

(12) 发明专利申请

(10) 申请公布号 CN 102135895 A

(43) 申请公布日 2011. 07. 27

(21) 申请号 201010612561. 9

(22) 申请日 2010. 12. 29

(71) 申请人 华为软件技术有限公司  
地址 210012 江苏省南京市雨花台区安德门  
94 号

(72) 发明人 陈辉

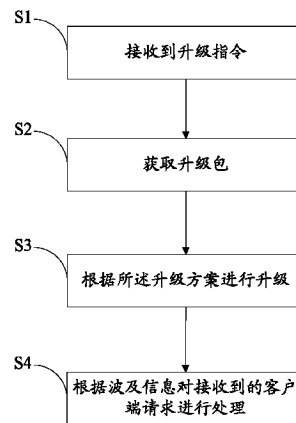
(74) 专利代理机构 深圳市深佳知识产权代理事  
务所 (普通合伙) 44285  
代理人 彭愿洁 李文红

(51) Int. Cl.  
G06F 9/445 (2006. 01)

权利要求书 2 页 说明书 5 页 附图 2 页

(54) 发明名称  
系统升级方法及系统

(57) 摘要  
本发明实施例公开了一种系统升级方法及系统,所述系统被预先划分为可动态加 / 卸载的多个模块,其中,任一模块通过预设接口与本模块之外的模块或系统进行数据交互,所述方法包括:接收到升级指令;获取升级包,所述升级包中包括升级方案;根据所述升级方案进行升级。可以看出,在本发明实施例所提供的技术方案中,系统被预先划分为可动态加 / 卸载的多个模块,在设计时,可令模块间可执行的功能或业务的耦合度尽量小,这样,当需要对某些功能或业务进行升级时,可仅对与之相关的模块进行升级,而其他模块还可提供正常的业务或功能。



1. 一种系统升级方法,其特征在于,所述系统被预先划分为可动态加 / 卸载的多个模块,其中,任一模块通过预设接口与本模块之外的模块或系统进行数据交互,所述方法包括:

接收到升级指令;  
获取升级包,所述升级包中包括升级方案;  
根据所述升级方案进行升级。

2. 如权利要求 1 所述的方法,其特征在于:

所述升级方案包括升级受限请求信息,所述升级受限请求信息包括在本次升级中业务处理受限的客户端请求的相关信息;

所述方法还包括根据所述升级受限请求信息对接收到的客户端请求进行处理。

3. 如权利要求 2 所述的方法,其特征在于,所述根据所述升级受限请求信息对接收到的客户端请求进行处理的具体实现方式为:

当接收到的客户端请求与所述升级受限请求信息相匹配时,返回提示正在升级的提示信息,否则对所述客户端请求进行正常处理。

4. 如权利要求 1-3 任一项所述的方法,其特征在于,当本次升级所涉及的模块为多个时,所述升级方案还包括模块的升级先后顺序。

5. 如权利要求 4 所述的方法,其特征在于,还包括:

确定本次升级所涉及的模块;

当本次升级所涉及的模块为多个时,根据所涉及各模块之间的预设关系,来设定模块的升级先后顺序;

根据本次升级所涉及的模块与客户端请求的预设关系,生成升级受限请求信息。

6. 如权利要求 5 所述的方法,其特征在于,

评估本次升级所波及的系统接口;

根据本次升级所波及的系统接口与客户端请求的预设关系,生成升级受限请求信息;

根据所述系统接口与系统各模块的预设关系,确定本次升级所涉及的模块的信息;

当本次升级所涉及的模块为多个时,根据所涉及各模块之间的预设关系来设定模块的升级先后顺序。

7. 一种系统,其特征在于,所述系统被预先划分为可动态加 / 卸载的多个模块,任一模块通过预设接口与本模块之外的模块或系统进行数据交互,所述系统包括过滤器和升级单元,所述过滤器包括接收单元,所述升级单元在所述接收单元接收到升级指令后,获取包括升级方案的升级包,并根据所述升级方案进行升级。

8. 如权利要求 1 所述的系统,其特征在于:

所述升级方案包括升级受限请求信息,所述升级受限请求信息包括在本次升级中业务处理受限的客户端请求的相关信息;

所述过滤器还包括处理单元,用于根据所述升级受限请求信息对接收到的客户端请求进行处理。

9. 如权利要求 8 所述的系统,其特征在于,当本次升级所涉及的模块为多个时,所述升级方案还包括模块的升级先后顺序。

10. 如权利要求 9 所述的系统,其特征在于,还包括评估单元,用于评估本次升级所波

及的系统接口,根据本次升级所波及的系统接口与客户端请求的预设关系,生成升级受限请求信息,根据所述系统接口与系统各模块的预设关系,确定本次升级所涉及的模块的信息,以及当本次升级所涉及的模块为多个时,根据所涉及的各模块之间的预设关系来设定模块的升级先后顺序。

## 系统升级方法及系统

### 技术领域

[0001] 本发明涉及数据处理领域,更具体地说,涉及系统升级方法及系统。

### 背景技术

[0002] 目前,系统升级,大部分是进行整体升级,在升级过程中,整个系统无法进行正常的业务,升级完成之后再业务重启。

[0003] 然而在实施本发明时,发明人发现,很多时候,升级仅是针对的一部分业务或功能,而上述升级方式,使得系统不能继续向用户提供其他不相关的业务或功能。

### 发明内容

[0004] 有鉴于此,本发明实施例目的在于提供可解决上述问题的系统升级方法及系统。

[0005] 为实现上述目的,本发明实施例提供如下技术方案:

[0006] 根据本发明实施例的一个方面,提供一种系统升级方法,所述系统被预先划分为可动态加/卸载的多个模块,其中,任一模块通过预设接口与本模块之外的模块或系统进行数据交互,所述方法包括:

[0007] 接收到升级指令;

[0008] 获取升级包,所述升级包中包括升级方案;

[0009] 根据所述升级方案进行升级。

[0010] 根据本发明实施例的另一个方面,提供一种系统,所述系统被预先划分为可动态加/卸载的多个模块,任一模块通过预设接口与本模块之外的模块或系统进行数据交互,所述系统包括过滤器和升级单元,所述过滤器包括接收单元,所述升级单元在所述接收单元接收到升级指令后,获取包括升级方案的升级包,并根据所述升级方案进行升级。

[0011] 从上述的技术方案可以看出,在本发明实施例所提供的技术方案中,系统被预先划分为可动态加/卸载的多个模块,在设计时,可令模块间可执行的功能或业务的耦合度尽量小,这样,当需要对某些功能或业务进行升级时,可仅对与之相关的模块进行升级,而其他模块还可提供正常的业务或功能。

### 附图说明

[0012] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0013] 图1为本发明实施例提供的系统升级方法流程图;

[0014] 图2为本发明实施例提供的过滤器工作流程图;

[0015] 图3为本发明实施例提供的系统升级方法另一流程图。

## 具体实施方式

[0016] 为了引用和清楚起见,下文中使用的技术名词、简写或缩写总结如下:

[0017] Internet:英特网;

[0018] URL:Uniform/Universal Resource Locator,统一资源定位符。Internet上的每一个网页都具有一个唯一的名称标识,通常称之为URL地址,这种地址可以是本地磁盘,也可以是局域网上的某一台计算机,更多的是Internet上的站点。简单地说,URL就是Web地址,俗称“网址”。

[0019] 依赖关系:如果两个元素其中一个的定义发生改变则会引起另一个元素发生变化则称这两个元素之间存在依赖关系。在软件开发中依赖关系非常常见,如函数间的依赖(函数A调用函数B,则函数A依赖函数B)、类之间的依赖(类A的某些方法调用了类B的方法,则类A依赖于类B)、模块间的依赖及系统间的依赖。

[0020] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0021] 目前,系统升级,大部分是进行整体升级,在升级过程中,整个系统无法进行正常的业务,升级完成之后再业务重启。

[0022] 然而在实施本发明时,发明人发现,很多时候,升级仅是针对的一部分业务或功能,而上述升级方式,使得系统不能继续向用户提供其他不相关的业务或功能。

[0023] 为解决上述问题,本发明实施例提供一种系统,所述系统被预先划分为可动态加/卸载的多个模块,任一模块通过预设接口与本模块之外的模块或系统进行数据交互,所述系统包括过滤器和升级单元,所述过滤器包括接收单元,所述升级单元在所述接收单元接收到升级指令后,获取包括升级方案的升级包,并根据所述升级方案进行升级。

[0024] 当然,过滤器可为动态加/卸载的某一模块的组成部分,也可为几个可动态加/卸载的模块的集合,也可为一个可动态加/卸载的模块,同理,升级单元也可作此类推。

[0025] 与之相对应,基于上述系统,本发明实施例同时还提供了一种系统升级方法,参见图1,上述方法至少包括:

[0026] S1、接收到升级指令;

[0027] S2、获取升级包,所述升级包中包括升级方案;

[0028] S3、根据所述升级方案进行升级。

[0029] 可以看出,在本发明实施例所提供的技术方案中,系统被预先划分为可动态加/卸载的多个模块,以便后续升级时,可直接加载新模块或重新升级模块。在设计时,可从最大程度得分解不同业务的处理逻辑,更充分降低模块间的耦合度,提高模块内聚性入手进行模块划分。这样,当需要对某些功能或业务进行升级时,可仅对与之相关的模块进行升级,而其他模块还可提供正常的业务或功能。

[0030] 需要说明的是,在将系统划分为可动态加/卸载的多个模块时,可视系统的不同作不同的划分。例如,一个简单的图书管理系统,可以划分为用户信息管理(读者注册、登录等服务)、图书信息管理(图书浏览、图书搜索等功能)、系统维护操作(网站维护、数据维护)三个模块。

[0031] 下面将以 Web 应用为例对本发明的技术方案进行介绍。

[0032] 对 Web 应用来说,过滤器是一个驻留在服务器端的 Web 组件,它可以截取客户端和资源之间的请求与响应信息,并对这些信息进行过滤。

[0033] 通常,当服务器端接受一个对资源的请求(即客户端请求)时,它将判断是否有过滤器与这个资源相关联。如果有,那么服务器端将把请求交给过滤器进行处理。

[0034] 过滤器中可以改变请求的内容,或者重新设置请求的报头信息,然后再将请求发送给目标资源。当目标资源对请求作出响应时,服务器端同样会将响应先转发给过滤器,而过滤器也可以对响应的内容进行转换,然后再将响应发送到客户端。

[0035] 图 2 示出了过滤器的一般工作流程:当一个客户端请求(经常是浏览器请求网页)来时,Web 服务器交给过滤器 1 处理,过滤器 1 处理后将客户端请求进一步提交给目标处理器 2,目标处理器 2 处理完后,再交由过滤器 1 处理,最后将响应信息返回给请求端(浏览器)。

[0036] 在本实施例中,可将上述 Web 系统划分为可动态加/卸载的多个模块,不同的客户端请求可能需要一个或多个上述模块进行处理。在本实施例中,上述升级包可包括升级受限请求信息,该升级受限请求信息包括本次升级业务处理受限的客户端请求的相关信息。上述在本次升级中业务处理受限的客户端请求的相关信息既可直接为客户端请求,可也为在本次升级中业务处理受限的客户端请求对应的 URL 地址。

[0037] 而上述系统的过滤器还可包括处理单元,以根据上述升级受限请求信息对接收到的客户端请求进行处理。

[0038] 与之相应,参见图 3,上述方法还可包括如下步骤:

[0039] S4、根据所述升级受限请求信息对接收到的客户端请求进行处理。

[0040] 其中,步骤 S4 的一种具体实现方式可为:

[0041] 当接收到的客户端请求与所述升级受限请求信息相匹配时,返回提示正在升级的提示信息,否则对所述客户端请求进行正常处理。

[0042] 这样,当用户通过客户端发送客户端请求后,如果处理该请求的模块正处于升级状态,那么将向用户返回正在升级的提示信息,而如果处理该请求的模块未处于升级状态,则可进行正常处理,从而保证可向用户提供正常的业务或功能。

[0043] 在本发明其他实施例中,上述升级方案至少包括本次升级所涉及的模块的信息,而当本次升级所涉及的模块为多个时,所述升级方案还包括模块的升级先后顺序。

[0044] 之所以提及升级先后顺序,是因为有些待升级的模块之间可能存在依赖关系。比如模块 A 依赖于模块 B,并且两个模块在本次升级中,需要先卸载再重新加载,那么在升级过程中,需要先卸载模块 A,再卸载模块 B,而重新加载则是相反的过程,需要先加载模块 B 然后再操作模块 A。当然,如果需升级的模块间没依赖关系,可不分升级先后顺序。而不分升级先后顺序,也可视为一种特殊的升级先后顺序。

[0045] 上述升级受限请求信息可以文件(比如 word 文档,列表)等方式存在。下面将以升级受限请求列表为例进行介绍,在升级受限请求列表中,记载有升级受限的 URL 地址。

[0046] 在本实施例中,上述升级方法流程如下:

[0047] Web 系统提供升级服务,比如提供升级页面。管理员点击升级下达升级指令后,升级单元到指定的路径下读取升级包,根据升级包中的升级方案进行升级。

[0048] 在升级前,可先设置升级标识(例如在某个目录下生成 update.identifier 文件),升级完成后清除升级标识(如上例,删除 update.Identifier 文件)。

[0049] 当过滤器的接收单元接收到客户端请求时,其处理单元首先判断升级标识是否设置,如果没有设置,则继续往下递交客户端请求;如已设置,则获取升级包中的“升级受限请求列表”,判断请求的 URL 地址是否在该列表中,如果在,则重定向到错误处理页面,如果不在则继续往下递交请求。

[0050] 上述错误处理页面可以根据业务场景来定制。例如,图书管理系统的错误处理页面可以展示“本图书管理系统正在升级中,请稍后访问”的信息。

[0051] 处理单元对于上述升级受限请求列表的获取可以是主动获取也可以是被动获取。例如,处理单元可在判断出升级标识已设置时,读取升级包中的升级受限请求列表,这是主动获取。另外,在进行升级时,升级单元也可从升级包中提取升级受限请求列表,并将其提供给处理单元。这种方式为被动获取。

[0052] 在将系统预先划分为各个模块时,各模块之间,模块与系统接口之间,模块与客户端请求之间的关系已经被预设,所以在设计升级包时,可通过上述预设关系,生成升级方案。

[0053] 在本发明其他实施例中,升级受限请求信息和升级先后顺序的确定可包括如下步骤:

[0054] 确定本次升级所涉及的模块;

[0055] 当本次升级所涉及的模块为多个时,根据所涉及的各模块之间的预设关系,来设定模块的升级先后顺序;

[0056] 根据本次升级所涉及的模块与客户端请求的预设关系,生成升级方案。

[0057] 对升级受限请求信息和升级先后顺序的确定还可有其他方式:例如:

[0058] 评估本次升级所波及的系统接口;

[0059] 根据本次升级所波及的系统接口与客户端请求的预设关系,生成升级受限请求信息;

[0060] 根据所述系统接口与系统各模块的预设关系,确定本次升级所涉及的模块的信息;

[0061] 当本次升级所涉及的模块为多个时,根据所涉及的各模块之间的预设关系来设定模块的升级先后顺序。

[0062] 举一个更具体的例子,比如图书管理系统要新版本提供新特性-用户信息管理能删除用户信息,这就需在升级包中开发此功能的代码。

[0063] 为了给新版本提供新的功能或功能修改,在开发升级包时,将提供新的模块、或给原模块新增或修改代码。此时,可确定受新版本代码影响的原版本相关接口(也即系统接口),由模块内的相关类、函数等修改可评估出模块的相关接口是否受影响。例如 java 的 struts 框架中,Model 层提供接口给 Action(Control 层)或 JSP(View 层)调用,需评估出这些接口中哪些受新代码影响。

[0064] 在评估时,可以根据方法(或类、函数,下面只以方法为例)的依赖关系来实现:如果方法 A 调用了方法 B,那么方法 A 就依赖于 B,如果方法 B 又调用了方法 C,那么方法 A 也间接依赖于方法 C。对于这些 Model 层接口,寻找依赖的所有接口实现或方法是否有变化,

只要有一个依赖的方法修改,这个接口就受影响。这种评估方式粒度比较粗,当一个方法的调用者多时,此方法的修改可能会引发大范围的方法受影响,好处在于比较简单、容易操作。

[0065] 另外,还可用其他评估方式:比如方法 B 调用方法 C,使用上面的评估方法一,当 C 有改动时,方法 B 受影响,但可能方法 C 只是方法 B 中的一个分支流程,其它分支是可以正常运行的,有些业务请求调用方法 B,但不会调用方法 C。

[0066] 因此,可在对每个方法调用前,先判断本方法是否有修改,如果有修改,则抛出错误异常,如果没有修改,则正常调用此方法。

[0067] 至于如何判断本方法是否有修改,实现方式可以多样,比如根据本方法相关的文件是否有更新来判断,或开发人员直接将类名、方法名等配置到文件中,当升级时通过读取此文件判断本方法是否有修改。

[0068] 与之相对应,本发明实施例以上的所有系统,还可包括评估单元,用于确定升级受限请求信息和升级先后顺序。比如评估单元可评估本次升级所波及的系统接口,根据本次升级所波及的系统接口与客户端请求的预设关系,生成升级受限请求信息,根据所述系统接口与系统各模块的预设关系,确定本次升级所涉及的模块的信息,以及当本次升级所涉及的模块为多个时,根据所涉及各模块之间的预设关系来设定模块的升级先后顺序。

[0069] 本说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似部分互相参见即可。对于实施例公开的装置而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。

[0070] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的程序可存储于一计算机可读取存储介质中,所述程序在执行时,可包括如上述各方法的实施例的流程。其中,所述的存储介质可为磁碟、光盘、只读存储记忆体 (Read-Only Memory, ROM) 或随机存储记忆体 (Random Access Memory, RAM) 等。

[0071] 对所公开的实施例的上述说明,使本领域专业技术人员能够实现或使用本发明。对这些实施例的多种修改对本领域的专业技术人员来说将是显而易见的,本文中所定义的一般原理可以在不脱离本发明的精神或范围的情况下,在其它实施例中实现。因此,本发明将不会被限制于本文所示的这些实施例,而是要符合与本文所公开的原理和新颖特点相一致的最宽的范围。



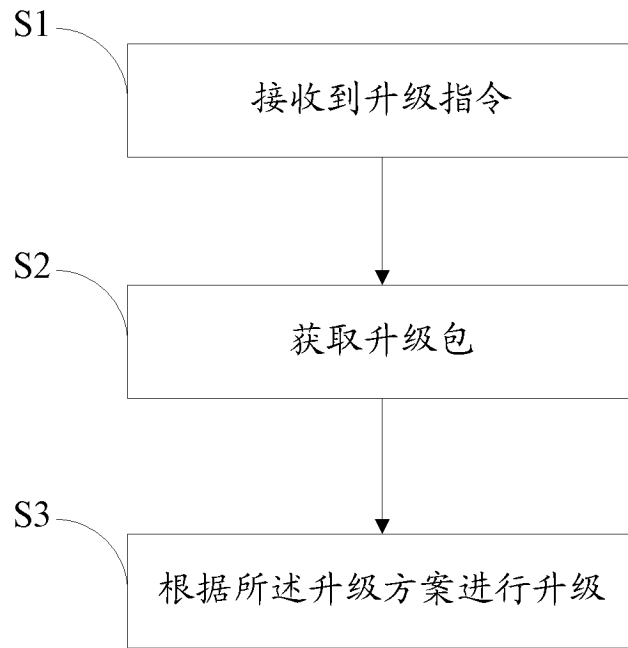


图 1

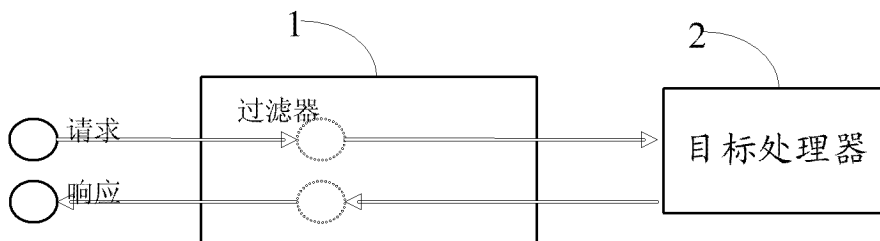


图 2

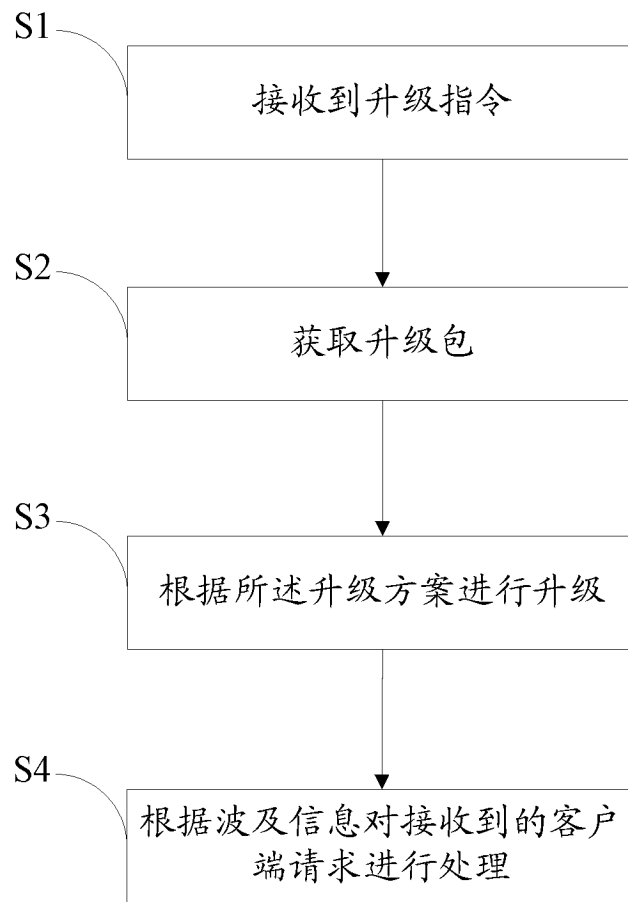


图 3