



(12) 发明专利

(10) 授权公告号 CN 102662705 B

(45) 授权公告日 2015.06.03

(21) 申请号 201210092354.4

[0061], [0074]-[0079] 段及图 5, 7-10.

(22) 申请日 2012.03.31

US 2009/0228531 A1, 2009.09.10, 说明书

(73) 专利权人 中国工商银行股份有限公司
地址 100140 北京市西城区复兴门内大街
55 号

[0024]-[0029] 及图 4-6.

CN 101739436 A, 2010.06.16, 全文.

CN 101969442 A, 2011.02.09, 全文.

US 2004/0111698 A1, 2004.06.10, 全文.

US 2006/0265469 A1, 2006.11.23, 全文.

(72) 发明人 杨祖龙 谢波 郑宝城 伍宏卫

审查员 孙韬敏

(74) 专利代理机构 中科专利商标代理有限责任
公司 11021

代理人 宋焰琴

(51) Int. Cl.

G06F 9/445(2006.01)

G06F 11/14(2006.01)

(56) 对比文件

US 2002/0111972 A1, 2002.08.15, 说明书
[0061], [0074]-[0079] 段及图 5, 7-10.

US 2002/0111972 A1, 2002.08.15, 说明书

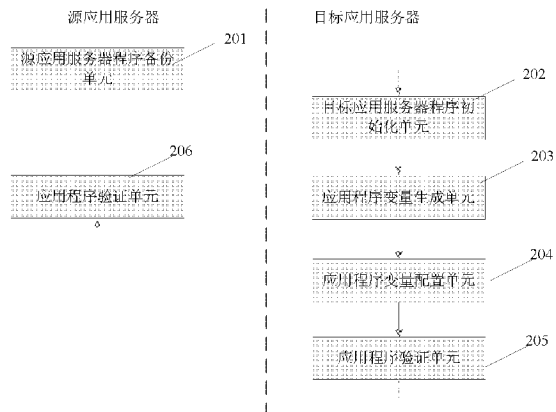
权利要求书3页 说明书7页 附图5页

(54) 发明名称

一种对计算机集群的系统环境进行升级的系统及方法

(57) 摘要

本发明公开了一种对计算机集群的系统环境进行升级的方法,包括步骤:通过 PC 客户端远程登录源服务器,对源应用程序进行备份;通过 PC 客户端远程登录目标服务器,在目标应用服务器上初始化操作;通过 PC 客户端远程登录目标服务器生成目标服务器应用程序变量配置文件;通过 PC 客户端远程登录目标服务器,将前一步骤中生成的变量内容写入应用程序对应的文件中;通过 PC 客户端远程登录目标服务器,启动目标应用程序并验证;如果验证通过,通过 PC 客户端远程登录源服务器,启动源应用程序并验证,从而完成迁移部署。应用本发明的方法,能够降低人力成本,提高环境升级效率,降低环境部署出错的概率,实现同时部署 MxN 套的环境的效果。



1. 一种对计算机集群的系统环境进行升级的系统,该系统包括:
源应用服务器模块,其上部署有现有环境源应用程序单元;
PC 客户端模块,用于用户远程登录源应用服务器模块或目标应用服务器模块,调用执行源应用服务器模块和目标应用服务器模块上的各功能单元;
一个或多个目标应用服务器模块,其包括从源应用服务器模块迁移过来的源应用程序单元;

其中,所述 PC 客户端模块包括:从源应用服务器模块向目标应用服务器模块迁移部署源应用程序单元的迁移模块;

所述迁移模块包括:通过 PC 客户端模块远程登录源应用服务器模块,对源应用程序单元进行备份的单元;通过 PC 客户端模块远程登录目标应用服务器模块,在目标应用服务器模块上进行初始化操作的单元;通过 PC 客户端模块远程登录目标应用服务器模块生成目标应用服务器应用程序变量配置文件的单元;通过 PC 客户端模块远程登录目标应用服务器模块,将生成的变量内容写入目标应用服务器应用程序对应的文件中的单元;通过 PC 客户端模块远程登录目标应用服务器模块,启动目标应用程序单元并验证的单元;如果验证通过,通过 PC 客户端模块远程登录源应用服务器模块,启动源应用程序单元并验证,从而完成迁移部署的单元;

其中,目标应用服务器模块进一步包括:应用程序变量生成单元,用于生成目标应用服务器应用程序变量配置文件,对于输入的源应用程序变量配置文件 SourceAppVar.xml,应用程序变量生成单元读取源应用程序变量配置文件,对其中的 Field 变量信息循环读取并逐一判断,对于所读出的一个变量信息,判断变量是否需要修改,如果不需要,则将该源应用程序变量原来的内容,写入目标配置文件 TargetAppVar.xml,如果需要,则分配新的变量内容并写入目标配置文件 TargetAppVar.xml,直到配置文件 SourceAppVar.xml 中所有变量都读取完成,然后输出新目标配置文件 TargetAppVar.xml;应用程序变量配置单元,用于将目标应用程序变量内容写入目标应用程序对应的文件中,对于输入的目标配置文件 TargetAppVar.xml,应用程序变量配置单元依次读取该目标配置文件中的目标应用程序变量相关信息,将目标应用程序变量的值更新写入目标应用程序对应文件中,直到配置文件中所有变量都读取完成。

2. 如权利要求 1 所述的系统,其特征在于,源应用服务器模块进一步包括:源应用服务器程序备份单元,用于对源应用程序单元打包备份传输;应用程序验证单元,用于启动源应用程序单元并进行验证。

3. 如权利要求 2 所述的系统,其特征在于,

目标应用服务器模块进一步包括:目标应用服务器程序初始化单元,用于目标应用服务器模块上执行初始化操作;应用程序验证单元,用于启动目标应用程序单元并进行验证。

4. 如权利要求 3 所述的系统,其特征在于,源应用服务器程序备份单元对源应用程序单元进行打包,如果源应用服务器模块为 UNIX 系统,则直接对源应用程序目录压缩打包为 tar 包,如果源应用服务器模块为 WINDOWS 系统,则直接对源应用程序目录压缩打包为 zip 包,存放于源应用服务器共享目录或文件服务器目录下,以便目标应用服务器获取。

5. 如权利要求 4 所述的系统,其特征在于,目标应用服务器程序初始化单元在目标应用服务器建立与源应用服务器应用程序单元一致的文件系统、目录、用户、组,从源应用服

务器共享目录或文件服务器取得源应用程序压缩包,解压源应用程序压缩包,确保目标应用服务器上最终所解压完成的应用程序所在文件系统及目录与源应用程序单元完全一致。

6. 如权利要求 5 所述的系统,其特征在于,应用程序验证单元用于在启动应用程序单元后判断应用程序服务是否正常,能否正常登录,如果正常,返回验证成功标志,如果不正常,返回验证失败标志。

7. 一种对计算机集群的系统环境进行升级的方法,该方法包括步骤:

通过 PC 客户端远程登录源应用服务器,对源应用程序进行备份;

通过 PC 客户端远程登录目标应用服务器,在目标应用服务器上进行初始化操作;

通过 PC 客户端远程登录目标应用服务器生成目标应用服务器应用程序变量配置文件;

通过 PC 客户端远程登录目标应用服务器,将前一步骤中生成的变量内容写入目标应用程序对应的文件中;

通过 PC 客户端远程登录目标应用服务器,启动目标应用程序并验证;

如果验证通过,通过 PC 客户端远程登录源应用服务器,启动源应用程序并验证,从而完成迁移部署;

其中,目标应用服务器进一步包括:应用程序变量生成单元,用于生成目标应用服务器应用程序变量配置文件,对于输入的源应用程序变量配置文件 SourceAppVar. xml,应用程序变量生成单元读取源应用程序变量配置文件,对其中的 Field 变量信息循环读取并逐一判断,对于所读出的一个变量信息,判断变量是否需要修改,如果不需要,则将该源应用程序变量原来的内容,写入目标配置文件 TargetAppVar. xml,如果需要,则分配新的变量内容并写入目标配置文件 TargetAppVar. xml,直到源应用程序变量配置文件 SourceAppVar. xml 中所有变量都读取完成,然后输出新目标配置文件 TargetAppVar. xml;应用程序变量配置单元,用于将目标应用程序变量内容写入目标应用程序对应的文件中,对于输入的目标配置文件 TargetAppVar. xml,应用程序变量配置单元依次读取该目标配置文件中的目标应用程序变量相关信息,将变量的值更新写入目标应用程序对应文件中,直到目标配置文件中所有变量都读取完成。

8. 如权利要求 7 所述的方法,其特征在于,所述源应用服务器进一步包括:源应用服务器程序备份单元,用于对源应用服务器程序打包备份传输;应用程序验证单元,用于启动源应用程序并进行验证。

9. 如权利要求 8 所述的方法,其特征在于,目标应用服务器进一步包括:目标应用服务器程序初始化单元,用于目标应用服务器上执行初始化操作;应用程序验证单元,用于启动目标应用程序并进行验证。

10. 如权利要求 9 所述的方法,其特征在于,源应用服务器程序备份单元对源应用程序进行打包,如果源应用服务器为 UNIX 系统,则直接对源应用程序目录压缩打包为 tar 包,如果源应用服务器为 WINDOWS 系统,则直接对源应用程序目录压缩打包为 zip 包,存放于源应用服务器共享目录或文件服务器目录下,以便目标应用服务器获取。

11. 如权利要求 10 所述的方法,其特征在于,目标应用服务器程序初始化单元在目标应用服务器建立与源应用服务器应用程序一致的文件系统、目录、用户、组,从源应用服务器共享目录或文件服务器取得源应用程序压缩包,解压源应用程序压缩包,确保目标应用

服务器上最终所解压完成的应用程序所在文件系统及目录与源应用程序完全一致。

12. 如权利要求 11 所述的方法,其特征在于,应用程序验证单元用于在启动应用程序后判断应用程序服务是否正常,能否正常登录,如果正常,返回验证成功标志,如果不正常,返回验证失败标志。

一种对计算机集群的系统环境进行升级的系统及方法

技术领域

[0001] 本发明涉及计算机集群中系统环境升级部署技术领域,特别是涉及一种对计算机集群的系统环境进行升级的系统及方法,用于开放平台 B/S(Browser/Server) 架构的软件系统应用服务器程序快速迁移和部署。

背景技术

[0002] 根据软件生命周期的不同阶段,需要依次搭建开发、测试、生产等多种软件系统环境。新的软件应用系统要依次经历开发单元测试环境、集成测试环境、系统测试环境、适应性测试环境、投产测试环境、实际生产环境多个阶段。

[0003] 由于软件应用系统一般需要分多期开发,因此以上各阶段的环境将并行存在,同时本阶段的系统环境需要继承上一阶段的系统环境。由于目前没有成熟可靠的应用程序快速迁移部署方法,各个应用系统具有自身不同的参数配置信息及升级安装文档、没有统一的可供通用迁移部署方法,现有系统环境升级的一般方法是,针对以上各个阶段的环境采用全新安装方式搭建,不仅需要手工安装各类系统软件,还需要手工更新一系列的应用程序级版本及补丁。其存在的缺陷是需要花费较多的人力及时间,也存在手工误操作导致环境出错的风险,同时新搭建的环境对于上一阶段系统环境的继承可能不完整,导致新系统功能、性能不稳定,且不能涵盖本系统前期所有功能模块,对后续测试验证形成一定的风险。

发明内容

[0004] 有鉴于此,本发明提出了一种开放平台常见的 B/S 结构的对计算机集群的系统环境进行升级的系统及方法,以满足各测试环境(或生产环境)下对同一应用系统并行存在多套环境进行快速搭建部署的需要,降低应用程序安装部署、大部分参数配置所消耗的人力成本,提高应用系统资源利用率和环境部署效率。

[0005] 本发明提出的对计算机集群的系统环境进行升级的系统包括:源应用服务器,其上部署有现有环境应用程序;PC 客户端,用于用户远程登录源应用服务器或目标应用服务器,调用执行源应用服务器程序和目标应用服务器上的各功能单元;企业局域网,用于连接源应用服务器与目标应用服务器;一个或多个目标应用服务器,其上部署从源应用服务器迁移过来的应用程序,其中从源应用服务器向目标应用服务器迁移部署应用程序包括:通过 PC 客户端远程登录源服务器,对源应用程序进行备份;通过 PC 客户端远程登录目标服务器,在目标应用服务器上初始化操作;通过 PC 客户端远程登录目标服务器生成目标服务器应用程序变量配置文件;通过 PC 客户端远程登录目标服务器,将前一步骤中生成的变量内容写入应用程序对应的文件中;通过 PC 客户端远程登录目标服务器,启动目标应用程序并验证;如果验证通过,通过 PC 客户端远程登录源服务器,启动源应用程序并验证,从而完成迁移部署。

[0006] 本发明提出的对计算机集群的系统环境进行升级的方法包括步骤:通过 PC 客户

端远程登录源服务器,对源应用程序进行备份;通过 PC 客户端远程登录目标服务器,在目标应用服务器上初始化操作;通过 PC 客户端远程登录目标服务器生成目标服务器应用程序变量配置文件;通过 PC 客户端远程登录目标服务器,将前一步骤中生成的变量内容写入应用程序对应的文件中;通过 PC 客户端远程登录目标服务器,启动目标应用程序并验证;如果验证通过,通过 PC 客户端远程登录源服务器,启动源应用程序并验证,从而完成迁移部署。

[0007] 使用本发明,可以满足测试环境(或生产环境)下对同一应用系统并行存在多套环境进行快速搭建部署的需要。同时也适用于在同一台服务器上部署多套不同应用程序的情况,使用本发明能在同一台服务器上快速生成多套应用程序基础环境,并在此基础上可由各个不同的应用程序根据自身应用特点进行配置,从而降低了应用程序安装部署的人力成本,提高了应用系统资源利用率和环境部署效率。

[0008] 经实践证明,在不影响应用系统的测试效果的前提下,通过对应用服务器程序变量自动生成、配置、进行快速迁移,可以节省逐一安装应用软件及其补丁的时间,从一个阶段到另一阶段同等复杂度环境的迁移时间能大幅减少,提高了测试、生产环境的维护效率。

附图说明

[0009] 图 1 是本发明对计算机集群的系统环境进行升级的系统结构示意图;

[0010] 图 2 是本发明源应用服务器和目标应用服务器的功能结构图;

[0011] 图 3 是本发明对计算机集群的系统环境进行升级的方法流程图;

[0012] 图 4 是源应用程序备份单元的工作流程图;

[0013] 图 5 是目标应用服务器初始化单元的工作流程图;

[0014] 图 6 是应用程序变量生成单元的工作流程图;

[0015] 图 7 是应用程序变量配置单元的工作流程图;

[0016] 图 8 是应用程序验证单元的工作流程图;

具体实施方式

[0017] 为使本发明的目的、技术方案和优点更加清楚明白,以下结合具体实施例,并参照附图,对本发明进一步详细说明。

[0018] B/S 体系结构是一个标准的三层结构,用户使用浏览器客户端程序通过广域网或 Intranet 内部网,使用 HTTP 或 HTTPS 通讯协议访问后台应用。其中,WEB 服务器完成静态或动态页面的显示以及一些数据完整性检查等简单逻辑;应用服务器完成所有业务处理逻辑,以及对数据库的访问和交互;数据库服务器则主要完成后台业务处理功能。

[0019] 应用服务器通常采用常用的中间件产品,如 Tomcat, BEA 公司的 WebLogic、IBM 公司的 WebSphere Application Server 等。应用服务器上的软件包括以下几个部分:操作系统软件(如 WINDOWS, AIX, Solaris 等)、应用服务器软件(如 WebSphere Application Server 等)、应用程序(通常包括程序文件、一些与应用有关的配置文件等)。

[0020] 本发明的目的主要是在目标服务器操作系统软件已经存在的前提下,实现应用服务器系统软件级及应用程序升级的一种快速迁移部署的系统及方法。

[0021] 图 1 是本发明提供的对计算机集群的系统环境进行升级的系统的示意图,该系统

包括源应用服务器 101、PC 客户端 102、企业局域网 103 和目标应用服务器,其中目标服务器可以有多台,例如第一目标应用服务器 104、……、第 N 目标应用服务器 105 等。下面对该系统的各部分进行详细描述。

[0022] 源应用服务器 101 上部署了现有环境应用程序,称为源应用服务器。用作为迁移部署的基础应用系统环境,是整个迁移部署的基础。如何迁移应用服务器的应用程序是本发明的核心。

[0023] 如图 2 所示,源应用服务器 101 进一步包括源应用服务器程序备份单元 201 和应用程序验证单元 206。源应用服务器程序备份单元 201,主要完成对源应用服务器程序打包备份传输功能。应用程序验证单元 206 用于启动源应用程序并进行验证。

[0024] 图 4 示出了源应用程序备份单元 201 的工作流程。

[0025] 参照图 4,在步骤 401:通过 PC 客户端远程登录源服务器,停止源应用程序及服务进程,以便进行环境迁移。在步骤 402:对源应用程序进行打包,一般情况下,如果源应用服务器为 UNIX 系统,则直接对源应用程序目录压缩打包为 tar 包,如果源应用服务器为 WINDOWS 系统,则直接对源应用程序目录压缩打包为 zip 包,存放于源服务器共享目录或文件服务器目录下,以便目标服务器获取。在步骤 403:将步骤 402 所生成的源应用程序压缩包传输保存至源服务器共享目录或文件服务器目录下。

[0026] 图 8 为应用程序验证单元 206 的工作流程。

[0027] 参照图 8,在步骤 801:通过 PC 客户端远程登录应用服务器,启动应用程序。在步骤 802:判断应用程序服务是否正常,能否正常登录。如果服务正常,则跳到步骤 803;如果服务不正常,则跳到步骤 804。在步骤 803:返回验证成功标志。在步骤 804:返回验证失败标志。

[0028] PC 客户端 102 是系统的人机交互界面,用户(系统维护人员)通过该装置远程登录源应用服务器或目标应用服务器,执行源应用服务器程序备份单元、验证单元或目标应用服务器初始化单元、应用程序变量生成、配置单元、验证单元,在服务器上进行相关快速迁移一套或多套系统环境的部署操作。

[0029] 企业局域网 103 用于连接源应用服务器与目标应用服务器之间的网络,一般为局域网。

[0030] 目标应用服务器(图 1 装置 104、图 1 装置 105) 是新的应用服务器环境,在该环境上需要部署从源应用服务器迁移过来的应用程序,将作为全新的一套应用程序环境。目标应用服务器可能有多个,即将一个源应用服务器程序迁移部署到多个目标应用服务器上。

[0031] 参照图 2,目标应用服务器包括目标应用服务器程序初始化单元 202、应用程序变量生成单元 203、应用程序变量配置单元 204、应用程序验证单元 205。目标应用服务器程序初始化单元 202 主要完成在目标应用服务器上创建目录、获取压缩包、解压等初始化操作。应用程序变量生成单元 203 用于生成目标服务器应用程序变量配置文件。应用程序变量配置单元 204 将应用程序变量内容写入应用程序对应的文件中。应用程序验证单元 205 用于启动目标应用程序并进行验证。

[0032] 图 5 示出了目标应用服务器初始化单元 202 的工作流程图。

[0033] 参照图 5,首先在步骤 501,通过 PC 客户端远程登录目标服务器,在目标服务器建立与源应用服务器应用程序一致的文件系统、目录、用户、组。如果迁移的系统是 UNIX 系

统,则需要目标服务器建立与源应用服务器应用程序一致的文件系统及目录、建立应用程序所属操作系统用户、组。如果目标服务器上无对应的文件系统及目录,则可以采用软链接的方式建立与源应用程序一样的文件系统及目录。

[0034] 如果迁移的源系统为 WINDOWS 系统,则只需要在目标服务器建立与源应用服务器应用程序一致的逻辑盘符及目录结构。如果目标服务器上无对应的逻辑盘符及目录结构,则也可以采用软链接等方式建立与源应用程序一样的逻辑盘符及目录结构。

[0035] 在步骤 502:通过 PC 客户端远程登录目标服务器,从源应用服务器共享目录或文件服务器取得源应用程序压缩包。由于源应用服务器或文件服务器与目标应用服务器都在同一局域网中,可采用共享方式或 FTP 方式二进制传输压缩包。

[0036] 在步骤 503,通过 PC 客户端远程登录目标服务器,解压源应用程序压缩包,确保目标服务器上最终所解压完成的应用程序所在文件系统(或逻辑盘符)及目录要与源应用程序完全一致。

[0037] 图 6 示出了应用程序变量生成单元 203 的工作流程图。

[0038] 参照图 6,在步骤 601,输入源应用程序变量配置文件 SourceAppVar.xml,文件采用 XML 标记语言。该文件内容片断举例如下:

[0039] <var>

[0040] <Field FieldNO = " 1" CfieldName = " 是否需要走统一认证"

[0041] EfieldName = " enableSSIC"

[0042] file = " /data/AppSrv01/rmts.ear/rmtsweb.war/WEB-INF/config/dse.xml "

[0043] value = " false" isUpdate = " 0" />

[0044] <Field FieldNO = " 2" CfieldName = " 客户机 IP 地址"

[0045] EfieldName = " clientIP"

[0046] file = " /data/AppSrv01/rmts.ear/rmtsweb.war/WEB-INF/config/dsedata.xml "

[0047] value = " 122.8.78.92:11070" isUpdate = " 1" />

[0048] <Field FieldNO = " 3" CfieldName = " WASJDBC 连接串"

[0049] EfieldName = " WasJDBCConnectionPool"

[0050] file = " /data/AppSrv01/rmts.ear/rmtsweb.war/WEB-INF/config/dsesrvce.xml "

[0051] value = " jdbc:oracle:thin:@122.8.7.18:1521:rmts" isUpdate = " 1" />

[0052] </var>

[0053] 其中 FieldNO 表示变量的序号, CfieldName 表示该变量的中文描述, EfieldName 表示该变量英文名称, file 表示该变量位于应用程序哪个配置文件中, value 表示该变量在源应用程序中所设置的值, isUpdate 表示迁移时是否需要修改, 0 表示不修改, 1 表示需要根据实际情况进行修改。

[0054] 在步骤 602,读取源应用程序变量配置文件,对其中的 Field 变量信息循环读取并逐一判断。

[0055] 在步骤 603,对于步骤 602 中所读出的一个变量信息,如果 isUpdate 标志为 1,则表示该变量的值需要对应修改,则跳到步骤 604;如果 isUpdate 标志为 0,则表示该变量的

值无需修改,则跳到步骤 605。

[0056] 在步骤 604,分配新的变量内容并写入目标配置文件 TargetAppVar.xml。

[0057] 在步骤 605,将该应用程序变量原来的内容,写入目标配置文件 TargetAppVar.xml。

[0058] 在步骤 606,判断配置文件 SourceAppVar.xml 中所有变量是否读取完成,如果没有读取完成,则跳到步骤 602 继续循环处理;如果读取完成,则跳到步骤 607。

[0059] 在步骤 607,输出新配置文件 TargetAppVar.xml,即应用程序迁移到目标服务器后,需要更新的目标配置文件。

[0060] 图 7 示出了应用程序变量配置单元 204 的工作流程图。

[0061] 参照图 7,在步骤 701:输入应用程序变量配置文件 TargetAppVar.xml,即图 5 流程中应用程序变量生成单元所输出的目标配置文件。在步骤 702:依次读取新目标配置文件中的应用程序变量相关信息。在步骤 703:将该变量的值更新写入目标应用程序对应文件中。在步骤 704:判断配置文件中变量是否读取完成,如果没有读取完成,则跳到步骤 702 继续循环处理;如果读取完成,则跳到步骤 705。在步骤 705:返回成功完成标志。

[0062] 图 8 示出了应用程序验证单元 205 的工作流程图。

[0063] 在步骤 801:通过 PC 客户端远程登录应用服务器,启动应用程序。在步骤 802:判断应用程序服务是否正常,能否正常登录。如果服务正常,则跳到步骤 803;如果服务不正常,则跳到步骤 804。在步骤 803:返回验证成功标志。在步骤 804:返回验证失败标志。

[0064] 概括来说,源应用服务器、目标应用服务器及 PC 客户端通过企业局域网连接起来,构成企业测试环境或生产环境中某个应用系统多套物理服务器环境。一般情况源应用服务器上部署了测试或生产环境某个应用系统程序,目标应用服务器与源应用服务器环境类似,上面部署为源应用服务器应用程序的扩展备份或该应用系统下一阶段的测试环境,与源应用服务器并行存在。例如源应用服务器可能部署了某个应用的开发测试环境、N 台目标服务器上可能部署了该应用的集成测试环境、系统测试环境、适应性测试环境、投产测试环境。在运用本发明方法进行系统环境升级时,需要以源应用服务器应用程序为基础进行迁移,按照相应方法部署到目标服务器环境上。

[0065] 图 3 为本发明对计算机集群的系统环境进行升级的方法流程图。

[0066] 参照图 3,在步骤 301,通过 PC 客户端远程登录源服务器,调用“源应用服务器程序备份单元”,对源应用程序进行备份。在步骤 302:通过 PC 客户端远程登录目标服务器,调用“目标应用服务器程序初始化单元”,在目标应用服务器上创建目录、获取压缩包、解压等初始化操作。在步骤 303:通过 PC 客户端远程登录目标服务器,调用“应用程序变量生成单元”,生成目标服务器应用程序变量配置文件。在步骤 304:通过 PC 客户端远程登录目标服务器,调用“应用程序变量配置单元”,将图 3 步骤 303 生成的变量内容写入应用程序对应的文件中。在步骤 305:通过 PC 客户端远程登录目标服务器,调用“应用程序验证单元”,启动目标应用程序并验证。在步骤 306:通过 PC 客户端远程登录源服务器,接收步骤 305 中返回的验证结果。在步骤 307:判断验证结果是否通过,如果没有通过,则跳到步骤 308;如果通过,则跳到步骤 309。在步骤 308:重复目标服务器上流程处理。在步骤 309:通过 PC 客户端远程登录源服务器,调用“应用程序验证单元”,启动源应用程序并验证。在步骤 310:迁移部署完成。

[0067] 其中源应用服务器程序备份单元、目标应用服务器程序初始化单元、应用程序变量生成单元、应用程序变量配置单元、应用程序验证单元的工作流程在前面已经参照图 4 至图 7 进行了详细描述,在此不再赘述。

[0068] 上面结合附图详细描述了本发明对计算机集群的系统环境进行升级的系统的组成结构以及方法的具体流程,在对本发明的具体应用过程中,用户可以按照下面的流程进行操作:1) 系统维护人员梳理需要迁移部署的应用系统,确定源应用服务器、目标应用服务器上应用程序文件系统、目录、用户、组等基本信息。2) 停止源应用服务器上的应用程序及服务进程,以便进行环境迁移。3) 对源应用程序进行打包,存放于源服务器共享目录或文件服务器上,以便目标服务器获取。4) 目标服务器进行初始化处理:获取源应用程序压缩包,并传输该压缩包到目标应用服务器临时目录下;在目标服务器建立与源应用服务器应用程序一致的文件目录系统、用户、组;在目标应用服务器解压源应用程序压缩包,确保目标服务器上最终所解压完成的应用程序所在文件系统(或逻辑盘符)及目录与源应用程序完全一致。5) 启动应用程序变量生成单元,对源应用程序变量配置文件 SourceAppVar.xml 进行循环处理,并根据读取的变量信息,相应生成新的目标应用程序变量配置文件 TargetAppVar.xml。6) 启动应用程序变量配置单元,对目标应用程序相关配置信息进行修改,如修改目标应用程序对应的数据库连接信息等。7) 启动应用程序验证单元,对目标服务器上应用程序进行验证。8) 若目标服务器重启并验证通过,则启动应用程序验证单元,对源服务器上应用程序进行重启并进行验证,恢复源应用对外服务。9) 迁移部署完成。

[0069] 综上所述,本发明涉及的关键点包括:1) 用 XML 文件描述应用程序变量字段配置项本发明将需要动态修改的变量配置项,通过 XML 标记语言进行描述。应用程序变量生成单元、配置单元首先解析出该 XML 文件中的各个变量配置项,然后判断并决定是否重新修改变量内容。由于 XML 标记语言具有良好的扩展性,方便的定义更多新的变量配置项,方便不同应用程序快速扩展。2) 应用程序迁移具有多平台适应性在目标服务器建立与源应用服务器应用程序一致的文件目录系统、用户、组。无论所迁移的系统是 UNIX 系统还是 WINDOWS 系统,只需要在目标服务器建立与源应用服务器应用程序一致的文件系统(或逻辑盘符)及目录、如果目标服务器上无对应的文件系统及目录,则可以采用软链接的方式建立与源应用程序一样的文件系统(或逻辑盘符)及目录。

[0070] 本发明提出的目前开放平台最常见的 BS(Browser/Server) 应用系统快速迁移部署的系统及方法,与现有技术的安装方式相比,具有以下几方面的优点:1) 降低人力成本,提高环境升级效率。由于人力及时间的要求,往往需要快速搭建一套或多套新的开放平台环境,如果采用全新安装方式搭建新的环境,要花费较多的人力及时间成本。本发明提供的开放平台应用程序快速迁移部署的系统及方法,降低了应用程序安装部署的人力成本,提高了环境升级效率。2) 降低环境部署出错的概率。由于应用系统的分期投产要求,在跨月度(季度)版本需要搭建多个环境并行开发测试。如果传统方式搭建新环境,由于需要安装过多版本或补丁,可能出现差错或遗漏,导致新搭建的环境对于上一版本系统环境的继承性不够,不能涵盖本系统前期所有功能模块,后续测试验证风险较大。本发明为这类环境提供了快速迁移部署的系统及方法,极大地降低了后续环境出现差错或遗漏的情况。3) 实现同时部署 MxN 套的环境的效果。(N 台服务器上,每台服务器上可以对应 M 个应用)传统方法由于中间层系统软件对端口的独占性要求,导致一台服务器上往往只能部署一个应用

的一套系统环境,本方法的一个重大特点是能快速“一键式”实现在一台应用服务器或多台服务器上部署 N 个应用的 M 套系统环境,极大的提高了系统资源的利用率。4) 方便快捷扩展。本发明用 XML 文件描述相关配置项。将需要动态修改的变量配置项,通过 XML 标记语言进行描述,形成变量配置清单 XML 文件。由于 XML 标记语言具有良好的扩展性,可以灵活定义配置项,方便快捷扩展,以适应不同应用程序的特殊配置要求。

[0071] 以上所述的具体实施例,对本发明的目的、技术方案和有益效果进行了进一步详细说明,应理解的是,以上所述仅为本发明的具体实施例而已,并不用于限制本发明,凡在本发明的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

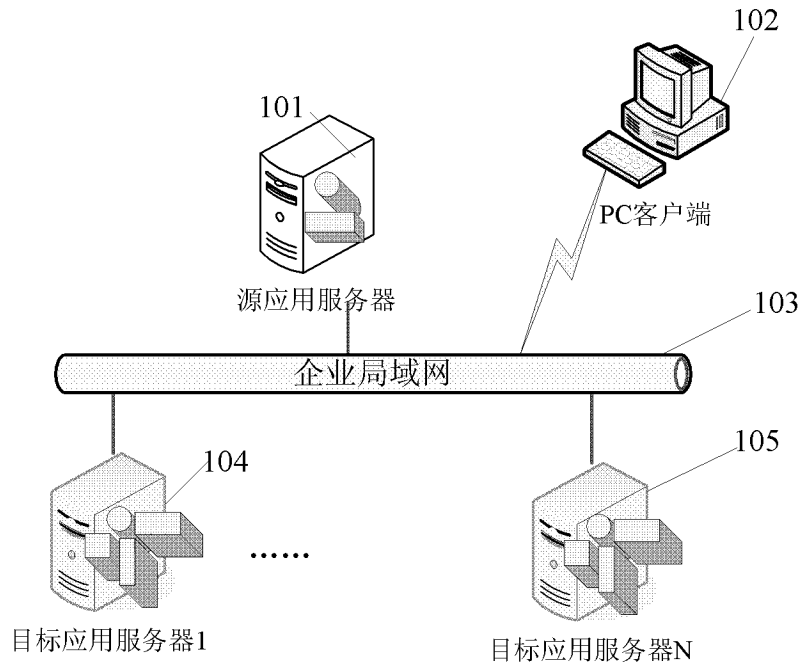


图 1

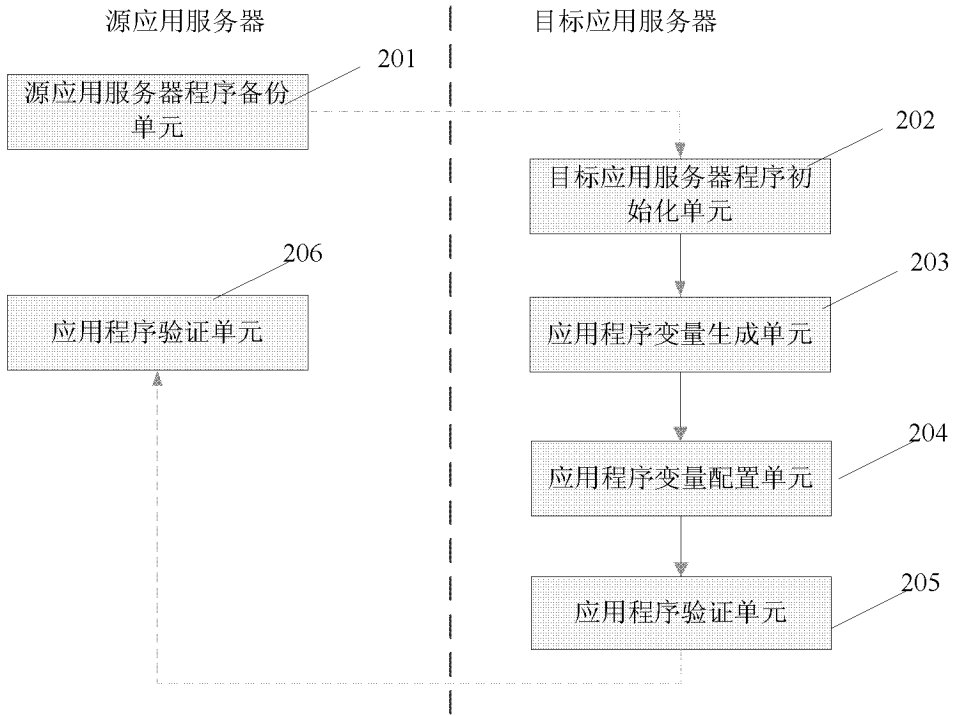


图 2

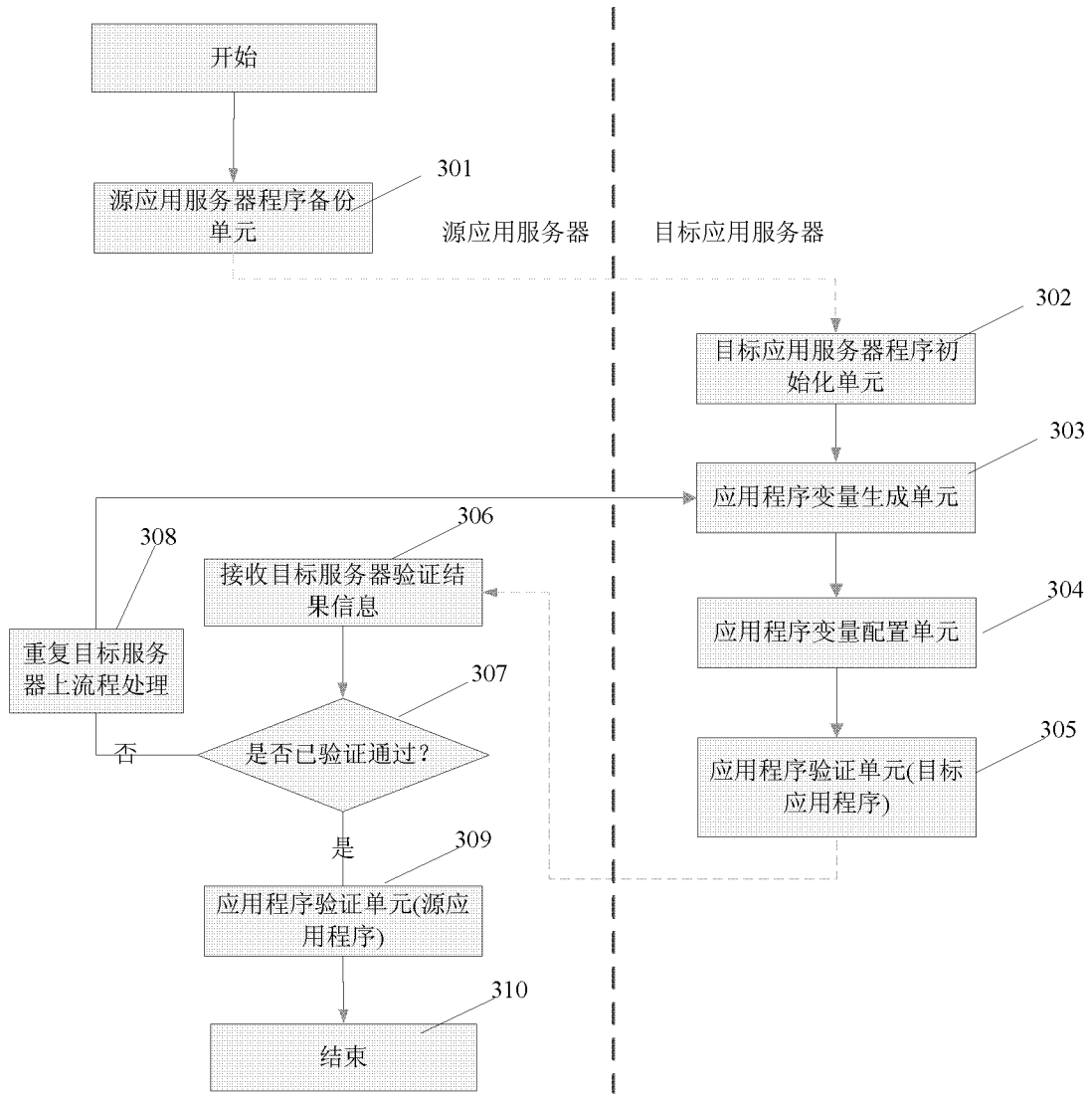


图 3

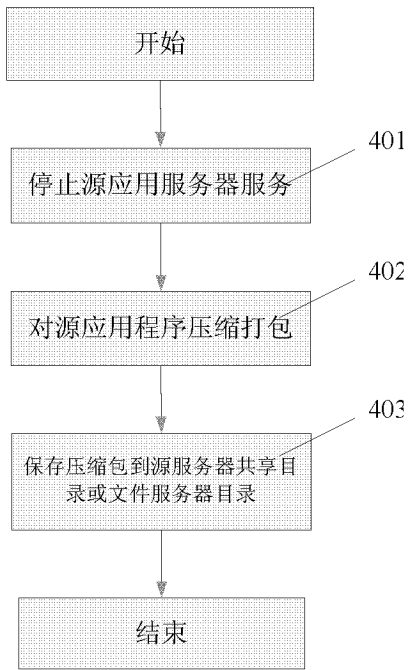


图 4

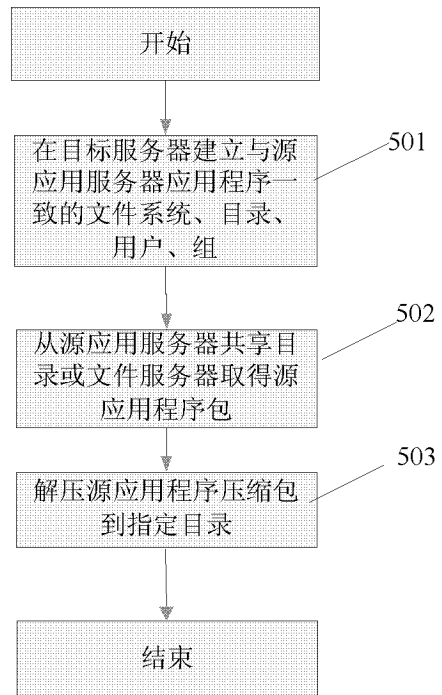


图 5

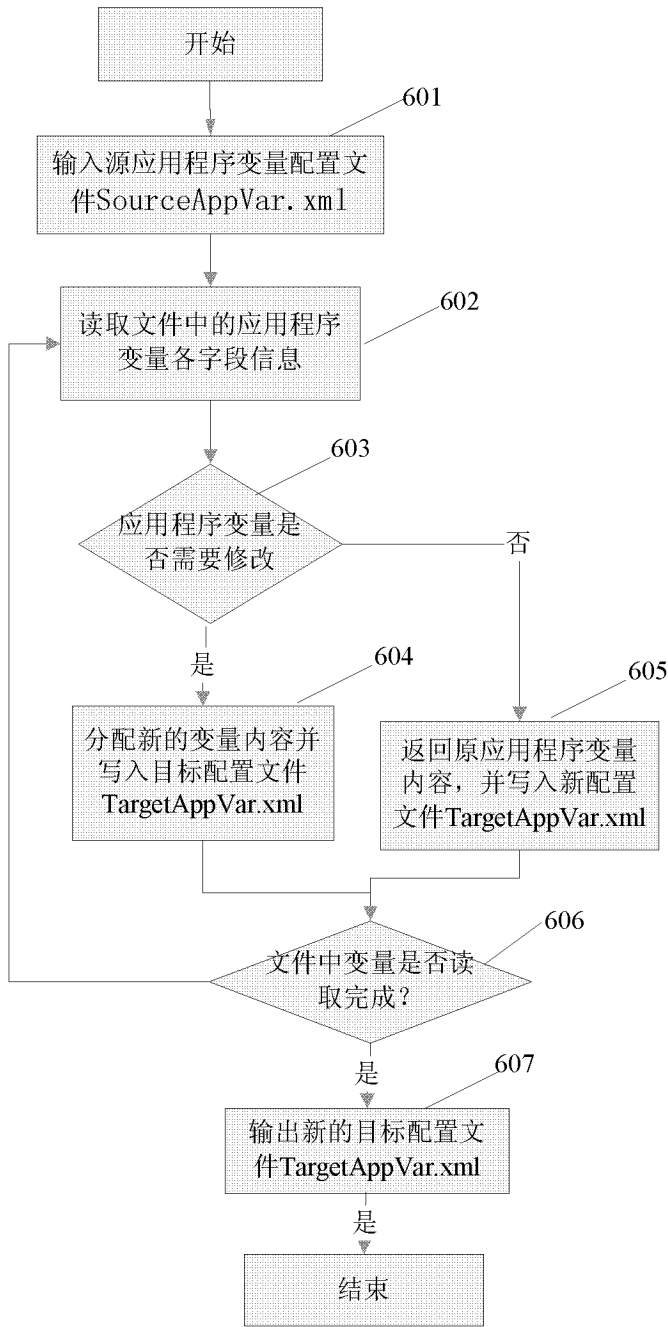


图 6

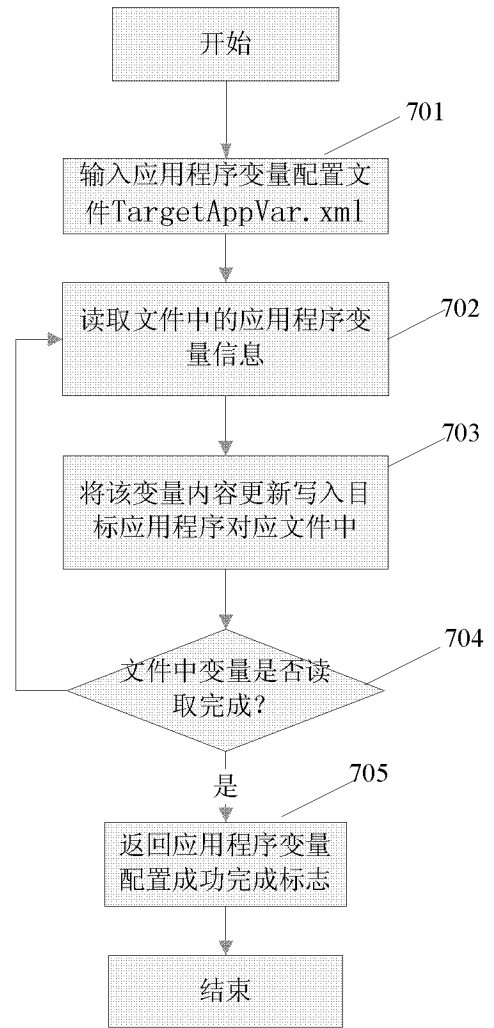


图 7

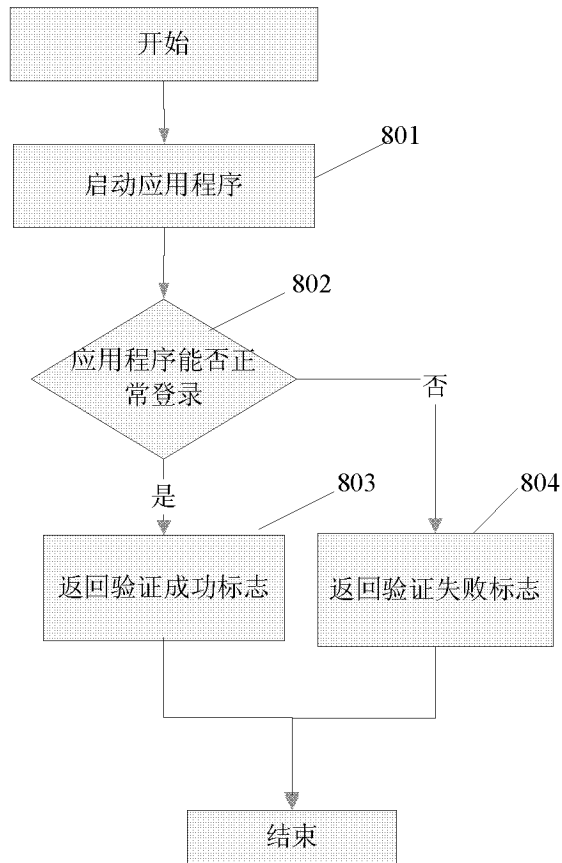


图 8