务器

系统、数据中心或某一地理位置, 并由系统负责管理将数据放置到这些区域的方式。

- **动态修改数据布局:** 可在不影响现有应用程序的情况下动态更新模式或数据布局。

- **异构:** 可在任何 J2EE 或 J2SE 1.4+JVM 中以单机形式执行。提供通用数据结构方法, 支持异构服务器环境。

高可伸缩性

WebSphere eXtreme Scale 通过使用数据分区达到高可伸缩性。WebSphere eXtreme Scale 将数据存储在若干服务器

中，这些服务器组成一个或多个数据网格，由于数据网格中的数据服务器相互独立，因此它的规模可以扩展到数千台服务器。

数据分区是将完整的数据拆分为多个分区 (partition)，每个分区只包含部分数据。当部署应用时，用户只需要指定数据分区策略，当服务器启动之后，数据将被自动拆分为多个部分，并平均地分布到不同的服务器上。WebSphere eXtreme Scale有两种分区策略供用户选择：固定分区总数与每个服务器固定分区数。选择前者将始终保持数据被拆分为固定的数目，而如果选择后者，分区数目将随着服务器数目的增加而扩大。在应用系统运行过程中，具体的数据分区与服务器之间的对应关系是对用户透明的，WebSphere eXtreme Scale会根据用户的配置，将所有分区在数据网格中进行合理的分布，以提供更高的可用性与更优的性能。

WebSphere eXtreme Scale通过数据分区的技术达到高可伸缩性。随着工作负载的增大，可以通过增加服务器数目来应对更多的请求。但是，对于伸缩性不强的系统，当增加服务器到一定数目时，由于服务器之间的交互也相应增加，再增加服务器也无法应对更高的工作负载。而在WebSphere eXtreme Scale的服务器分别处理存储在其上的数据，具有很高的独立性。因此WebSphere eXtreme Scale能够提供线性的可伸缩性，其可支持的最大应用负载可随着服务器数目的增大而线性增大。(见图1)

在WebSphere eXtreme Scale中，服务器数目是在运行时刻动态变化

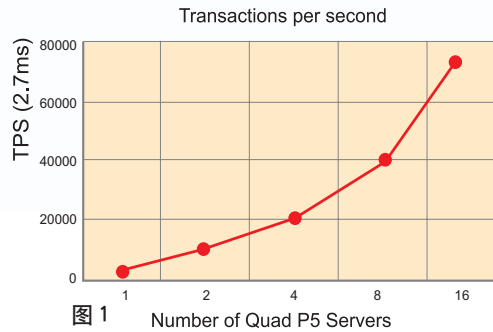


图1 Number of Quad P5 Servers

的，因此，它也提供了动态的可伸缩性。在应用系统运行过程中，随着服务器数目的变化，数据分区将从已有服务器自动转移到新增加的服务器上。并且在数据转移过程中，可以不中断应用系统的正常运行，它可以持续地提供服务。随着数据分区的重新分布，用户请求将分布到更多的服务器上，从而可以支持更高的工作负载。

高可用性

WebSphere eXtreme Scale通过内置的数据副本技术实现了高可用性。

对于WebSphere eXtreme Scale中的每一份数据分区，都提供了一份主数据块 (primary shard) 以及相应的若干份副本数据块 (replica shard)。当主数据分区失效时 (例如该分区所在的服务器宕机)，某一份副本分区会自动转换为主分区，继续提供服务。而为了保证系统的高可用性，

WebSphere eXtreme Scale会自动将同一个数据分区的主从数据块放在不同的服务器上。(见图2)

主数据块与副本数据块之间会自动通过同步、异步两种方式进行复制。其中，同步复制将保证主/从数据完全保持一致，在运行过程中不会丢失数据；而异步复制方式在复制过程中可能丢失数据，但会

大幅提供系统性能。

此外，为提高系统可用性，WebSphere eXtreme Scale还提供了基于地区数据副本的功能。用户可以指定主数据块在地区A (如北京)，而其对应的副本数据块在地区B (如上海)，当位于地区A的服务器因故障全部失效时，地区B的副本数据块可以转换为主数据块继续为用户提供服务。

相关参考资源

WXS 7.0 infocenter: <http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r0/index.jsp>

WXS homepage: <http://www-01.ibm.com/software/webservers/appserv/extremescale/> ■

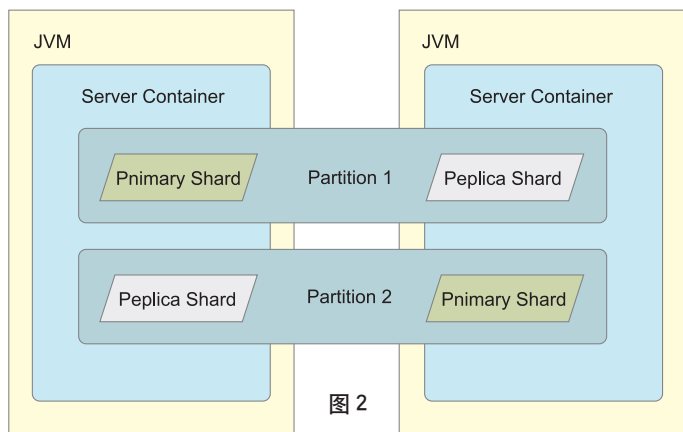


图2

WebSphere 应用基础平台 总揽及最新技术介绍

姚勇 / 文

WebSphere应用服务器,从1998年的诞生到现在,已经发展成为业界最为广泛使用的JavaEE服务器之一。历经10年,今天的WebSphere早已不仅仅是一个应用服务器,而是发展成为一个全面的中间件产品线和构建SOA架构的核心,从而为各种商业应用提供动态整合的业务流程以及高可用的应用基础平台。大家所熟悉的WebSphere应用服务器,作为整个WebSphere家族和SOA的基础平台,也从一个产品逐渐发展成为一个大的产品家族。那么在这一期的新技术专栏里,我们也特别请到了IBM中国开发实验室的专家,为大家揭开这一系列WebSphere/SOA应用基础平台产品的神秘面纱。

这里我先简单对这一系列的WebSphere应用基础平台产品(WebSphere Application Infrastructure)做一个简单的分类和介绍。后面还会分别有几位IBM中国实验室的专家对其中非常重要的一些产品做更进一步的说明。

整个WebSphere的产品家族大体可以分为下面几类产品:应用基础平台;流程管理;应用互连与整合;门户;电子商务;移动和语音;业务规则管理与优

化;以及行业解决方案。而应用基础平台产品线,作为WebSphere/SOA的基础,能够帮助客户非常高效地去搭建,部署,和管理各种商业应用,并能够在各种复杂条件下非常智能的去管理应用基础平台。

下面这张表涵盖了WebSphere基础应用平台主要产品分类,以及它们所对应的主要业务需求。

大家可以去以下这个站点,查到所有WebSphere基础应用平台相关产品的信息。

<http://www.ibm.com/software/websphere/products/application-infra-structure/>

在给大家做了一个整体介绍之后,后面是分别对WebSphere基础应用平台中的WAS V7功能部件包,WAS Hypervisor Edition, WebSphere CloudBurst Appliance, 和WebSphere eXtreme Scale几个最新产品的简介。今后,我们还会有更多,更详细的分享,通过WebSphere中国介绍给大家。■

| 业务需求 | 产品分类 | 产品 |
|------------------------------------|--|---|
| 一致,敏捷,高性价比的构建,部署,和管理各种商业应用,达成商业目标。 | 应用基础 (Application Foundation) | WebSphere 应用服务器 (WAS) WAS 功能部件包 (Feature Packs) WAS Community Edition WebSphere sMash |
| 适合应对动态商业环境,降低IT基础平台成本开销 | 智能管理 (Intelligent Management) | WAS ND (Network Deployment) WAS Hypervisor Edition WebSphere CloudBurst Appliance (WCA) WebSphere Virtual Enterprise (WVE) |
| 满足对超大规模应用,高性能实时响应,及高可扩展的核心应用的要求。 | 极限事务处理 (Extreme Transaction Processing) | WebSphere eXtreme Scale (WXS) WebSphere Compute Grid (WCG) WebSphere Real Time |

IBM WebSphere MQ 备份和恢复

甘荃/文



对于生产系统，备份和恢复是非常重要的。在出现紧急故障，例如磁盘损坏，文件以外删除，电源故障等，我们需要制定一个完善的系统恢复方案。IBM WebSphereMQ 产品架构某些方面很类似于数据库产品，例如WebSphereMQ 的队列和数据库的表的功能很类似。在一个应用系统中，数据库的备份和恢复是非常重要的。同样，WebSphere MQ 作为通讯中间件产品，我们也需要考虑 WebSphere MQ 备份和恢复方案。数据库的备份恢复分为物理和逻辑备份和恢复。WebSphere MQ 的备份恢复也可以分为系统，介质备份和对象备份和恢复。值得注意的是 WebSphere MQ 这系统和对象备份恢复方不能做到类似数据库的基于时间点和故障点恢复，下面我们对 WebSphere MQ 这三种备份和恢复方法进行介绍。

1 WebSphere MQ 的系统备份和恢复

1.1 系统备份 WebSphere MQ 队列管理器方法

1) 首先要确保队列管理器不在运行。如果您尝试备份正在运行的队列管理器，备份可能会不一致，因为复制文件时可能正在文件进行更新。

备份应该在没有任何数据传输的时候，如果可能，以正常方法停止您的队列管理器。尝试执行 `endmqm -w` (等待关闭)；只有当它失败时，使用 `endmqm -i` (立即关闭)。

2) 使用配置文件中的信息，查找队列管理器放置其数据和日志文件的目录。

从配置文件 `/var/mqm/mqs.ini` 定位队列管理器目录。在 `QueueManager` 节查找到要备份的队列管理器。

配置文件 `mqs.ini` 清单如下：

AllQueueManagers:

DefaultPrefix=/var/mqm

ClientExitPath:

ExitsDefaultPath=/var/mqm/exits

ExitsDefaultPath64=/var/mqm/exits64

LogDefaults:

LogPrimaryFiles=3

LogSecondaryFiles=2

LogFilePages=1024

LogType=CIRCULAR

LogBufferPages=0

LogDefaultPath=/var/mqm/log

QueueManager:

Name=MS_JLW_DS

Prefix=/var/mqm/node1

Directory=MS_JLW_DS

其前缀和目录属性标识了队列管理器目录。队列管理器目录的全路径为：`<Prefix>/qmgrs/<Directory>`，例如：队列管理器 `MS_JLW_DS` 的全路径为 `/var/mqm/node1/qmgrs/MS_JLW_DS`。

从队列管理器目录中的 `qm.ini` 配置文件定位队列管理器日志目录。日志节的 `LogPath` 属性标识了该队列管理器的日志目录。

队列管理器 `MS_JLW_DS` 的 `qm.ini` 的配置清单：

ExitPath:

ExitsDefaultPath=/var/mqm/exits/

ExitsDefaultPath64=/var/mqm/exits64/

Log:

LogPrimaryFiles=20

LogSecondaryFiles=10

LogFilePages=16384

```
LogType=CIRCULAR
LogBufferPages=0
LogPath=/var/mqm/node1/log/MS_JLW_DS/
LogWriteIntegrity=TripleWrite
```

Service:

```
Name=AuthorizationService
EntryPoints=13
```

ServiceComponent:

```
Service=AuthorizationService
Name=MQSeries.UNIX.auth.service
Module=/opt/mqm/lib64/amqzfu
ComponentDataSize=0
```

从清单中可以看出队列管理器 MS_JLW_DS 的 LogPath 为 /var/mqm/node1/log/MS_JLW_DS/。

3) 备份所有队列管理器的数据和日志文件目录, 包括所有子目录。

确保没有丢失任何文件, 特别是日志控制文件和配置文件。某些目录可能是空的, 但是以后恢复备份时全部需要, 因此需要全部完整地保存它们。

例如: 针对上面的队列管理器 MS_JLW_DS, 我们需要备份目录 /var/mqm/node1/qmgrs/ MS_JLW_DS 和目录 /var/mqm/node1/log/MS_JLW_DS/。

4) 为了保留备份文件的所有权限, 例如: 在 unix 系统我们可以使用 tar 命令完成。

例如:

```
tar cvf MS_JLW_DS.qm.tar /var/mqm/node1/qmgrs/
MS_JLW_DS
```

```
tar cvf MS_JLW_DS.qmlog.tar /var/mqm/node1/log/
MS_JLW_DS/
```

1.2 恢复 WebSphere MQ 队列管理器的方法

1) 使用 dspmq 命令检查, 确保队列管理器不在运行。

2) 查找队列管理器放置其数据和日志文件的目录。方法同备份 MQ 队列管理器第 2 步。

3) 对您要放置备份数据的目录进行一个临时备份, 例如把队列管理器目录重命名。

4) 把备份的队列管理器数据和日志文件复制到正确的

位置。

5) 检查得到的目录结构, 确保您有所有必需的目录。

6) 如果正确备份和恢复了数据, 队列管理器可以成功启动。

2 WebSphere MQ 的 media 备份和恢复

为了使用 media 备份和恢复, 队列管理的日志模式必须是 linear 模式。

2.1 WebSphere MQ media 备份

对于一个运行的队列管理器, 我们可以做 media 备份, 备份命令是 rcdmqimg。

rcdmqimg 命令是把 WebSphere MQ 对象或一组对象的镜像写到日志中。

备份命令参考如下:

> 备份队列管理器 QM1 的所有对象

```
rcdmqimg -m QM1 -t all *
```

> 备份队列管理器 QM1

```
rcdmqimg -m QM1 -t qmgr
```

> 备份所有以 SYSTEM 和 QL 开始的队列

```
rcdmqimg -m QM1 -t q SYSTEM*
```

```
rcdmqimg -m QM1 -t q MQ61*
```

> 备份所有的 process

```
rcdmqimg -m QM1 -t prcs *
```

2.2 WebSphere MQ media 恢复

进行 media 恢复, 需要在队列管理器运行的状态下进行。

使用命令 rcrmqobj, 被恢复的对象将自动地恢复到最后的 commit 状态。

恢复命令参考:

> 恢复队列管理器 QM1 的所有对象

```
rcrmqimg -m QM1 -t all *
```

> 恢复队列管理器 QM1 的所有队列

```
rcrmqobj -m QM1 -t q *
```

> 恢复所有以 SYSTEM 和 QL 开头的名字的队列

```
rcrmqimg -m QM1 -t q SYSTEM*
```

```
rcrmqimg -m QM1 -t q QL*
```

> 恢复所有的 Process

```
rcrmqimg -m QM1 -t prcs *
```


3 WebSphere MQ的对象备份和恢复

WebSphere MQ 的用户在有些情况下需要备份队列管理器的各个对象。而 WebSphere MQ 本身并不提供备份的功能。

WebSphere MQ SupportPac 提供了一个程序 MS03 可以进行 WebSphere MQ 队列管理器中的对象的备份。SupportPac (saveqmgr) 保存所有的对象，例如 queues, channels 和 Process 等，把这些定义输出到一个文件中。其原理是通过 Programmable Command Format(PCF) 命令将队列管理器及其对象的属性转存到相应数据结构中，然后再以 WebSphere MQ 脚本命令的格式将这些属性以脚本命令的格式保存到一个脚本文件中。

恢复队列管理器时只需要创建一个同名的队列管理器，然后运行备份时生成的脚本即可。该程序是一组使用 C 语言和 MQI 接口编写的代码，用户需要根据自己所使用的操作系统平台对该程序进行编译。该软件的执行需要队列管理器及其命令服务器处于运行状态。

该 SupportPac 提供了不同平台的 makefile。

该 SupportPac 可以在 WebSphere MQ 的 SupportPac 网站下载，网址为：<http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg27007197>和http://www-01.ibm.com/support/docview.wss?rs=171&q1=mA1J&uid=swg24000673&loc=en_US&cs=utf-8&lang=en

3.1 MS03 的安装配置步骤

目前 SupportPac Version 是 6.1.2。支持的版本有 WebSphere MQ7.0, WebSphere MQ6.0, WebSphere MQ 5.3。

在 Intel 平台上安装 SupportPac 步骤如下：

- > 复制 ms03_win.zip 文件到一个临时目录。
- > 使用 winrar 等工具对文件进行解压。
- > 然后可以删除 ms03_win.zip

> 随后按照解开后的 readme 进行操作即可。

在 Unix 平台上安装 SupportPac 步骤如下：

> 下载 ms03_unix.tar.Z 并存放在 /tmp 目录下。

> 执行 `uncompress -fv /tmp/ms03_unix.tar.Z`，对文件进行解压。

> 执行 `tar -xvf /tmp/ms03_unix.tar`，进行解包。

> 执行 `rm /tmp/ms03_unix.tar`，删除 tar 包。

> 解包之后将产生所需的文件。

> 随后按照解开后的 readme 进行操作即可。

MS03 编译生成的可执行文件 saveqmgr 的详细命令参数可以在源文件 saveqmgr.c 中查到。

在一个应用系统中，数据库的备份和恢复都非常重要，同样，WebSphere MQ 作为通讯中间件产品，我们也需要考虑 WebSphere MQ 备份和恢复方案。

MS03 可以备份远程队列管理器，但在编译的时候需要选择 MQ 客户端动态链接库。该编译选项在各个平台的 makefile 中也是已经定义好了的。

3.2 备份 WebSphere MQ 对象的方法

备份 WebSphere MQ 配置可以在线进行，不需要停止队列管理器，下面举例说明。

备份本地队列管理器 QM1 的步骤如下：

启动队列管理器：`strmqm QM1`

启动命令服务器：`strmqcsv QM1`

备份队列管理器：`saveqmgr -m QM1 -v 53 -f saveqmgr.tst`

WebSphere MQ 的对象配置被保存在 saveqmgr.tst 文件中。该文件可以用来恢复用。

3.3 恢复 WebSphere MQ 对象的方法

如果 WebSphere MQ 系统被破坏了，或者需要重新创建，我们可以使用备份的对象文件进行恢复。

恢复本地队列管理器的步骤如下：

创建本地队列管理器：`crtmqm QM1`

恢复队列管理器及其对象：`runmqsc QM1 < saveqmgr.tst`

有些情况下可能需要重新启动队列管理器。■

IBM WAS 6X 内存泄漏分析

张建平 / 文

一. 前言

我们很多 WAS 的管理员可能都会遇到过 OOM (OutOfMemory Error) 的问题, 最开始都会觉得很迷茫, 不知所措, 下面我就为大家介绍一下关于 OOM 问题的分析过程, 供大家在今后的案例中做参考。

我们用一套 WAS6.0.2.23 环境做案例, 这套环境是部署在 HP-UX B.11.23 平台运行的。一会儿, 我们会从下面的日志中看到大量的异常抛出, 后来经过几种分析方法可以得出是由于应用程序代码问题使 WAS 系统内存泄漏, 从而导致服务器性能下降, 不可用。下面我就来介绍一下这次分析故障的全过程。

二. 检查系统环境以及 WAS 的相关版本:

在分析问题前, 我们需要做一些前期的相关信息收集, 以便之后分析问题的时候使用。我们需要检查该 WAS 平台部署的操作系统环境、WAS 的具体版本情况和 JDK 版本情况, 具体步骤操作如下:

1. 操作系统版本信息收集

我们需要使用管理员权限 (root) 登陆到系统, 在系统中执行 `uname -a`, 输出如下:

```
HP1#[/]uname -a
HP-UX HP1 B.11.23 U 9000/800 2089833882 不限用户数的许可证
```

通过以上信息可以得出该操作系统是 HP-UX B.11.23 PA32 位操作系统。

2. WAS 版本信息收集

我们需要到 `/opt/IBM/WebSphere/AppServer/bin` 下, 执行 `versionInfo.sh`, 输出如下:

```
HP1#[/opt/IBM/WebSphere/AppServer/bin] ./versionInfo.sh
WVER0010I: Copyright (c) IBM Corporation 2002, 2005;
```

All rights reserved.

WVER0011I: WebSphere Application Server R6.0

WVER0012I: VersionInfo reporter V1.15.3.2, 日期 11/23/06

IBM WebSphere Application Server 产品安装状态报告
报告的日期和时间 2009-6-29 09:29:36+0800

安装:

产品目录 /opt/IBM/WebSphere/AppServer

版本目录 /opt/IBM/WebSphere/AppServer/properties/
version

DTD 目录 /opt/IBM/WebSphere/AppServer/properties/
version/dtd

日志目录 /opt/IBM/WebSphere/AppServer/logs

备份目录 /opt/IBM/WebSphere/AppServer/properties/
version/update/backup

TMP 目录 /var/tmp

安装平台:

名称 IBM WebSphere Application Server

版本 6.0

产品列表:

ND 已安装

已安装的产品:

名称 IBM WebSphere Application Server - ND

版本 6.0.2.23

标识 ND

构建级别 cf230739.19

构建日期 10/4/07

结束安装状态报告

通过以上信息可以得出该 WAS 版本为 6.0.2.23, 并且是 ND 模式部署。

3. JDK 版本信息收集

需要到/opt/IBM/WebSphere/AppServer/java/bin下，执行 java-version，输出如下：

```
HP1#[/opt/IBM/WebSphere/AppServer/java/bin].java -
version
```

```
java version "1.4.2.13"
```

```
Java(TM) 2 Runtime Environment, Standard Edition (IBM
build 1.4.2.13-070317-13:26 20070708)
```

```
Java HotSpot(TM) Server VM (build 1.4.2.13-
070317-15:01-PA_RISC2.0 PA2.0 (aCC_AP), mixed mode)
```

```
IBM Java ORB build orb142-20070708 (SR9)
```

```
XML build XSLT4J Java 2.6.13
```

```
XML build IBM JAXP 1.2.0
```

```
XML build XML4J 4.3.11
```

通过以上信息可以得出，该 JDK 版本为 1.4.2.13。

到目前为止，我们所需要的基本信息已经收集完毕，下面我们具体分析问题。

三. 问题描述

在系统运行高峰时期，系统出现 WEB 页面打开缓慢，登陆应用系统缓慢，登陆后应用运行缓慢，有时甚至无法访问，出现 HANG 机的状态。

四. 问题分析前准备工作

1. 问题简要概述

通过应用访问出现的现象，我们可以初步定位该问题类似于 hang 机的问题，hang 机情况是一种特殊情况，主要表现为客户的请求无响应，但是 java 进程仍然存在。该问题引起的原因可能有很多种，主要有以下几种原因会造成 hang 机情况：

A. 程序代码某些线程造成死锁或程序中存在死循环。

B. 程序代码编写不当造成某些资源无法被释放或被耗尽（如：网络链接，数据库链接，内存使用情况）

C. 不适当的调优参数引起性能瓶颈而引起 hang 机

a. Web 容器 (JVM), 数据库

b. Web 容器等待一个外部资源的响应 resource

c. 资源的限制 (如 cpu、Memory 典型的 Java heap、

network)

根据以上这些情况来看，可能仅仅靠看日志无法分析问题发生的根本原因，必须要进行一次全面的检查分析。

为了不影响客户访问，我们可以让负载均衡停止给问题 server 分发用户。

2. 问题信息收集

根本的解决 hang 问题的办法，是要找出问题的根源。要找到问题的根源需要充分完整的数据进行判断和分析。需要在问题发生的时候收集完整的数据。我们可以根据 IBM 官方网站关于 HANG 机时所需要的数据进行收集：

```
http://www-1.ibm.com/support/docview.wss?
rs=180&context=SSCMPB9&q1=MustGatherDocument&uid=
swg21127574&loc=en_US&cs=utf-8&lang=en
```

五. 分析收集到的数据

1. 分析 server 日志中 SystemOut.log 日志文件

发现日志中出现大量报错信息：

```
[09-6-30 11:08:32:178 GMT+08:00] 00000087
TimeoutManage I WTRN0006W: 120 秒后事务
000001222F20C35F00000006001D0C0CB73E7897D992DD80
D35710243AB5595B750F960A000001222F20C35F00000006001
D0C0CB73E7897D992DD80D35710243AB5595B750F960A00000001
已超时。
```

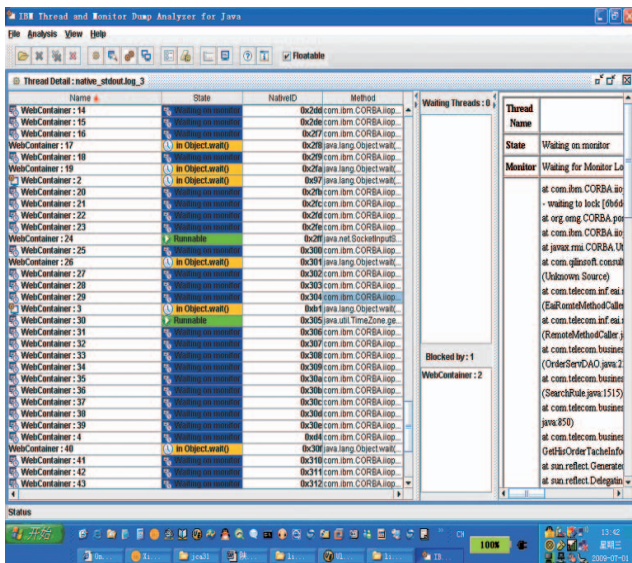
```
[09-6-30 11:08:32:248 GMT+08:00] 00000028
TimeoutManage I WTRN0006W: 120 秒后事务
000001222F20C3A100000006001D0C19B73E7897D992DD80
D35710243AB5595B750F960A000001222F20C3A100000006001
D0C19B73E7897D992DD80D35710243AB5595B750F960A00000001
已超时。
```

```
[09-6-30 11:08:50:790 GMT+08:00] 0000012a
CoordinatorCo W HMGR0152W:检测到 CPU 饥饿。当前线程
调度延迟是 27 秒
```

通过上述大量报错信息可以得出有事物在执行的时候出现超时现象，并且发现了 CPU 饥饿的情况，出现超时的话，我们需要分 javacore 来看一下具体超时的线程。

2. 分析 Server 日志中 native_stdout.log 中的 javacore

这里要说明一下，HP-UX 操作系统在做了 javacore 后，不会单独生成 javacore 文件，所有 javacore 信息会记录在 native_stdout.log 中。我们通过 IBM 的 JCA 工具进行分析 native_stdout.log 中的线程堆栈信息：



检查发现在出现问题的时候，WebContainer 线程堆栈中存在大量的线程在处于等待状态，而且都属于同一个线程：

```
t com.ibm.CORBA.iiop.ClientDelegate.is_local
(ClientDelegate.java:585)
- waiting to lock [6b6d49b0] (a com.ibm.CORBA.iiop.
ClientDelegate)
at org.omg.CORBA.portable.ObjectImpl.is_local
(ObjectImpl.java:383)
at com.ibm.CORBA.iiop.UtilDelegatImpl.isLocal
(UtilDelegatImpl.java:670)
at javax.rmi.CORBA.Util.isLocal(Util.java:262)
```

看到这里，我们可初步认定与 CORBA 的 iioop 有关，但是现在还不能确定故障原因，因为可能有应用在使用这个协议，导致在 javacore 线程堆栈信息中存在着大量的事物超时。

继续分析 native_stdout.log 发现下面的信息

```
: 1326139K->1229114K(1441792K), 30.4377153 secs]
1326139K->1229114K(1559808K), [Perm : 148194K->
144027K(148736K)] Heap after GC invocations=10381:
```

Heap

```
def new generation total 118016K, used 0K [b0c00000,
b8c00000, b8c00000)
```

```
eden space 104960K, 0% used [b0c00000, b0c00000,
b7280000)
```

```
from space 13056K, 0% used [b7f40000, b7f40000,
b8c00000)
```

```
to space 13056K, 0% used [b7280000, b7280000,
b7f40000)
```

```
tenured generation total 1441792K, used 1229114K
[58c00000, b0c00000, b0c00000)
```

```
the space 1441792K, 85% used [58c00000, a3c4e988,
a3c4ea00, b0c00000)
```

```
compacting perm gen total 144128K, used 144027K
[40c00000, 498c0000, 58c00000)
```

```
the space 144128K, 99% used [40c00000, 498a6eb0,
498a7000, 498c0000)
```

```
}, 30.4479307 secs]
```

```
java.lang.OutOfMemoryError: Java heap space
```

```
Dumping heap to java_pid11913.hprof ...
```

通过分析 GC 日志发现，出问题时候 GC 垃圾回收消耗了 30 seconds 的时间，这是不正常的。说明系统在内存使用上可能存在问题，导致 CPU 时间片不够用。并且发现 java.lang.OutOfMemoryError 错误，这说明系统存在内存泄漏现象，需要检查一下 JVM 配置参数和具体分析内存对象使用情况。

因为环境是部署在 HP-UX 平台的，所以出现内存泄漏的时候，系统不会自动生成 dump 文件，必须要配置一些参数能够在内存泄漏的时候，生成 dump 文件。

我们可以根据 IBM 官方网站的要求设定参数，当 heap 堆出现异常时，会出现 heapdump 镜像文件。

```
http://www-01.ibm.com/support/docview.wss?
uid=swg21242314
```

六. 检查 JVM 配置参数

在出现内存泄漏的情况下，我们应该首先检查一下这个

server的JVM配置参数是否配置合理, 因为HP-UX平台下的WAS有些参数必须要在JVM实参中配置才可以使server正常使用。

分析 server.xml 文件里的配置参数:

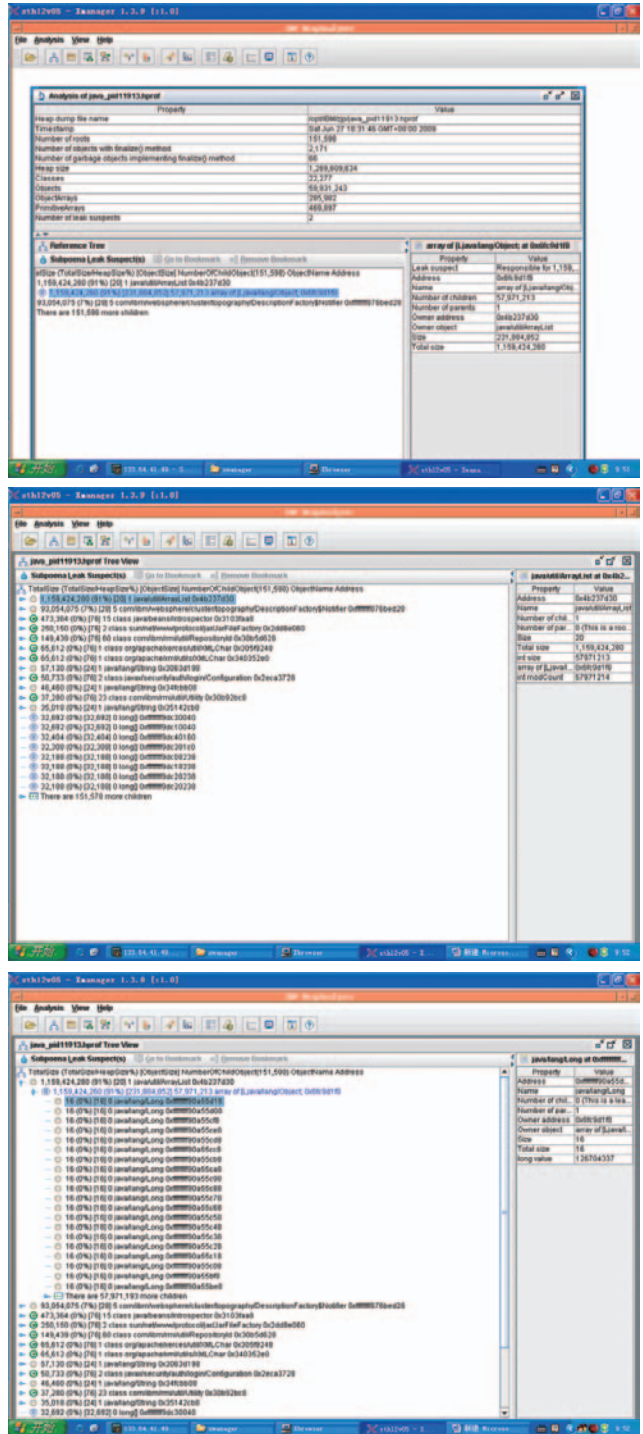
```
<jvmEntriesxml:id="JavaVirtualMachine_1170571581490"
verboseModeClass="false" verboseModeGarbage
Collection="true" verboseModeJNI="false" initialHeapSize=
"512" maximumHeapSize="1536" runHProf="false"
debugMode="false" debugArgs="-Djava.
compiler=NONE-Xdebug-Xnoagent-Xrunjdpw:
transport=dt_socket,server=y,suspend=n,address=7777"
genericJvmArguments="-Djava.awt.headless=true-XX:
+DisableExplicitGC-XX:+HeapDumpOnOutOfMemoryError-
XX:+PrintClassHistogram-XX:+PrintGCTimeStamps-XX:
+PrintGCDetails-XX:+PrintHeapAtGC-XX:PermSize=128m-
XX:MaxPermSize=384m-XX:MaxNewSize=128m-XX:
MaxDirectMemorySize=256m-Djava.net.
preferPv4Stack=true-Xverbosegc:file=/tmp/fileout/file.out"
disableJIT="false">
```

根据 IBM 的信息中心中的提示, HP-UX 上部署的 WAS 应该配置的参数我们这里都已经配置, 所以可以认定 server 的参数配置合理, 在上面的信息中我们还看到了之前说的在内存泄漏需要配置的参数, 在该 server 的参数中也存在 (-XX: +HeapDumpOnOutOfMemoryError), 证明本次内存泄漏已经生成了 heapdump 镜像文件。(这个只是检查的一种方法, 其实在 native_stdout.log 日志中我们可以看到 dump 出了一个文件的记录, 证明参数中配置过了相关参数, 但是需要提醒的是, 在分析问题之前, 参数配置的检查是必须的。)

七. 分析 java_pid11913.hprof 内存镜像 dump 文件

通过 IBM 的 HeapAnalyzer 工具分析 java_pid11913.hprof 文件发现:

JVM 1.5 G Heap 堆, 90% 都是 java/util/ArrayList 对象
JVM 堆栈中存在着1亿左右的java.util.ArrayList对象没有释放。
这是极其不合理的



同时我们使用 MDD4J 工具进行 java_pid11913.hprof 文

件分析。结果如下：

JVM 1.5 G Heap 堆, 90% 都是 java/util/ArrayList 对象
JVM 堆栈中存在着1亿左右的java.util.ArrayList对象没有
释放。

和之前 HeapAanlzer 工具分析出的结果相同。

– Leaking class suspects section

The analysis has found 5 drop suspects with the top most
suspect showing a drop in total reach size accounting for 30
percent of the total heap size of the primary dump.

Leak suspect [1]

The leak suspect root java.util.ArrayList is responsible
for 91% of the total Java heap. It has 1,159,424,280 refer-
ences to the type: java.lang.Long. Please review the object
java.util.ArrayList to assess how much Java heap is required
by the object.

Suspected memory leaking regions:

Owner chain – Dominator tree:

The leaking root is either a root node in the heap space
or there are fewer than three parents to the root node.

java.util.ArrayList, object0x5e72a680, reaches: 895,654,
568 bytesLEAK_ROOT <--- Object suspected of leaking
|java.lang.Object[], array0x834e60e8, reaches: 895,654,
556 bytesLEAK_CONTAINER <---Object holding the leak-
ing objects

|---Leaking unit:

|---java.lang.Long, object0x9120a928, reaches: 609,031,
208 bytes

|---java.lang.Long, object have grown by 1,159,424,280
instances

The leak suspect is a class of type: java.util.ArrayList

The leak suspect size is: 896MB

The object the leak suspect is holding that contains leak-
ing objects is a class of type: java.lang.Object[][]

The object the leak suspect is holding that contains leak-
ing objects has a size of: 896MB

The object that is suspected to be leaking is a class of

type: java.lang.Long

The object that is suspected to be leaking has a size of:
609MB

The object that is suspected to be leaking occurs: 1,159,
424,280 times.

Leak suspect [2]

The leak suspect root java.util.ArrayList is responsible
for 91% of the total Java heap. It has 1,159,424,280 refer-
ences to the type: java.lang.Long. Please review the object
java.util.ArrayList to assess how much Java heap is required
by the object.

Suspected memory leaking regions:

Owner chain – Dominator tree:

The leaking root is either a root node in the heap space
or there are fewer than three parents to the root node.

java.util.ArrayList, object0x5e72a680, reaches: 895,654,
568 bytesLEAK_ROOT <---Object suspected of leaking
|java.lang.Object[], array0x834e60e8, reaches: 895,654,
556 bytes

|---java.lang.Long, object0x9120a928, reaches: 609,031,
208 bytesLEAK_CONTAINER <--- Object holding the leak-
ing objects

|---Leaking unit:

|---java.lang.Long, object0x5e79c188, reaches: 16
bytes

|---java.lang.Long, object have grown by 1,159,424,280
instances

The leak suspect is a class of type: java.util.ArrayList

The leak suspect size is: 896MB

The object the leak suspect is holding that contains leak-
ing objects is a class of type: java.lang.Long

The object the leak suspect is holding that contains leak-
ing objects has a size of: 609MB

The object that is suspected to be leaking is a class of
type: java.lang.Long

The object that is suspected to be leaking has a size of:

16 bytes

The object that is suspected to be leaking occurs: 1,159,424,280 times.

八. 分析总结

综合上面的分析，我们的结论是应用程序使用的 java.util.ArrayList 对象方法占用了 JVM 中 91% 的内存使用，并且存在大量的内存泄漏，导致系统在业务高峰时候容易出现系统性能下降，最终导致系统不可用。

九. 解决方案及建议

1. 请相关开发人员检查程序代码，检查程序架构及逻辑，由于内存泄漏对象为 java.util.ArrayList，所以像 HashMap、Vector?、java.util.ArrayList 等静态集合类的对象方法使用最容易引起内存泄漏，因为这些静态变量对象方法的生命周期与应用程序是一致的，如果该 java.util.ArrayList 是静态的，那么它肯定将一直存在，而其中所有的 Object 对象也就不能被释放，因为它们也将一直被该 java.util.ArrayList? 引用。所以请检查程序中的 java.util.ArrayList 对象包含的代码是否存在问题。并进行相关代码、程序的修改。

2. 检查修改完程序后，重新发布应用程序。监控看问题是否再次发生，如果问题依然发生我们再继续收集新的数据来分析是否原因和之前相同。

3. 建议在新应用程序上线前，在测试环境做好压力测试。

十. 总结

以上就是我做的一次内存泄漏问题分析的全过程，关于分析工具，全部是由 IBM 官方网站提供的免费工具，这类工具定期会有更新，但是没有技术支持。当然，并不能说我使用的工具是最适合的，大家还需要看哪些工具适合自己的实际情况，而使用分析工具，制定分析方法。

虽然这次分析的操作系统是 HP-UX，可能会有些局限性，但是其实其它操作系统都是大同小异，无非就是步骤问题需要稍微改动，比如像：AIX 或者 LINUX 等操作系统，可能根本不需要加那些附属的参数，就可以直接生成 DUMP 文件，javacore 文件也会单独生成等等，诸如此类的问题有很多，但是分析内存泄漏问题基本可以按照我所说的方法去分析问题。所有的事情都重在实践，没有实践过肯定不会发现新的问题，没有发现新的问题，肯定就不会有新的收获。当代社会技术在不断更新，作为 IT 界的精英们也应该不断地创新自己的技术，这样才能跟上这个飞一般前进的社会。

最后再给大家一个 IBM 官方网站关于如何选择合适的分析工具的链接，可以看一下各种免费分析工具的用途，供大家参考，来选择自己适合的工具，

http://www.ibm.com/developerworks/cn/websphere/techjournal/0807_supauth/0807_supauth.html ■

使用 Websphere Business Monitor 监控 Message Broker 中的自动流

赵卓、张楠 / 文

了解 WebSphere Business Monitor V6.2 如何监控 Message Broker 中的自动流，以及在 Websphere Process Server 和 Message Broker 的混合环境中如何进行端到端的监控。本文与以下业务角色的工作息息相关：监控管理人员、流程开发人员和业务分析人员。

引言

WebSphere Business Monitor 是（以下称为 Monitor）IBM 的 Business Process Management (BPM) 解决方案的

一个有机组成部分，完成了从流程运行到流程建模的闭环。作为一个全面的业务活动监视 (BAM) 解决方案，提供了接近实时的业务性能视图。Monitor 实时监控运行中的流程，及时发

现并消除性能瓶颈,同时跟踪关键性能指标(Key Performance Indicators) 并且统计流程信息。在发现运行中的流程出现反常情况时,主动做出响应,从而可以计划和作出有针对性的措施。同时还更多的挖掘商业智能,进一步进行趋势分析。

本文将说明如何对Message Broker的流进行业务监控,特别是如何在Webphere Process Server和Message Broker的混合环境中进行端到端的监控。通过这一完整的监控方案,不仅可以实现BPM的监控,更可以全面获得在BPM中业务流程所调用的Message Broker自动化流的相关信息,使得监控可以从流程环节深入到流程的节点内。本文所说明的方案应用的具体版本信息是:Message Broker 6.1.0.4、WebSphere Business Monitor 6.2、WPS 6.2

Message Broker 的监控方案和消息流的开发

Message Broker 是一款 Enterprise Service Bus (ESB) 产品,具备将多种通信协议、数据格式和业务系统接入总线的强大能力。所有的服务以流的形式接入 Message Broker。

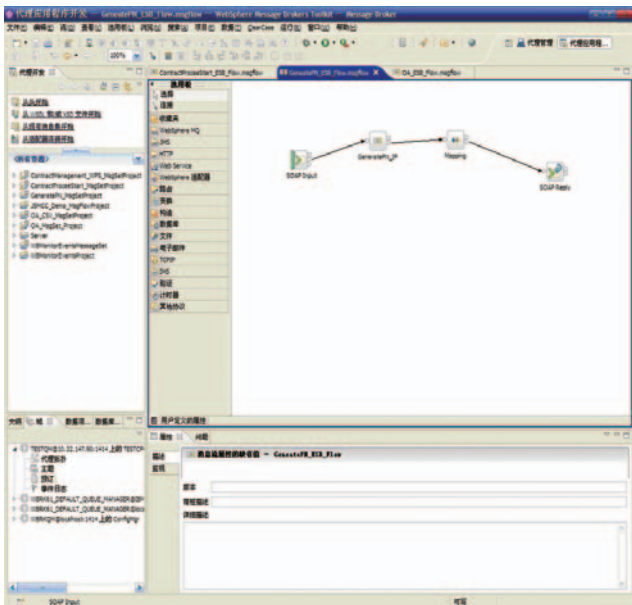


图 1. Message Broker 消息流编辑器 - 服务接入

Message Broker 提供了一个 Support Pac, 通过它可以获得 Message Broker 运行的统计信息, 如可以看到当前每一类服务的处理情况、CPU 时间、处理数量、处理数据大

小等信息。



图 2. Message Broker Support Pac 提供的监控

这一监控方式的重点是IT层面的监控,更多的还是获取在运行环境中的硬件资源使用情况的信息。对于在业务层面面临的问题,如一段时间以来流的汇总执行数据,当前流的执行状态等等,仍然显得无能为力。同时提供的展示手段也有限,不能查看到每一条执行的流的微观信息,也不能定义关键性能指标(Key Performance Indicators)、发现问题、预测趋势、报告异常、采取措施。

使用 WebSphere Business Monitor 监控 Message Broker 流程

CEI 机制

Common Event Infrastructure(CEI,公共事件基础架构)是一种旨在为业务应用提供事件管理的可嵌入技术。这种事件基础架构可以将来自多个异构事件源的事件集中在一起并予以持久化,然后再将事件分发给不同的消费者。

通过使用 CEI,松散耦合的各种产品对各自的事件管理加以整合,从而可以实现跨域企业资源和事件关联的端到端监管。举例来说,一个网络监控应用发出的事件和一个安全监控应用发出的事件,如果不使用 CEI来进行集中管理,而是各自为政,那么它们各自的事件是很难关联起来,从而实现统一监管的。

CEI可以方便地实现事件的产生、传送、持久化和消费,但它并不负责定义事件本身。事件是由事件源,即业务应用来定义的,包括事件类型、事件组、过滤条件和关联关系等。

Monitor通过WebSphere Application Server实现了CEI功能,它主要包括以下几个组件: Common Base Event、

Emitter、Event Service 和 Event Catalog。而使用 CEI 的业务应用或解决方案应该包含如下组件：Event Source、Event Consumer、Event Catalog Application。

图3展示了事件通过CEI从事件源到事件消费者的大致过程

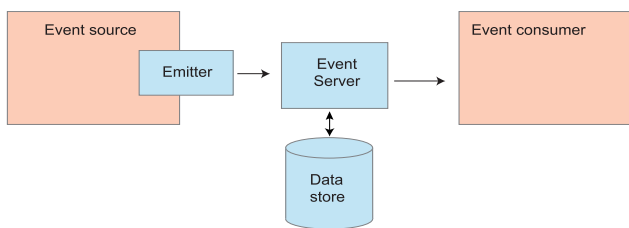


图 3. 从事件源到事件消费的过程

安装配置 message-driven bean (MDB)

Monitor 对 Message Broker 的监控使用 WebSphere MQ 作为 JMS Provider。在 Message Broker 的队列管理器中需要做以下配置：

1. 确保有 TCP listener 定义并启动。
2. 确保 Server Connection 通道已建立。

3. 在 Message Broker 的队列管理器上运行指定脚本 MQJMS_PSQ.mqsc，以生成必需的 JMS 队列。mqsc 脚本文件在 WebSphere MQ 安装路径的 java/bin 子目录下，运行命令：`runmqsc WBRK61_DEFAULT_QUEUE_MANAGER < MQJMS_PSQ.mqsc`

配置 WebSphere MQ 为 JMS Provider

在 monitor server 的控制台中做以下配置

1. 定义主题连接工厂：

在 monitor server 的控制台中，定义 JMS 主题连接工厂，选择 WebSphere MQ message provider 类型。

输入名称和 JNDI 名称，例如 topicConn 和 `.jms/topicConn`。

Hostname 输入运行 WebSphere Message Broker 队列管理器的主机名或 IP 地址。

Port 输入运行 WebSphere Message Broker 队列管理器定义的 TCP 侦听器端口号。

Transport Type 如果队列管理和 monitor server 在同一节点上，使用 BINDINGS，否则使用 CLIENT。

Channel 输入定义的 Server Connection channel

Queue manager 输入 Message Broker 队列管理器名

Broker control queue 输入 `SYSTEM.BROKER.`

`CONTROL.QUEUE`

Broker Queue Manager 输入 Message Broker 队列管理器名

其他使用缺省值。

在 Broker version 下的 Advanced 部分，选择 BROKER 作为 Broker message。

2. 定义主题：

在 monitor server 的控制台中，定义 JMS 主题，选择 WebSphere MQ messaging provider 类型。

输入名称和 JNDI 名称，例如 topicGP 和 `and.jms/topicGP`

Base Topic Name 输入监控主题名，例如 `$/SYS/Broker/WBRK61_DEFAULT_BROKER/Monitoring/WBMonitorEventsExecGroup/GeneratePN_ESB_Flow`，其中 WBRK61_DEFAULT_BROKER 是 Message Broker 代理的名称，WBMonitorEventsExecGroup 是 Message Broker 执行组的名称，GeneratePN_ESB_Flow 是要监控的消息流的名称。

3. 安装 MDB 应用程序：

在 monitor server 的控制台中，安装 WMBEventEmitterEAR.ear，在安装过程中，选择显示所有的安装项和参数，注意 Bind listeners for message driven bean，记录其中 listener port 名称。

如果 monitor server 启用了安全性，那么需要对 WMBEventEmitterEAR 做以下设置：

设置 WMBEventEmitterEAR 的 Security role to user/group mapping，把 Administrative user 映射为 eventEmitter role，对 eventEmitter role 选择所有认证用户。

设置 WMBEventEmitterEAR 的 User RunAs roles 为 Administrative user

4. 生成 MDB 的侦听器：

在 monitor server 的控制台中，选择 应用服务器 > Communications 的 Messaging > Message Listener Service > Listener Ports。

定义第 3 步中声明的监听端口：

Connection factory JNDI name 输入定义主题连接工厂时使用的 JNDI 名称: jms/topicConn

Destination JNDI name 输入定义主题时使用的 JNDI 名称: jms/topicGP

重启服务器, 确保监听端口以及 WMBEventEmitterEAR 应用程序在运行。

在 Message Broker 环境中检查对主题的订阅是否成功

在 Message Broker Toolkit 中, 切换到 Broker Administration 透视图, 在域面板中打开所发布的 Broker (代理) 的 Subscriptions (订阅) 页签, 执行查询操作。如果以上步骤成功应该可以看到对于主题的订阅 \$SYS/Broker/WBRK61_DEFAULT_BROKER/Monitoring/WBMonitorEventsExecGroup/GeneratePN_ESB_Flow

在 Message Broker 消息流中配置事件源

Monitor 的业务监控是基于事件机制的监控, 从 Message Broker 6.1.0.2 之后, 就引入了对事件机制的支持, 随着版本的更新与增强, 消息流监控的方法也有不同, 但是总的来说是越来越方便, 越来越简单。

使用 Message Broker toolkit 中的消息流编辑器配置监视设置。可以设置的内容包括发送哪些事件以及事件正文应提供什么内容。当消息流部署到 Broker 上之后, 运行的消息流从活动的事件源中自动发送事件到指定的位置。由于 message-driven bean (MDB) 的配置, 这一位置会被 monitor server 所监听, 其中的内容也会被 monitor server 捕获并处理。图 4 显示了作为消息流编辑器一部分的监视属性的示例配置。

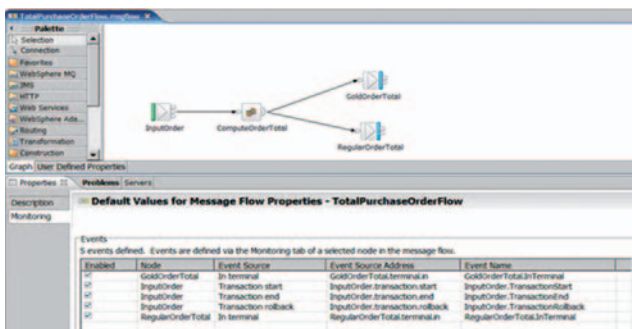


图 4. 消息流编辑器 - 监视配置

Message Broker 增强了从消息体中捕获复杂的数据, 并将其插入到发送的业务事件中的能力。用 XPath 表达要捕获的数据。如图 5 所示, Message Broker Toolkit 中提供的 XPath 表达式生成器可以帮助以图形化的方式编写这些表达式。

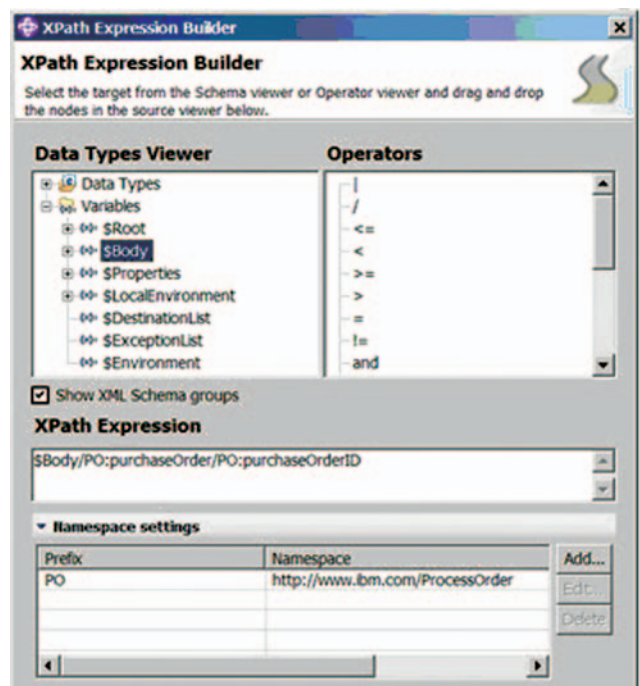


图 5. 使用 Message Broker XPath 表达式编辑器指定事件内容

定义了发送事件的配置后, 从 Message Broker Toolkit 导出消息集的 XSD 模式文件。然后将这些模式文件导入到 Monitor 开发环境, 用于创建使用 Message Broker 事件的监视模型。

启动对发布的消息流的监控

在成功部署消息流后, 启动 Broker 对消息流的监控功能。
`mqschange flow monitoring WBRK61_DEFAULT_BROKER -e WBMonitorEventsExecGroup -f GeneratePN_ESB_Flow -c active`

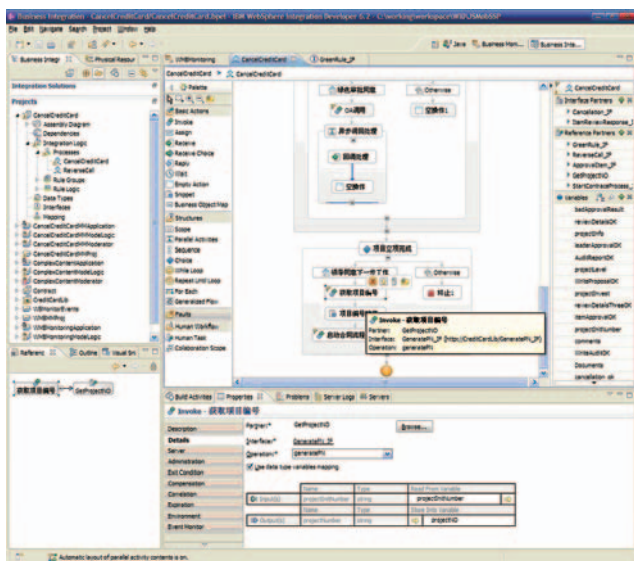
小结

通过以上的步骤, Message Broker 可以按照指定的格式发出消息, 被 monitor 捕获并使用。在 Monitor 开发环境中,

可以根据收到的事件格式，开发相应的 monitor model，以任何期望的方式展示监控的内容。

在 Webphere Process Server 和 Message Broker 的混合环境中进行端到端的监控

作为BPM方案的一部分，monitor具备流程运行监控的能力，而流程运行平台 Websphere Process Server 所调用的自动流程可能以Message Broker的方式来实现。因此整个BPM方案是一个WebSphere Process Server + Message Broker的混合环境，由于monitor具备了对Message Broker的监控能力，因此在这样的环境中可以实现一个完整的端到端的监控，从流程环节到自动化流程内部。图6是在这样一个混合环境中的流程实现，其中的获取项目编号节点是一个调用Message Broker消息流的自动化节点。



在 monitor 开发环境中，分别对 WPS 和 Message Broker 创建 monitor model 和监控上下文，按照“使用 WebSphere Business Monitor 监控 Message Broker 流程”一节中的方法对 monitor 和 Message Broker 进行设置，并最终完成 monitor model 的开发。将开发好的所有监控程序部署到 monitor server 上；WPS 应用程序部署到 process server 上；Message Broker 消息流部署到 Broker 上。

使用 WebSphere Business Monitor dashboard 查看监控的效果

打开 Business Space，创建业务监控面板。根据开发过的 monitor model，选择监控instance、KPI、diagram、dimension 等等。在这样一个混合环境中，监控的事件既可以来自于WPS，也可以来自于Message Broker。这就使得我们可以在同一个监控界面上看到所有的监控内容。图7反映了监控的效果。

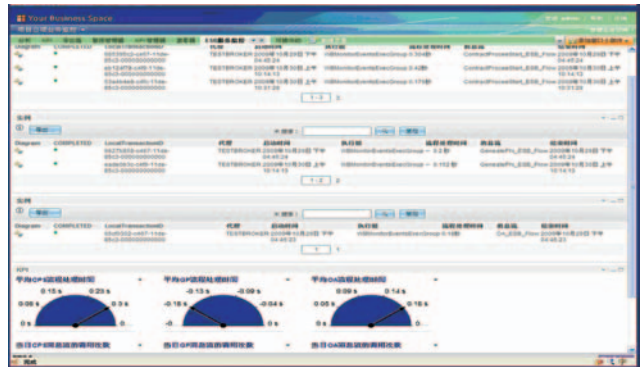


图7. Message Broker 监控效果—实例和KPI

WMBMonitorEmitter ear 包下载：



总结

WebSphere Business Monitor 提供了用于监视业务流程和活动的功能。覆盖了不仅仅局限于 BPM 产品的范围。所有基于 CEI 机制的应用系统的业务活动，都可以得到监控。本文对在 WebSphere Process Server 和 Message Broker 的混合环境中进行端到端的全业务流程监控提出了具体的实现方法，尤其侧重于实现对 Message Broker 的监控。

致谢

作者非常感谢华东WebSphere 售前技术支持团队和Monitor SWAT 团队的大力帮助，没有他们，这篇文章不可能完成。特别感谢以下团队负责人：Dong Chun、Ying Li Zhu、Li Ke Jia、Wei Jing Jing、Zhou Qi、Huang Yan Fang。■

Websphere 性能优化之页面静态方案

戴科冕 / 文

通过 IBM Http Server (下文称为 IHS) 静态页面分离技术与利用 WebSphere Application Server (下文称为 WAS) 的动态高速缓存 (Dynamic Cache) 技术优化访问页面性能。客户端访问静态页面无须与后端数据交互, 减少了容器与数据源的压力, 从而达到优化系统的性能。

IBM Http Server 不但可以作为 WAS 的前一级 Http 请求队列的缓冲, 减轻 WAS 的压力, 还可以利用其静态页面分离方案减轻部分压力。在系统的部署或者构建的时候, 我们可以把一些非重要的静态页面放在 IHS 中, 当客户端发送请求访问该页面资源时, 由 IHS 直接响应请求并将静态页面资源返回客户端, 无须再经过 WAS, 从而达到减轻达到 WAS 负担的目的。

除了 IHS 静态页面分离方案, 另外还可以利用 WAS 的动态高速缓存技术, 将 J2EE 应用中静态的 (Html、Flash、Css、Js、Jpg、Gif 图片等文件) 和动态的 (需要与数据库、Web Service 等服务交互才能得到的数据) 内容缓存到应用服务器的 JVM 中 (如: Jsp、Servlet、*.do 等请求)。在访问相关被缓存的资源时, 所有的相关输出都可以直接从 JVM 的内容中获得, 而无须再与数据库、Web Service、静态文件等交互, 从而有效地提高系统的性能。

IHS 分离静态内容方案

整个操作过程主要分 3 步完成

- * 关闭 Was 的 File Serving Servlet 服务
- * 重新生成插件 (Plugin-cfg.xml)
- * 修改 IHS 配置文件并拷贝静态文件到指定的目录

关闭 Was 的 File Serving Servlet 服务

1) 在 Was 中部署的 Web 应用中分别找到 2 个在 /WEB-INF/ 目录下面的 ibm-web-ext.xmi 文件。



参考路径:

> IBM/WebSphere/AppServer/profiles/节点/conf/cell/applications/应用包名/deployments/应用包名/应用包名/WEB-INF/目录下

> IBM/WebSphere/AppServer/profiles/节点/应用安装部署目录/应用包名/应用包名/WEB-INF/目录下

2) 打开 ibm-web-ext.xmi 文件, 找到 fileServingEnabled 属性项, 把它设为 false, 默认是 true。找不到该属性项可直接手工添加 (fileServingEnabled="false")。

如:

```
<xml version="1.0" encoding="UTF-8"
<com.ibm.ejs.models.base.extensions.webappext:
WebAppExtension
  xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XML" xmlns:com:
  ibm.ejs.models.base.extensions.webappext="webappext.
  xmi" xmi:id="WebAppExtension_1260496854906"
  fileServingEnabled="false">
  <webApp href="WEB-INF/web.
  xml#WebApp_1260496854890" />
  <jspAttributes xmi:id="JSPAttribute_1260496854906"
  name="reloadEnabled" value="true" />
  <jspAttributes xmi:id="JSPAttribute_1260496854907"
  name="reloadInterval" value="10" />
</com.ibm.ejs.models.base.extensions.webappext:
WebAppExtension>
```

3) 重启 Was, 重启后校验配置是否生效。访问 Web 应用中的静态文件, 如会报 404 错误的, 这是正常现象, 表示 Was 中的 File Serving servlet 服务功能已经被关闭了。

重新生成插件(Plugin-cfg.xml)

> 登陆 Was 控制台, 环境 ->更新全局 Web 服务器插件配置 ->确定,重新生成插件。

> 服务器 ->Web 服务器 ->选择相应的 web 服务器 ->生产插件 ->传播插件

> 检查配置文件是否正取。

打开IHS/Plugins/conf/web服务名/plugin-cfg.xml. 找到 UriGroup 项, 检查其配置。如 Uri 中的配置还是指定为 Name= “/*”, 须注释该行配置, 否则 IHS 会将所有的请求都返回给 WAS 处理, 那么所有访问静态内容都会报 404 错误。可通过手工配置相应无须过滤的 Uri。

具体参考配置如下:

```
<UriGroup Name= "default_host_kpiCluster_URIs">
<!--<Uri AffinityCookie= "JSESSIONID"
AffinityURLIdentifier= "jsessionid" Name= "/*" /> -->
<Uri AffinityCookie= "JSESSIONID"
AffinityURLIdentifier= "jsessionid" Name= "/*.do" />
<Uri AffinityCookie= "JSESSIONID"
AffinityURLIdentifier= "jsessionid" Name= "/*.jsp" />
<Uri AffinityCookie= "JSESSIONID"
AffinityURLIdentifier= "jsessionid" Name= "/
j_security_check" />
<Uri AffinityCookie= "JSESSIONID"
AffinityURLIdentifier= "jsessionid" Name= "/
ibm_security_logout" />
</UriGroup>
```

修改 IHS 配置文件并拷贝静态文件到指定的目录

1) 在 IHS 的默认路径, 打开 \IBM HTTP Server\conf\http.conf 文件, 找到如下配置:

```
#
# DocumentRoot: The directory out of which you will
serve your
# documents. By default, all requests are taken from
this directory, but
```

```
# symbolic links and aliases may be used to point to
other locations.
```

```
#
DocumentRoot "D:\IBM\HTTPServer\htdocs/en_US"
#
# This should be changed to whatever you set
DocumentRoot to.
```

```
#
<Directory "D:\IBM\HTTPServer\htdocs/en_US">
可将 D:\IBM\HTTPServer\htdocs/en_US 设置为新的
的目录, 自定义新的静态页面目录。
```

我们将非重要的静态文件放到 E:\PageFiles\2009-12-11\ 目录下, 配置如下:

```
DocumentRoot "E:\PageFiles\2009-12-11"
<Directory "E:\PageFiles\2009-12-11">
2) 把应用中的静态文件拷到 上述在 Httpd.conf 配置文
件定义好的路径中去。
```

最后可根据上面的配置, 将静态文件都是从 Was 应用中拷贝到 E:\PageFiles\2009-12-11\ 目录中。

3) 重新启动 IHS

通过页面再次访问静态页面, 此时不会再出现 404 错误了, IHS 接到请求访问静态页面内容时, 会直接读取 E:\PageFile\2009-12-11\ 返回页面请求, 通过该配置可以进一步减轻 Was 的压力。

启用 WAS 动态高速缓存

启用动态高速缓存服务同样也分为 3 步

- 在 WAS 控制台上启用动态高速缓存服务
- 为应用服务器配置 Servlet 高速缓存
- cachespec.xml 配置文件

在 WAS 控制台上启用动态高速缓存服务:

- 1) 登录 Was 的管理控制台
- 2) 在管理控制台中, 启用“服务器->应用程序服务器->Server1->容器服务->动态高速缓存服务”
(勾选 ‘在服务器启动时启用服务’)
- 3) 确定并保存配置

为应用服务器配置 Servlet 高速缓存:

1) 在管理控制台中,启用“服务器->应用程序服务器->Server1->Web 容器设置->Web 容器->启用 servlet 高速缓存

(勾选‘启用 servlet 高速缓存’)

2) 确定并保存配置

cachespec.xml 配置文件

1) 在应用目录下 WEB-INF/下增加cachespec.xml 配置文件, 通过该文件指定缓存对象。

内容如下: 注意红色部分, 要正确指定 cachespec.dtd 文件位置, 一般情况下是在 /IBM/WebSphere/Appserver/properties 目录下。

```
<?xml version= "1.0"?>
<!DOCTYPE cache SYSTEM "D:\IBM\WebSphere\AppServer\properties\cachespec.dtd">
<cache>
  <cache-entry>
    <class>servlet</class>
    <name>index.jsp</name>
    <cache-id>
      <component id= "*" type= "parameter">
        <required>false</required>
      </component>
      <component id= "" type= "pathinfo">
        <required>false</required>
      </component>
      <component id= "host" type= "header">
        <required>false</required>
      </component>
      <timeout>180</timeout>
    </cache-id>
  </cache-entry>
</cache>
```

2) 在完成前三步后, 在服务器上安装 Cache Monitor 组

件, 来监控刚才配置的动态高速缓存 (Dynamic Cache) 是否成功运行。

发布 IBM/WebSphere/AppServer/installableApps/CacheMonitor.ear, 并且将该应用与要业务应用部署在用一个 Server 下。

3) 部署完毕后, 重启整个 WAS。

4) 测试校验部署是否生效。

分别访问:

> http://127.0.0.1/index.jsp

> http://127.0.0.1/index.jsp123

通过访问 http://127.0.0.1/cachemonitor 就可以看到 index.jsp 已进行了缓存

如图所示:



以上的两种技术一种是基于 WEB 服务器一级别的优化, 另一种则是基于 Websphere 应用服务器上的优化, 合理的使用两种技术, 可更有效的帮助应用系统提高性能。

虽然在 WAS V6 以上的版本中, Web Container 对于 Http 请求采用了非阻塞 I/O 方式, 使得 IHS 的这一功能作用已经非常弱化, 但在大并发访问的情况下, 还是建议大家使用 IHS+WAS 架构, 使用 IHS 缓冲客户端请求与并且利用其得静态页面技术, 直接返回静态资源, 减轻应用服务器压力; 开启 Was 动态高速缓存, 缓存一些变动较少的 Jsp、Servlet 等资源, 进一步有效的优化系统的性能。分别将 IHS 与 WAS 部署到两台物理主机上, 可以让在系统在性能扩展上更加灵活, 利用垂直扩展优化可更有效的提高系统的性能。■

主持人：北京传诚科技有限公司 CEO 晓黎

嘉宾：南京浦口信息中心副主任 史小华
IBM 软件集团大中华区总工程师 寇卫东 博士
国研科技有限公司董事 周宁

SmartWork 三人行： 南京浦口区政府数据交换项目经验交流

主持人：小华主任，可否请您谈一谈南京浦口政府有什么样的数据交换业务需求？数据交换在数字浦口中的重要性在哪里？



史小华：主持人好！在回答这个问题前，我想首先介绍一下“数字浦口”的基本情况。2006年为了对接南京市“一城三区”城市发展思路和城市总体规划中对全区提出的“建设现代化的南京江北新市区”功能定位要求，实现浦口区“十一五”规划所提出的战略目标，全面启动和加快推进浦口区的信息化建设，加速推进浦口区国民经济和社会信息化的发展，在区委、区政府的直接领导下，制定了“数字浦口”第一个三年规划（2006年至2008年），在这个规划中明确了“数字浦口”的技术特征是：采用面向服务的（SOA）总体框架，通过信息交换平台、目录服务平台及公共应用支撑服务平台，实现各个应用系统间互联、互通和互操作；对于分散、异构的城市信息资源体系（如：信息资源、服务资源、流程资源和管理资源），在兼顾资源现有配置与管理状况的条件下，实现无缝整合、有效利用和最大增值。实现资源、应用、服务、安全的一体化管理。在规划引领下，经过三年多的建设，基本实现了预期的规划目标。

在当时的规划中，除了提出“数字浦口”的技术特征外，同时也提出了“数字浦口”的服务特征：即以面向城市的公众服务为建设的重点，将政府服务职能信息化和社会公共服务信息化放在突出的地位。广泛开展公益性信息服务，提升社会公共信息资源的普遍服务能力，消除“数字鸿沟”给城市发展所造成的障碍。通过“数字浦口”的服务特征体现，使浦口区政府、企业、农业、商务和社会公众真正享受到数字化城市建设所带来的便利服务。我想对服务特征的定义与理解是我们建设数据交换系统的总体业务需求。更具体地讲，就是我们在建设面向内部的管理系统与决策支持系统时，需要面向各个管理主题及决策主题通过数据交换系统提供的统一服务与管理功能，

实现对不同数据源的采集、传输与加工处理，在国研软件承建的SOA平台框构及其应用项目中，通过数据交换系统已实现了11个部门、1200个指标数据的处理。

另外一方面，根据浦口区新一轮建设“南京现代化江北新城”的发展目标，需要能够面向“南京现代化江北新城”的建设形成城市规划、城市建设、城市管理的一体化平台，同时在政府内部资源整合的基础上，提高面向浦口区各类企业、居民、投资者、旅游者的信息化服务水平，在这个过程中肯定需要进行政府内部资源与外部资源的数据交换，这些我想应该都是数据交换系统的建设源动力。

谈到数据交换系统的重要性，我想做一个形象的比喻，大家知道浦口区与南京主城区隔了一条长江，每天有大量的人、车、资源通过长江大桥、二桥、三桥及正在规划建设中的大桥、过江隧道、地铁南北往来，这些南北往来的人、车、资源构成了现实中的经济与社会生活。在“数字浦口”建设中，数据交换系统可以比作信息世界中的长江大桥、过江隧道及地铁。同时为了保证这些交通枢纽能够高效利用、方便管理，还要建立相应的调度、监控与管理体制。大家可以设想一下，如果大桥交通出了问题，会给现实生活带来什么麻烦，就能理解数据交换系统的重要性。

主持人：寇博士请您谈一谈：从技术方面，怎样的架构设计才能满足南京浦口的数据交换需求？南京浦口数据交换架构的特点有哪些？



寇卫东：小华主任刚刚讲了浦口的需求。这些需求主要是实现各个应用系统间互联、互通和互操作；无缝整合并且有效利用分散、异构的城市信息资源体系。广泛开展公益性信息服务，提升社会公共信息资源的普遍服务能力。在建设面向内部的管理系统

与决策支持系统时,需要面向各个管理主题及决策主题通过数据交换系统提供的统一服务与管理功能,实现对不同数据源的采集、传输与加工处理。

要满足这些需求就需要设计一个技术方案,它能够实现各个应用系统间互联、互通和互操作,无缝整合分散、异构的各种信息资源。从技术架构方面讲,因为需要整合异构系统,首先架构设计必须要考虑开放性、支持国际标准、可扩展性。第二点,架构设计也要考虑灵活的数据模型及其支持多种数据接口问题,支持多通信协议问题。第三点是架构要支持多种编程模式以及非标准应用的集成问题。第四点是并发、平衡负载和可靠性以及服务质量的控制,系统的安全控制及监控。把握好这四点,就能保障架构设计满足南京浦口的数据交换需求。

南京浦口数据交换架构的特点包括它基于 IBM 消息传递中间件技术及消息代理;它采用了国际标准业务消息的数据模型;它基于开放标准,整合性和集成性强;它包含数据库接口、文档接口、Web Services 等多种数据接口,最大地考虑非标准应用的集成;支持多种编程平台,可以由用户采用多种方法进行功能扩展;它是一个标准的并可扩展的系统构架;此架构考虑到了并发、平衡负载和可靠性问题。总之,它是基于 SOA 的架构,通过 IBM Websphere Message Broker 实现了 SOA 的服务总线 (ESB) 的落地。

从技术细节上说,在南京浦口数据交换架构中,通过安全可靠的数据传输,支持数据的持久性,即使宕机也不允许数据丢失。同时也支持数据的非持久,即使数据丢失也能通过重来完成。它也支持系统间的同步和异步通信。支持多种形式的路由,路由的实现可以是基于传输数据的内容,也可以是基于外部数据源定义,或者可以基于用户自己的算法规则。其支持的路由方式也可以有多种形式,如单向传递、请求/响应、发布/订阅、服务的分解与组装等。支持数据格式转换。此外,它也支持事件服务,包括事件的发现和发布,发布和订阅等。作为 SOA 的总线,它支持多通信协议,它的数据模型非常灵活,它也支持流程处理。服务质量的控制和系统的安全控制及监控也是它的总线功能的一部分。

主持人: 小华主任,很多厂商都号称有数据交换的技术平台和产品,当时你们基于什么样的考虑,选择了 IBM 为浦口作

为数据交换产品的提供商?



史小华: 刚才我谈了“数字浦口”的基本情况 & 数据交换系统的业务需求与重要性。通过三年多的建设,我们的理解是信息系统的核心价值在于应用,而应用的好坏在于规划的前瞻性及可行性,因此,浦口区的信息化建设非常注重规划驱动。主持人希望了解我们为什么选择 IBM 作为浦口区数据交换产品的提供商,我想主要是基于以下几个理由:第一、IBM 具有很强的咨询服务能力,数据交换系统作为一个技术平台,其建设的目标是应用,而这种应用是既要满足当前的需求,又要面向未来的,IBM 有能力在应用上给我们提出一些前瞻性的应用场景。第二、IBM 提出的“按需应变”的产品策略及作为产品标准的积极参与和推进者,产品标准的符合性及功能的完整性是 IBM 的优势。第三、数据交换系统的关键指标是数据交换的稳定性,而 MQ 在消息传输上的稳定性、效率在业界是事实上的标准。所以浦口区的数据交换系统是基于 MB/MQ 构建的,以上,应该是我们选择 IBM 产品作为构建数据交换系统的主要原因。

主持人: 寇博士可否请您谈一谈 IBM 在数据交换方面有哪些产品和成功案例。



寇卫东: 我想先将 IBM 在数据交换方面的产品给大家做个介绍。IBM 在数据交换方面的一系列产品:

Websphere MQ: 它是一款点到点的数据交换产品,该产品历史悠久,在很多国家都有成功应用。它能提供通用的信息传递主干,使用通用接口以使信息可以自由流通。允许信息在所有应用与技术服务之间流通,从而提供端到端事务处理能力,并提供更高的可用性、可伸缩性和安全性。在中国的政府、银行保险、电信、石油化工、汽车制造、交通运输、物流等等行业都有普遍和成功的应用。

Websphere ESB Server: 该产品主要是在 J2EE 架构之下及 SOA 架构之下的服务总线产品,主要实现服务之间路由消息传递的最佳化,在服务请求之间做协议和消息格式的转换,也能处理不同来源的业务事件,它是一个基于标准的集成平台,它具有很多优势,例如连接、使用比较简单,与其他 WebSphere 产品的整合也非常好,另外它可以支持国际标准。异构系统也可以通过适配器连接到 Websphere ESB server 实现的总线上。

Websphere Message Broker: 它具有在异构系统中支持数据交换的强大功能;它能够以智能的、实时的方式,从多种设备类型、业务单元和位置向几乎遍布政府或者企业(甚至政府或者企业之外)的全部系统和应用,路由和转换消息与数据。它能生成并部署新的业务程序,同步多个系统中的业务事件,整合不同平台上的应用,在应用之间转换信息格式并且实现路由。浦口区政府此次就是依靠该产品来实现数据交换。

Websphere DataPower XI50,它是新一代的硬件化的ESB实现数据交换产品,可以实现路由、数据转换、异构协议功能的桥接、中介四大功能。它为ESB用户提供了新的选择。作为硬件化的ESB, XI50的各种功能通过硬件、固件及部分软件协同完成,充分满足了ESB的功能需求,从而简化了ESB解决方案的优势,在很多应用场景中, XI50实现了“开箱即用”的ESB,用户仅需接通电源,为其配置并联通网络,并通过XI50提供的Web管理控制台配置所需的服务,即可完成ESB的构建工作。

IBM在数据交换方面有很多成功案例。比如,近几年,随着信息化进程的不断深入,江苏省政府的信息量急剧上升。江苏省政府为完善政府信息化,集成已有的政府机关内部信息应用,并将省政府和其他地市信息系统连成一体,提高办公效率,江苏省政府信息中心采用了IBM的电子政务解决方案。数据交换是这一方案的核心功能之一。又比如,国家卫生统计网络中在各个省级卫生统计网络中也包括数据交换功能,并且使用了IBM Websphere的MQ产品来实现数据交换功能。北京朝阳区数字城管项目以及朝阳区发改委项目也都包括了基于IBM数据交换方案的核心子系统。河南郑州新郑新区采用了Websphere MB/MQ构建物流平台的核心数据及服务交换体系。中国远洋集团、中国国际航空公司、陕西宝鸡市政府、山东烟台市政府、重庆高新区等等项目中的数据交换子系统都是基于IBM Websphere产品的数据交换方案的成功案例。

主持人: 小华主任,当时浦口基于什么样的考虑选择了国研软件作为项目的实施方?



史小华: 浦口区通过这几年的建设,一直在思考这样一个问题,信息中心的职能定位是什么,是不是只是传统意义上的技术支持部门。我们认识到,随着电子政务的建设从部门应用为主逐步发

展到以数据中心为主再过渡到以业务协同为主,这个过程中信息中心的职能在发生重大的改变。在以数据为中心与业务协同为主的电子政务建设阶段,信息中心所拥有的主要资源不再是基础设施与各个业务系统,而是存在于数据中心的数数据,信息中心的职能从系统建设和维护为主,逐步转变为数据管理、信息资源应用开发为主。这样,信息中心应该是一个面向数据或者信息资源开发的业务部门,而作为一个业务部门应该有属于自己的业务系统,不是一个开发平台。对于这样的一个理解,其实我们一直在寻找具有同样理念的合作伙伴,很幸运,国研软件的咨询和技术开发团队在这点上跟我们的理解非常一致,并且他们已经积累了在IBM产品上的开发政府决策型需求的经验,我想这是我们选择国研软件作为项目实施方的主要原因。当然,这个项目是采取公开招标的,评标专家对于国研软件投标方案中体现出来的业务理解能力和系统开发经验是非常认可的。

主持人: 周总请您分享一下国研软件在实现南京浦口区数据交换平台工作中,遇到了什么样的困难?您的团队是怎样克服这些困难的?



周宁: 正如浦口区领导所说的,对数据交换系统的准确定位是浦口区选择我们作为实施方的主要依据。但是,为了实现这样一个定位,其实还是有相当大的难度的,大家都知道,IBM的产品其实是与业务无关的,这样产品的通用性就能得到保障,而作为一个数据交换的业务系统,其实我们需要做的是和IBM专家一起分析政府行业的各种数据交换业务需求,包括各种数据源的形式、数据格式、数据大小、加密的要求、数据交换基于业务视角的流程编排、业务数据的应用模式等等,我们需要在建立数据交换业务模型的基础上,通过调用IBM的MB/MQ产品提供的API封装一层业务逻辑,本质上是在数据交换系统运行时注入业务规则,实现一次开发、一次部署,屏蔽IBM产品的底层开发环境,实现新增数据交换需求的“零”编码方式。在这个过程中,既有业务抽象的困难,也需要对IBM产品进行深入的了解,非常幸运的是浦口区用户在业务上给了我们大力的支持,IBM在系统的架构以及产品的接口方面也给我们提供了很多支持和资源保障,这些都是项目成功的最主要原因。

主持人: 小华主任请您总结一下南京浦口数据交换平台的成功经验有哪些?



史小华: 主持人, 一个项目的成功其实需要很多的努力, 通过数据交换平台项目的实施, 我认为主要在于以下几个方面: 一、作为系统建设方一定要建立业务导向的视角, 而不是技术导向的视角, 从业务需求和交换系统的定位把握系统的目标。对于数据交换系统而言, 其实通过IBM产品本身提供的接口与包含在产品内的工具, 是可以实现交换的需求, 但是这样的实现需要我们信息中心的人员完全掌握MB/MQ的开发过程, 并且具备很强的开发能力, 而且这样的开发结果还很难提供一体化的业务流程管理视图。而业务导向的视角其实说明一点就是建设方要具备换位思考能力, 要具有很强的业务需求分析能力及交流能力。二、我们说重视业务需求导向, 并不是说不重视产品功能, 对于数据交换系统这样一个应用环境比较复杂的系统, 选择好的产品是至关重要的, IBM的MB/MQ产品恰好具有这样特征。三、有了好的业务规划、有了好的产品, 还需要好的架构设计, 还需要建设方具备把这两者结合在一起的能力。这里面系统承建方和作为产品原厂商的IBM的支持就非常重要, 说到底IBM产品的价值最终是通过信息中心的工作体现出来的。

主持人: 周总请您谈一谈国研软件在南京浦口数据交换平台中与IBM团队的合作情况。



周宁: 作为系统的承建方除了要有对业务需求的准确理解外, 还需要对产品的深度了解。南京浦口数据交换平台的成功实施是基于国研软件的实施团队长期与IBM合作的基础, 在项目实施的过程中, 国研软件的实施团队在项目定义阶段、需求分析阶段、架构设计阶段、POC验证阶段、功能开发阶段及项目评估阶段与IBM的团队进行了紧密的合作, IBM在培训、资料、技术设计及测试环境等多方面给予了充分的支持。另外, 2009年6月IBM专门针对国研软件在南京的客户安排了一次基于IBM的MB/MQ产品的专题培训, 让客户对IBM产品及最终的业务功能之间的联系与区别能有一个感性的认识。在此, 也要对浦口区及IBM表示感谢!

主持人: 寇博士您对与国研合作以及南京浦口数据交换平台的成功经验两方面可否有补充?



寇卫东: 我希望将我们的感受给大家分享一下。南京浦口数据中心数据交换成功的经验主要有以下几点:

第一是领导重视。特别浦口区领导的重视, 从项目的前期规划到具体设计实施工程中, 我个人就曾经多次与戚鲁书记和其他浦口区领导见面、交流、策划; 戚鲁书记对IBM的建议给予了充分的重视, 多次召集浦口区领导集体听取IBM和合作伙伴的汇报。IBM领导对浦口这个项目也非常的重视, IBM团队能够全力投入到这个项目当中。国研领导也就是周总在整个项目实施过程中起到了至关重要的作用。

第二就是合作伙伴的协作。IBM与国研的密切协作, 刚才周总已经介绍过, 我就不多讲了。

第三就是明确需求。IBM团队与国研团队一起, 深入了解了浦口区政府不同部门的业务需求。

第四是IBM内部协同协作。大家都知道IBM是一个很大的公司, 而任何一个大公司内部的协同协作总是一个挑战。这一次, 我们得到了IBM领导、技术团队和销售团队的强有力的支持, 保证了项目设计、POC、以及成功实施。

第五是IBM Websphere世界领先的架构思想和产品也是能够保证项目成功的关键。

小华主任刚才讲到, 目前南京浦口数据交换系统已成功实现了11个部门、1200个指标数据的处理。这又一次证实了IBM Websphere的强大功能。

我非常高兴今天与各位领导一起探讨这方面的成功经验, 非常感谢! ■





J2EE 应用性能问题分析

泊涯 / 文

分析 J2EE 应用程序的性能问题是一项艰辛的任务。性能测试分析人员的责任是确保应用程序具备优越的性能且可持续运行的程序。在出现性能问题时能够准确判断性能问题点。下面是本人多年在做性能测试分析过程中, 针对引起性能问题的根本原因, 提出如何准确诊断并迅速排除性能问题的建议方法, 以提供参考。

1.1 OutOfMemoryError 之程序内部问题

1.1.1 问题描述:

- 1、系统性能随着时间的增加逐渐下降, 系统性能随负载的增加逐渐下降
- 2、可用内存逐渐减少
- 3、前台页面响应时间逐渐增大。
- 4、后台日志出现内存溢出信息等现象。

1.1.2 问题诊断:

1、内存使用率随时间或负载的增加呈线性增长, 重启后系统可恢复正常, 但是过一段时间问题又重现, 而且内存泄漏呈双倍增长, 导致系统内存消耗随时间呈指数曲线变化。

2、通过 `headdump` 下的日志用 `HeapAnalyzer` 进行分析诊断。

3、通过各种日志分析工具对日志进行分析诊断, 例如 `g a` 工具分析 `native_stderr.log Gc` 垃圾日志。

4、通过 `Jprofile` 进行性能诊断分析。

1.1.3 问题分析:

- 1、在处理字符串时, 使用 `StringBuffer` 而不使用字符串连接
- 2、基本变量类型不使用对象内类, 例如使用 `INT` 不使用 `INTEGER`
- 3、减少 `NullPointerException`。

4、对象引用完没释放

2.1 OutOfMemoryError 之与数据库交互问题

2.1.1 问题描述:

- 1、系统性能随着时间的增加逐渐下降, 并出现死锁,
- 2、可用内存逐渐减少
- 3、前台页面响应时间逐渐增大。
- 4、后台日志出现内存溢出信息。

2.1.2 问题诊断:

1、通过 `headdump` 下的日志用 `HeapAnalyzer` 进行分析诊断。

2、通过各种日志分析工具对日志进行分析诊断, 例如 `g a` 工具分析 `native_stderr.log Gc` 垃圾日志。

3、通过 `Jprofile` 进行性能诊断分析。

2.1.3 问题分析:

- 1、JDBC 语句出现资源泄漏, 引发 Java 桥接层和后台系统出现严重性能问题。
- 2、通常是由于遗漏了 `Finally` 模块, 或者只是没有用 `close` 函数关闭代表外部资源的对象。

3、JDBC 连接数不足

4、是数据库问题引起

3.1 OutOfMemoryError 之

与系统交互问题

3.1.1 问题描述:

- 1、可用内存逐渐减少
- 2、前台页面响应时间逐渐增大。
- 3、后台日志出现内存溢出信息。

3.1.2 问题诊断:

1、通过 `headdump` 下的日志用 `HeapAnalyzer` 进行分析诊断。

2、通过各种日志分析工具对日志进行分析诊断, 例如 `g a` 工具分析 `native_stderr.log Gc` 垃圾日志。

3、通过 `Jprofile` 进行性能诊断分析。

3.1.3 问题分析:

1、由于 JVM 参数设置大小问题导致, 通过测试修改 JVM 大小合适值。

2、物理内存不足。

4.1 高 CPU 使用

4.1.1 问题描述:

- 1、系统性能下降, CPU 使用率很高。
- 2、IO 使用率很高
- 3、客户前端响应时间很慢

4.1.2 问题诊断:

1、进行 `cord dump` 下日志在通过 `JCA` 进行日志分析问题。

2、查看服务器下的各种日志通过各种日志分析工具对日志进行分析诊断。

3、通过 JPROFILE 进行分析

4.1.3 问题分析:

- 1、死循环的编码缺陷导致可预见的死锁，需要对循环进行拆分等优化。
- 2、线程在 while 语句返回值是真的情况下发生阻塞，将最终演变成为 CPU 使用率过高，同时导致内存使用率过高，磁盘

IO 使用率高等问题

总结:

如果在出现性能问题时根据以上应用分析的条款进行排查是否是应用程序代码引起性能问题，如果不是而且应用服务器监控到的内存使用率或者 CPU 使用率指标

达到性能要求指标，但是整体性能例如响应时间之类的还无法达到预期的要求，此时可以查看 WEB 服务器的配置、应用服务器中间件参数调整、操作系统参数调整以及数据库参数调整是否有问题或者 JDK 版本不是最新版本需要升级等问题导致的性能问题。■

小型机设置 WebSphere 的默认编码全解密

本文整理自 WebSpherechina 网站博客

当安装了 WebSphere 的小型机的默认编码不是 GBK，而应用程序是以 GBK 作为默认编码会导致一系列的乱码问题。

在这个情况下，我们可以通过 WebSphere 的控制台设置来修正该问题。以下 WebSphere 简称 WS，WS 有 WS5 和 WS6，两个有些不同。

WS5:

点开“服务器”节点 --> 应用程序服务器，找到自己应用使用的服务器。

进入，在“其他属性”中找到“进程定义”，

然后在其下的“其他属性”中找到 Java 虚拟机。

然后找到“一般 JVM 自变量”将其值设置为：-Dfile.encoding=GBK

然后保存，重新启动 WS 服务。注意是将 WS 重启。

telnet 连到服务器，stopServer.sh server1

WS6:

“服务器” --> 应用程序服务器，找到自己应用使用的服务器。

WS6 有不同，注意。

找到“服务器基础结构”，展开其下的“Java 和进程管理”，然后进入“进程定义”。

进入后找到“其它属性” --> “Java 虚拟机”：

将“Java 虚拟机”设置为：-Dfile.encoding=GBK

另：在 UNIX 下直接输入命令：

./startServer.sh server1 -Dfile.encoding=GBK 也可。■

关于 MB 集群的构架心得

卢汉利 / 文

经过一年多对 Websphere Message Broker (WMB) 的研究探讨，初步了解了构建 WMB 集群的基本框架：

1、构建 WMO 集群

MB 是运行在 MQ 的基础上的，因此，要实现 MB 集群，首先必须实现 MQ 的集群，建立 MQ 集群有下面几种方式：

(1) 通过操作系统集群构建 HA 架构，例如采用 WINDOWS cluster, AIX HACMP 等，这种结构必须采用共享存储方式实现，然后在 HA 结构上建立 MQ cluster，其有点是通过操作系统实现故障接管，MQ cluster 本身具有负载均衡功能。

(2) 直接在双机（多机上）建立 MQ cluster，通过负载均衡设备（例如 F5）实现故障接管，MQ cluster 本身具有负载均衡功能。其缺点是硬件成本较高

(3) 直接在双机（多机上）建立 MQ cluster，其中有一台作为数据路由 MQ，其有点是结构简单，缺点是数据路由 MQ 为单点故障

2、构建 WMB 集群

在 WMO 集群每个节点创建 WMB 的 ConfigureManager 和 Broker 组件，上述组件使用的 mq 必须是集群的

分别在 WMO 集群每个节点的 Broker 组件部署执行组 和消息流、消息集。■

一、集群环境 WAS 负载均衡问题



IHS 应该起分发作用，但我发现当集群内一个 server 当了，客户端通过 IHS 访问应用时，老试着去访问当掉的那个 server，这是为什么呢，如何解决啊？

答

在集群环境中设置了 HTTP 插件负责负载均衡后，在有些情况下当某个集群成员失效时，插件不能及时地实现失效接管或者根本无法实现失效接管。在多数情况下，造成这种情况的原因是对 HTTP 插件工作原理的误解或者不恰当的插件配置。另外，Web 服务器是单线程还是多线程也会影响到失效接管的效果。

下面的内容主要是为了帮助用户理解：1，HTTP 插件的失效接管功能是如何利用某些性能调节参数；2，给用户一些关于这些参数的合理建议值。

注意：下面的内容是是基于 IBM HTTP Server（以下简称 IHS）完成的。但是，这些信息对于插件支持的常用 Web 服务器（例如：IIS，SunOne，Domino 等）也是有效的。

在 WAS 的集群环境中，当 HTTP 插件无法将请求发送给某一个集群成员的时候，HTTP 插件有提供失效接管的能力。在默认情况下，插件判断某个集群成员宕机并启用失效接管功能将客户端请求发往其他可用服务器的条件有：

- HTTP 插件无法和集群成员的传输端口建立连接。

- HTTP 插件监测到一个刚刚建立的连接在活动期间被集群成员过早地关闭掉。

在 HTTP 插件的配置文件 plugin-cfg.xml 中有一些可调整的参数。通过调整这些参数能够改变 HTTP 插件判断一个集群成员宕机并启用失效接管功能的快慢。

二、在生产环境上，WAS 一般需要配置的参数



在保障应用能运行的基础上，还需要设置什么参数，如 jvm、web 连接池、数据库连接池。可否罗列一下？

答

参数指标参数所在位置 线程池(WebContainer)最小值服务器 > 应用程序服务器 > 其他属性 > 线程池 > default 最大值线程不活动时间会话管理会话超时应用程序服务器 > server1 > 会话管理 Java 虚拟机初始堆大小应用程序服务器 > Server1 > Java 进程管理 > 进程定义 > Java 虚拟机最大堆大小禁用 JTIWEB 容器 Servlet 高速缓存服务器 > 应用程序服务器 > Server1 > Web 容器设置 > web 容器 JDBC 连接池连接超时 JDBC 提供者 > Oracle JDBC Driver > 数据源 > 连接池属性最大连接数最小连接数收集时间未使用的超时时效超时还有一些操作系统的参数，这些都能在 WAS 的信息中心找到，不同平台需要调整的参数也不一样。调整这些参数只是起到辅助作用，如果想要应用系统正常的运行，关键还是看应用程序的代码质量。◆

三、WebSphere6.1

正常使用一年后的莫名问题



websphere6.1 使用一年后，控制台中的应用服务器，应用程序的状态都显示不可用了，现在只能从后台进行服务器的启用和停止，没法通过控制进行操作。操作系统是 solaris 后台系统日志为：SSL 握手故障：已从目标主机：端口“10.67.2.159:8881”发送主题 DN 为“CN=CWServer-E，O=IBM，C=US”的签署者。可能需要将该签署者添加到本地信任库“/opt/WebSphere6/AppServer/profiles/Dmgr01/config/cells/CWServer-ECe1101/trust.p12”，它位于从 SSL 配置文件“security.xml”装入的 SSL 配置别名“CellDefaultSSLSettings”中。来自 SSL 握手异常的扩展错误消息为“No trusted certificate found”。◆

答

在管理控制台中找到旧的证书，点击“创建自签署证书”，其中公共名填写主机名，有效期可以适当填写长一些（例如 10 年），组织名可以写 IBM 或者您的公司缩写。创建完毕，在证书列表中，选择旧的证书，然后点击“替换”，将旧的证书换为新的证书确定，保存，同步注销管理控制台。◆

四、cluster 中权重值修改后，怎样才能生效？



cluster 中权重值修改后，怎样才可以使权重值生效呢？

答

改了权重值后，只用重启对应的应用实例，再更新下 web 插件就可以了：
1：配置新的权重信息；
2：更新 web 插件，并传播插件；
3：重启相应的应用服务器。◆

五、请问一下，websphere nd 6.0 中如何实现 session 复制?



我做了两台机器的集群，现在这两台机器的 session 不同步。

A 机

应用程序服务器 > A> 会话管理 > 分布式环境设置 > 内存到内存复制；

复制域: abcluster;

复制方式: 我选择的是客户机和服务器。

B 机

应用程序服务器 > B> 会话管理 > 分布式环境设置 > 内存到内存复制；

复制域: abcluster ;

复制方式: 我选择的是客户机和服务器。

设置完后，在同一浏览器中输入 A 机器的页面链接地址，在程序中获取的当前 Session ID 值，浏览器窗口不关，直接输入 B 机器的页面链接地址，这两次显示的 Session Id 值不一样。我的程序在 weblogic 的集群中运行，采用同样的方法时，它们的 Session ID 值是不变的！请教了！

答

Session 的复制只是把 A 机器 Session 的对象拷贝到 B 机器上，但并不激活这些对象。只有当 A 机器停止的情况下，Cluster 才会激活 B 机器上的 Session 对象。

例如：当页面访问请求发送到 IHS，IHS 通过插件向 WAS 转发，而插件是支持 Session 的亲缘性。也就是说当你的 session 是在 A 机器上生成的，那么当第二次请求带着 sessionID 回来的时候，插件还会把他们分发到 A 机器上。如果 A 停机了，插件将会从群集中选择其它的服务器，同时通知群集激活那台服务器上的 Session 对象。

七、WebSphere6.1 正常使用一年后的莫名问题管理器



websphere6.1 使用一年后，控制台中的应用服务器，应用程序的状态都显示不可用，现在只能从后台进行服务器的启用和停止，没法通过控制进行操作。

操作系统是 solaris 后台系统日志为：SSL 握手故障：已从目标主机: 端口“10.67.2.159:8881”发送主题 DN 为“CN=CWServer-E, O=IBM, C=US”的签署者。可能需要将该签署者添加到本地信任库“/opt/WebSphere6/AppServer/profiles/Dmgr01/config/cells/CWServer-ECe1101/trust.p12”，它位于从 SSL 配置文件“security.xml”装入的 SSL 配置别名“CellDefaultSSLSettings”中。来自 SSL 握手异常的扩展错误消息为“No trusted certificate found”。

答

在管理控制台中找到旧的证书，点击“创建自签署证书”，其中公共名填写主机名，有效期可以适当填写长一些（例如 10 年），组织名可以写 IBM 或者您的公司缩写。创建完毕，在证书列表中，选择旧的证书，然后点击“替换”，将旧的证书换为新的证书确定，保存，同步注销管理控制台。◆

六、WAS hang 有自动 kill 的方法吗?



Systemout.log 中提示

[06-10-18 8:53:06:702
CST] 0000000e

ThreadMonitor W

WSVR0605W; 线程“WebContainer ;
16137” (00007396) 已活动了 746086
毫秒，可能被挂起了。服务器中可能总
共挂起了 29 个线程。

答

不能自动 kill，因为应用服务器无法判断 Hang。

八、【WAS 入门开窍班】 cluster 中权重值修改后，怎样才能生效?



cluster 中权重值修改后，怎样才能使权重值生效呢?

答

改了权重值后，只用重启对应的应用实例，再更新下 web 插件就可以了

- 1: 配置新的权重信息；
- 2: 更新 web 插件，并传播插件；
- 3: 重启相应的应用服务器。





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，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@twtgroup.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。

- 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。



IBM WebSphere行业解决方案的专业纸质读物

《WebSphere 中国》杂志叙述许多成功的行业解决方案，同时也分享了各种问题的解决经验。其生动的图案与丰富的内容，不但吸引了我们这些追求技术人的眼球，也让我们不用被学究式的枯燥乏味的话语搞得头昏脑胀。如果你正为项目一筹莫展，或是为系统性能问题愁眉苦脸，那这本杂志你一定不能错过。

——中数通信息有限公司 管理支撑部开发经理 戴科冕

《WebSphere 中国》是一本适合系统集成商、独立软件开发者的各类技术人员了解、学习、提高WebSphere的杂志。她既可以让你追踪WebSphere软件发展的技术前沿精粹又可以让你找到适合自己的实战指南。她既可以让你了解企业级的解决方案又可以让你欣赏到叫人眼前一亮的小技巧。她既可以让你看到专家的点评又可以让你看到身边的同事朋友的才艺展示。

——富通东方软件部技术经理 张东焕

《WebSphere 中国》杂志目前已经出到第四期了，虽然它的开始时间并不是很长，但是作为国内第一个WebSphere领域的技术杂志，前景我十分看好。在内容方面我们可以看到像寇卫东、刘宏宇等专家的文章，其技术含量在国内也是首屈一指的，同时杂志内容的来源是大众化的，许多作者都是使用WebSphere相关产品很多年的人，具有着相当丰富的搭建、维护和调优等工作的经验，他们写的很多文章都是众多WebSphere用户经常遇到的情况，给许多读者带来了技术上的帮助。我也是受益者之一。

众所周知，一本杂志的核心价值在于这本杂志的内容，没有好的内容，就没有读者和市场。而作为IT技术领域的杂志，其内容的好坏在于文章的技术含量和读者的需求，这一点，《WebSphere 中国》做的还是很不错的，每期杂志征稿前都会在论坛做调研，读者提需求，出版的文章也都是读者最需要的技术型文章，这是这本杂志基于大众的一个特点。我也在第三期和本期杂志上发表了一些技术性文章，希望能给广大读者带来受益。

祝愿《WebSphere 中国》杂志持续发展，越办越好，打造成为中国WebSphere领域的领头羊刊物。

——中国电信集团陕西公司 系统集成部，中间件高级工程师