



(12)发明专利

(10)授权公告号 CN 105630470 B

(45)授权公告日 2018.12.21

(21)申请号 201410608412.3

(22)申请日 2014.10.31

(65)同一申请的已公布的文献号

申请公布号 CN 105630470 A

(43)申请公布日 2016.06.01

(73)专利权人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼资本大厦一座四层847号邮箱

(72)发明人 吴振华

(74)专利代理机构 上海百一领御专利代理事务所(普通合伙) 31243

代理人 陈贞健

(51)Int.Cl.

G06F 8/30(2018.01)

(56)对比文件

CN 104090776 A,2014.10.08,

CN 103534684 A,2014.01.22,

CN 103593173 A,2014.02.19,

CN 102207866 A,2011.10.05,

CN 101042645 A,2007.09.26,

WO 2011082627 A1,2011.07.14,

US 6487713 B1,2002.11.26,

US 2006150144 A1,2006.07.06,

US 2013080995 A1,2013.03.28,

审查员 于潇

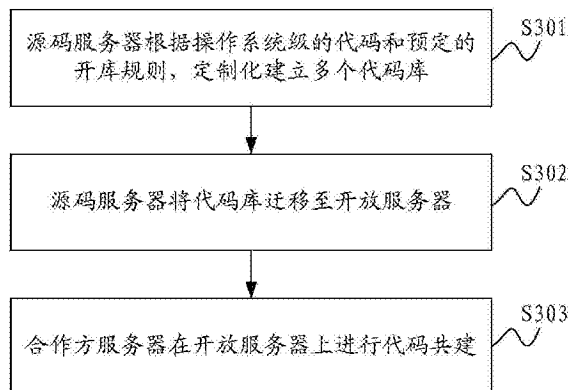
权利要求书3页 说明书10页 附图3页

(54)发明名称

代码共建方法、系统、源码服务器及开放服务器

(57)摘要

本申请提供了一种代码共建方法、系统、源码服务器以及开放服务器,所述源码服务器和合作方服务器能够访问所述开放服务器,所述开放服务器和所述合作方服务器不能访问所述源码服务器;所述代码共建方法包括步骤有:所述源码服务器根据操作系统级的代码和预定的开库规则,定制化建立多个代码库;所述源码服务器将所述代码库迁移至所述开放服务器;所述合作方服务器在所述开放服务器上进行代码共建。借此,本申请能够满足多个公司对于操作系统代码库结构的指定需求,可增强移植性及完善共建效果,提高效率并减小成本;并且提高了代码共建的安全性,且交互更加高效。



1. 一种源码服务器,其特征在于,所述源码服务器与开放服务器通信连接,所述开放服务器和合作方服务器通信连接,所述源码服务器和所述合作方服务器能够访问所述开放服务器,所述开放服务器和所述合作方服务器不能访问所述源码服务器;所述源码服务器包括:

代码库定制模块,用于根据操作系统级的代码和预定的开库规则,定制化建立多个代码库;

代码库处理模块,用于将所述代码库迁移至所述开放服务器,供所述合作方服务器在所述开放服务器上进行代码共建。

2. 根据权利要求1所述的源码服务器,其特征在于,所述代码库定制模块进一步包括:

代码库定制子模块,用于根据所述开库规则,对操作系统级的代码包进行扫描,建立起包括多个所述代码库的代码库结构,生成描述所述代码库结构的索引文件,并将所述代码包中文件创建并上传到所述代码库中。

3. 根据权利要求1所述的源码服务器,其特征在于,所述开库规则根据所述源码服务器和/或所述合作方服务器对代码库结构的指定需求制定。

4. 根据权利要求1所述的源码服务器,其特征在于,所述代码库处理模块进一步包括:

处理子模块,用于将操作系统级的所述代码进行适配、二进制化集成并测试;

第一检查子模块,用于若测试通过后,将所述代码进行安全检查;

迁移子模块,用于若代码安全检查通过后,将所述代码库迁移至所述开放服务器。

5. 根据权利要求1所述的源码服务器,其特征在于,所述源码服务器设于内部网络,所述开放服务器设于对外开放网络,所述合作方服务器设于互连网络;所述内部网络和所述互连网络能够访问所述对外开放网络,所述对外开放网络和所述互连网络不能访问所述内部网络。

6. 根据权利要求1所述的源码服务器,其特征在于,所述源码服务器的所述代码库定制模块进一步包括:

扫描子模块,用于扫描所述代码及日志获取每个所述代码库的数据量、变更频率和读取频率;

计算子模块,用于根据每个所述代码库的所述数据量、所述变更频率、所述读取频率和预定的代码库拆分值算法,计算出所述代码库的拆分值;

拆库子模块,用于根据预定的需求库总数,以及保证每个所述代码库的拆分值相近的拆分规则,对所述代码库进行拆库。

7. 根据权利要求6所述的源码服务器,其特征在于,所述代码库拆分值算法为:

代码库的拆分值 $G = p_c * (S * VC) + p_d * (S * VD)$;

其中 p_c 和 p_d 分别为加权值, S 为当前代码库的数据量, VC 为当前代码库的数据变更频率, VD 为当前代码库的数据读取频率。

8. 一种开放服务器,其特征在于,所述开放服务器分别与源码服务器和合作方服务器通信连接,所述源码服务器和所述合作方服务器能够访问所述开放服务器,所述开放服务器和所述合作方服务器不能访问所述源码服务器;所述开放服务器包括:

代码存储模块,用于接收并存储所述源码服务器根据操作系统级的代码和预定的开库规则所定制化建立的多个代码库,供所述合作方服务器在所述开放服务器上进行代码共

建。

9. 根据权利要求8所述的开放服务器,其特征在于,所述开放服务器进一步包括:

第二检查模块,用于对存储到所述开放服务器中的所述代码库中的所述代码进行代码安全检查;

代码库开放模块,用于若代码安全检查通过后,将所述开放服务器中的所述代码库进行开放。

10. 根据权利要求8所述的开放服务器,其特征在于,所述开放服务器为所述合作方服务器提供指定端口的访问权限。

11. 根据权利要求10所述的开放服务器,其特征在于,所述开放服务器创建授权账号,所述授权账号对应所述开放服务器的所述指定端口的访问权限;所述合作方服务器通过所述授权账号访问所述开放服务器的所述指定端口进行代码共建。

12. 根据权利要求8所述的开放服务器,其特征在于,所述源码服务器设于内部网络,所述开放服务器设于对外开放网络,所述合作方服务器设于互互联网;所述内部网络和所述互互联网能够访问所述对外开放网络,所述对外开放网络和所述互互联网不能访问所述内部网络。

13. 一种代码共建系统,其特征在于,所述代码共建系统包括有如权利要求1~7任一项所述的源码服务器、如权利要求8~12任一项所述的开放服务器以及所述合作方服务器。

14. 一种代码共建方法,其特征在于,所述代码共建方法应用于包括源码服务器、开放服务器和至少一合作方服务器的代码共建系统中,所述源码服务器和所述合作方服务器能够访问所述开放服务器,所述开放服务器和所述合作方服务器不能访问所述源码服务器;所述代码共建方法包括步骤有:

所述源码服务器根据操作系统级的代码和预定的开库规则,定制化建立多个代码库;

所述源码服务器将所述代码库迁移至所述开放服务器;

所述合作方服务器在所述开放服务器上进行代码共建。

15. 根据权利要求14所述的代码共建方法,其特征在于,所述源码服务器根据操作系统级的代码和开库规则,定制化建立多个代码库的步骤包括:

所述源码服务器根据所述开库规则,对操作系统级的代码包进行扫描,建立起包括多个所述代码库的代码库结构,生成用于描述所述代码库结构的索引文件,并将所述代码包中文件创建并上传到所述代码库中。

16. 根据权利要求14所述的代码共建方法,其特征在于,所述开库规则根据所述源码服务器和/或所述合作方服务器对代码库结构的指定需求制定。

17. 根据权利要求14所述的代码共建方法,其特征在于,所述源码服务器将所述代码库迁移至所述开放服务器的步骤包括:

将操作系统级的所述代码进行适配、二进制化集成并测试;

若测试通过后,将所述代码进行安全检查;

若代码安全检查通过后,将所述代码库迁移至所述开放服务器。

18. 根据权利要求17所述的代码共建方法,其特征在于,所述将代码库迁移至所述开放服务器的步骤之后包括:

所述开放服务器对所述代码库中的所述代码进行代码安全检查;

若代码安全检查通过后,将所述开放服务器中的所述代码库进行开放。

19. 根据权利要求14所述的代码共建方法,其特征在于,所述开放服务器为合作方服务器提供指定端口的访问权限。

20. 根据权利要求19所述的代码共建方法,其特征在于,所述开放服务器创建授权账号,所述授权账号对应所述开放服务器的所述指定端口的访问权限;

所述合作方服务器在所述开放服务器上进行代码共建的步骤包括:

所述合作方服务器通过所述授权账号访问所述开放服务器的所述指定端口进行代码共建。

21. 根据权利要求14所述的代码共建方法,其特征在于,所述源码服务器设于内部网络,所述开放服务器设于对外开放网络,所述合作方服务器设于互联网络;所述内部网络和所述互联网络能够访问所述对外开放网络,所述对外开放网络和所述互联网络不能访问所述内部网络。

22. 根据权利要求14~21任一项所述的代码共建方法,其特征在于,所述源码服务器根据操作系统级的代码和预定的开库规则,定制化建立多个代码库的步骤之后还包括:

扫描所述代码及日志获取每个所述代码库的数据量、变更频率和读取频率;

根据每个所述代码库的所述数据量、所述变更频率、所述读取频率和预定的代码库拆分值算法,计算出所述代码库的拆分值;

根据预定的需求库总数,以及保证每个所述代码库的拆分值相近的拆分规则,对所述代码库进行拆库。

23. 根据权利要求22所述的代码共建方法,其特征在于,所述代码库拆分值算法为:

代码库的拆分值 $G = pc * (S * VC) + pd * (S * VD)$;

其中 pc 和 pd 分别为加权值, S 为当前代码库的数据量, VC 为当前代码库的数据变更频率, VD 为当前代码库的数据读取频率。

代码共建方法、系统、源码服务器及开放服务器

技术领域

[0001] 本申请涉及互联网技术领域,尤其涉及一种代码共建方法、系统、源码服务器以及开放服务器。

背景技术

[0002] 目前业内的操作系统级开放代码,采取开源社区模式,如谷歌(google)的安卓(Android)开源社区。所述开源社区模式解决了单向开放的问题,但无法适应解决代码共建的需求。开放代码是指公司A将自己的代码开放给公司B使用。而代码共建属于开放代码的一种模式,公司A将自己的代码开放给公司B使用,公司B将代码修改后提供给公司A。而操作系统级的代码共建,其模式主要为线上或线下代码同步(如邮件、硬盘、U盘),即公司A将自己的代码打包,发送给公司B,公司B基于公司A的代码进行研发,再同步给公司A,然后产出产品。

[0003] 目前操作系统的代码共建技术,主要存在以下几个问题:

[0004] 安全性问题:如果采取邮件、网盘等线上代码同步模式,若代码在同步过程中被劫持或被第三方网盘公司获取,这对于在操作系统上的架构上创新及新功能的保护是极其不利的,代码一旦泄露带来的损失巨大。

[0005] 效率性问题:如果采用硬盘等线下代码同步模式,虽然代码安全得到了保证,但是效率过低,每次同步代码带来的成本过高,在互联网的高速发展情况下很难满足业务需求。

[0006] 交互性及成本问题:传统模式中,虽然基于同一套代码进行研发,但是由于双方代码管理工具及建库模式上的差异,两公司的研发人员交互效率低下,无法实现提交代码的即时性和研发人员对于彼此修改的说明都不能很明晰的展现出来。这对于操作系统级代码的研发无疑增加很大的成本。

[0007] 综上可知,现有技术在实际使用上显然存在不便与缺陷,所以有必要加以改进。

发明内容

[0008] 针对上述的缺陷,本申请的目的在于提供一种代码共建方法、系统、源码服务器以及开放服务器,其通过定制化建立代码库,能够满足多个公司对于操作系统代码库结构的指定需求,可增强移植性及完善共建效果,提高效率并减小成本;并且提高了代码共建的安全性,且交互更加高效。

[0009] 为了实现上述目的,本申请提供一种源码服务器,所述源码服务器与开放服务器通信连接,所述开放服务器和合作方服务器通信连接,所述源码服务器和所述合作方服务器能够访问所述开放服务器,所述开放服务器和所述合作方服务器不能访问所述源码服务器;所述源码服务器包括:

[0010] 代码库定制模块,用于根据操作系统级的代码和预定的开库规则,定制化建立多个代码库;

[0011] 代码库处理模块,用于将所述代码库迁移至所述开放服务器,供所述合作方服务

器在所述开放服务器上进行代码共建。

[0012] 根据本申请所述的源码服务器,所述代码库定制模块进一步包括:

[0013] 代码库定制子模块,用于根据所述开库规则,对操作系统级的代码包进行扫描,建立起包括多个所述代码库的代码库结构,生成描述所述代码库结构的索引文件,并将所述代码包中文件创建并上传到所述代码库中。

[0014] 根据本申请所述的源码服务器,所述开库规则根据所述源码服务器和/或所述合作服务器对代码库结构的指定需求制定。

[0015] 根据本申请所述的源码服务器,所述代码库处理模块进一步包括:

[0016] 处理子模块,用于将操作系统级的所述代码进行适配、二进制化集成并测试;

[0017] 第一检查子模块,用于若测试通过后,将所述代码进行安全检查;

[0018] 迁移子模块,用于若代码安全检查通过后,将所述代码库迁移至所述开放服务器。

[0019] 根据本申请所述的源码服务器,所述源码服务器设于内部网络,所述开放服务器设于对外开放网络,所述合作方服务器设于互连网络;所述内部网络和所述互连网络能够访问所述对外开放网络,所述对外开放网络和所述互连网络不能访问所述内部网络。

[0020] 根据本申请所述的源码服务器,所述源码服务器的所述代码库定制模块进一步包括:

[0021] 扫描子模块,用于扫描所述代码及日志获取每个所述代码库的数据量、变更频率和读取频率;

[0022] 计算子模块,用于根据每个所述代码库的所述数据量、所述变更频率、所述读取频率和预定的代码库拆分值算法,计算出所述代码库的拆分值;

[0023] 拆库子模块,用于根据预定的需求库总数,以及保证每个所述代码库的拆分值相近的拆分规则,对所述代码库进行拆库。

[0024] 根据本申请所述的源码服务器,所述代码库拆分值算法为:

[0025] 代码库的拆分值 $G = pc * (S * VC) + pd * (S * VD)$;

[0026] 其中 pc 和 pd 分别为加权值, S 为当前代码库的数据量, VC 为当前代码库的数据变更频率, VD 为当前代码库的数据读取频率。

[0027] 本申请还提供一种开放服务器,所述开放服务器分别与源码服务器和合作方服务器通信连接,所述源码服务器和所述合作方服务器能够访问所述开放服务器,所述开放服务器和所述合作方服务器不能访问所述源码服务器;所述开放服务器包括:

[0028] 代码存储模块,用于接收并存储所述源码服务器根据操作系统级的代码和预定的开库规则所定制化建立的多个代码库,供所述合作方服务器在所述开放服务器上进行代码共建。

[0029] 根据本申请所述的开放服务器,所述开放服务器进一步包括:

[0030] 第二检查模块,用于对存储到所述开放服务器中的所述代码库中的所述代码进行代码安全检查;

[0031] 代码库开放模块,用于若代码安全检查通过后,将所述开放服务器中的所述代码库进行开放。

[0032] 根据本申请所述的开放服务器,所述开放服务器为所述合作方服务器提供指定端口的访问权限。

[0033] 根据本申请所述的开放服务器,所述开放服务器创建授权账号,所述授权账号对应所述开放服务器的所述指定端口的访问权限;所述合作方服务器通过所述授权账号访问所述开放服务器的所述指定端口进行代码共建。

[0034] 根据本申请所述的开放服务器,所述源码服务器设于内部网络,所述开放服务器设于对外开放网络,所述合作方服务器设于互连网络;所述内部网络和所述互连网络能够访问所述对外开放网络,所述对外开放网络和所述互连网络不能访问所述内部网络。

[0035] 本申请还提供一种代码共建系统,包括所述源码服务器、所述开放服务器以及所述合作方服务器。

[0036] 本申请还提供一种代码共建方法,所述代码共建方法应用于包括源码服务器、开放服务器和至少一合作方服务器的代码共建系统中,所述源码服务器和所述合作方服务器能够访问所述开放服务器,所述开放服务器和所述合作方服务器不能访问所述源码服务器;所述代码共建方法包括步骤有:

[0037] 所述源码服务器根据操作系统级的代码和预定的开库规则,定制化建立多个代码库;

[0038] 所述源码服务器将所述代码库迁移至所述开放服务器;

[0039] 所述合作方服务器在所述开放服务器上进行代码共建。

[0040] 根据本申请所述的代码共建方法,所述源码服务器根据操作系统级的代码和开库规则,定制化建立多个代码库的步骤包括:

[0041] 所述源码服务器根据所述开库规则,对操作系统级的代码包进行扫描,建立起包括多个所述代码库的代码库结构,生成用于描述所述代码库结构的索引文件,并将所述代码包中文件创建并上传到所述代码库中。

[0042] 根据本申请所述的代码共建方法,所述开库规则根据所述源码服务器和/或所述合作服务器对代码库结构的指定需求制定。

[0043] 根据本申请所述的代码共建方法,所述源码服务器将所述代码库迁移至所述开放服务器的步骤包括:

[0044] 将操作系统级的所述代码进行适配、二进制化集成并测试;

[0045] 若测试通过后,将所述代码进行安全检查;

[0046] 若代码安全检查通过后,将所述代码库迁移至所述开放服务器。

[0047] 根据本申请所述的代码共建方法,所述将代码库迁移至所述开放服务器的步骤之后包括:

[0048] 所述开放服务器对所述代码库中的所述代码进行代码安全检查;

[0049] 若代码安全检查通过后,将所述开放服务器中的所述代码库进行开放。

[0050] 根据本申请所述的代码共建方法,所述开放服务器为合作方服务器提供指定端口的访问权限。

[0051] 根据本申请所述的代码共建方法,所述开放服务器创建授权账号,所述授权账号对应所述开放服务器的所述指定端口的访问权限;

[0052] 所述合作方服务器在所述开放服务器上进行代码共建的步骤包括:

[0053] 所述合作方服务器通过所述授权账号访问所述开放服务器的所述指定端口进行代码共建。

[0054] 根据本申请所述的代码共建方法,所述源码服务器设于内部网络,所述开放服务器设于对外开放网络,所述合作方服务器设于互联网络;所述内部网络和所述互联网络能够访问所述对外开放网络,所述对外开放网络和所述互联网络不能访问所述内部网络。

[0055] 根据本申请所述的代码共建方法,所述源码服务器根据操作系统级的代码和预定的开库规则,定制化建立多个代码库的步骤之后还包括:

[0056] 扫描所述代码及日志获取每个所述代码库的数据量、变更频率和读取频率;

[0057] 根据每个所述代码库的所述数据量、所述变更频率、所述读取频率和预定的代码库拆分值算法,计算出所述代码库的拆分值;

[0058] 根据预定的需求库总数,以及保证每个所述代码库的拆分值相近的拆分规则,对所述代码库进行拆库。

[0059] 根据本申请所述的代码共建方法,所述代码库拆分值算法为:

[0060] 代码库的拆分值 $G = pc * (S * VC) + pd * (S * VD)$;

[0061] 其中 pc 和 pd 分别为加权值, S 为当前代码库的数据量, VC 为当前代码库的数据变更频率, VD 为当前代码库的数据读取频率。

[0062] 本申请提供操作系统级的代码共建技术方案,由源码服务器根据操作系统级的代码和预定的开库规则,定制化建立多个代码库,并将所述代码库迁移至开放服务器,合作方服务器在开放服务器上进行代码共建。借此,本申请通过定制化建立代码库,能够满足多个公司对于操作系统代码库结构的指定需求,可增强移植性及完善共建效果,提高效率并减小成本;并且,由于本申请采用服务器点对点单向授权共建方法,大幅度减少代码中途泄露的可能性,提高了代码共建的安全性;另外,由于本申请中多个公司之间在同一台开放服务器上进行研发,其实现效果与在同公司研发效果类似,因此交互更加高效。

附图说明

[0063] 图1是本申请代码共建系统的结构示意图;

[0064] 图2是本申请优选代码共建系统的结构示意图;

[0065] 图3是本申请代码共建方法的流程图;

[0066] 图4是本申请优选代码共建方法的流程图。

具体实施方式

[0067] 为了使本申请的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本申请进行进一步详细说明。应当理解,此处所描述的具体实施例仅仅用以解释本申请,并不用于限定本申请。

[0068] 在本申请一个典型的配置中,终端、服务网络的设备和可信方均包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[0069] 内存可能包括计算机可读介质中的非永久性存储器,随机存取存储器

[0070] (RAM)和/或非易失性内存等形式,如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

[0071] 计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。

计算机的存储介质的例子包括,但不限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器 (CD-ROM)、数字多功能光盘 (DVD) 或其他光学存储、磁盒式磁带,磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质,可用于存储可以被计算设备访问的信息。按照本文中的界定,计算机可读介质不包括非暂存电脑可读媒体 (transitory media),如调制的信号和载波。

[0072] 图1是本申请代码共建系统的结构示意图,所述代码共建系统100包括源码服务器10、开放服务器20和至少一合作方服务器30,所述源码服务器10和合作方服务器30能够访问开放服务器20,所述开放服务器20和合作方服务器30不能访问源码服务器10,即所述开放服务器20采用单向网络隔离的模式。优选的是,源码服务器10设于内部网络,开放服务器20设于对外开放网络,合作方服务器30设于互连网络 (Internet),此实现各服务器之间的网域隔离,并且内部网络和互连网络能够访问对外开放网络,所述对外开放网络和互连网络不能访问内部网络。

[0073] 本申请提供一种源码服务器10,所述源码服务器10与开放服务器20通信连接,开放服务器20和合作方服务器30通信连接,源码服务器10和合作方服务器30能够访问开放服务器20,开放服务器20和合作方服务器30不能访问源码服务器10;所述源码服务器10包括:

[0074] 代码库定制模块11,用于根据操作系统级的代码和预定的开库规则,定制化建立多个代码库。通常,每个操作系统版本约300个代码库。所述开库规则根据源码服务器10和/或合作服务器30对代码库结构的指定需求制定,从而满足各个公司对于操作系统代码库结构的指定需求。

[0075] 代码库处理模块12,用于将代码库迁移至开放服务器20。

[0076] 所述合作方服务器30,用于在开放服务器20上进行代码共建。这里,合作方服务器30直接在开放服务器20上将代码修改,源码服务器10可以直接获取修改后的代码。

[0077] 在本申请具体实施例中,其实现方案为:基于基础代码包,进行定制化建立代码库(每个操作系统版本约300个小库),这些小库基于开库规则进行库深度及广度进行扫描,建立起代码库结构,生成索引文件,并基于代码包中文件批量创建并上传到代码库中,整个操作系统代码库创建内部代码完毕;该代码在内部研发人员进行适配后,进行二进制化集成,测试完毕后,进行代码开放;代码开放前,进行自动化安全检查,即扫描共建库中是否存在不想开放的源码文件,通过后,将整个库自动化迁移至开放服务器20;在开放服务器20上进行进一步自动化代码安全检查,校验法务协议是否签署,若均已经通过,进入下一步;创建开放厂家的账号,打通开放服务器20,合作方服务器30的外部研发人员用该帐号与内部研发人员进行代码共建,整个代码管理可采用已经有的开源工具。

[0078] 本申请解决多个公司操作系统代码的代码共建问题。由于操作系统具有与其它代码不具备的特性,安全性高、代码量大、代码结构复杂多变(芯片型号多样),主要解决以下问题:1、代码合作涉及的安全性及高效性问题;2、操作系统代码量大及与芯片个性化导致代码结构复杂的问题。传统的建库,只是根据需求创建库,进行代码管理。本申请不仅实现了自动化,并根据具体的使用情况进行建库,使得研发人员更精确的拿到所需的代码,大幅度减少下载及上传时间,提高研发效率。

[0079] 图2是本申请优选代码共建系统的结构示意图,所述代码共建系统100包括源码服

务器10、开放服务器20和至少一合作方服务器30,源码服务器10和合作方服务器30能够访问开放服务器20,开放服务器20和合作方服务器30不能访问源码服务器10。优选的是,源码服务器10设于内部网络,开放服务器20设于对外开放网络,合作方服务器30设于互联网络。内部网络和互联网络能够访问对外开放网络,所述对外开放网络和互联网络不能访问内部网络。

[0080] 本申请提供一种源码服务器10,所述源码服务器10与开放服务器20通信连接,开放服务器20和合作方服务器30通信连接,源码服务器10和合作方服务器30能够访问开放服务器20,开放服务器20和合作方服务器30不能访问源码服务器10;所述源码服务器10包括:

[0081] 代码库定制模块11,用于根据操作系统级的代码和预定的开库规则,定制化建立多个代码库。所述开库规则根据源码服务器10和/或合作服务器30对代码库结构的指定需求制定,从而满足各个公司对于操作系统代码库结构的指定需求。

[0082] 所述代码库定制模块11进一步包括:

[0083] 代码库定制子模块111,用于根据开库规则,对操作系统级的代码包进行库深度及库广度扫描,建立起包括多个代码库的代码库结构,生成索引文件,并将代码包中文件批量创建并上传到代码库中。

[0084] 扫描子模块112,用于扫描代码及日志获取每个代码库的数据量、变更频率和读取频率。

[0085] 计算子模块113,用于根据每个代码库的数据量、变更频率、读取频率和预定的代码库拆分值算法,计算出代码库的拆分值。

[0086] 优选的是,所述代码库拆分值算法为:

[0087] 代码库的拆分值 $G = pc * (S * VC) + pd * (S * VD)$,其中pc和pd分别为加权值,S为当前代码库的数据量,VC为当前代码库的数据变更频率,VD为当前代码库的数据读取频率。

[0088] 拆库子模块114,用于根据预定的需求库总数,以及保证每个代码库的拆分值相近的拆分规则,对代码库进行拆库。

[0089] 本申请定制化技术针对整个代码包,采用权值计算的方法进行扫描,最后得出一个各代码库权值均衡的定制化方案,再根据需求库总数量进行库的拆分值选取。

[0090]

一级属性	数据量	数据变更频率	数据读取频率
符号	S	VC	VD
说明	当前目录(即当前代码库)的数据总量	当前目录的文件变更次数	当前目录的文件下载次数

二级属性	变更度	读取度
计算方法	$S * VC$	$S * VD$
加权值	pc	pd
说明	当前目录的数据变更度	当前目录的数据读取度

[0091] 注:权值可根据具体的使用情况进行微调

	三级属性	拆分值
[0092]	计算方法	$G=pc*(S*VC)+pd*(S*VD)$
	说明	每个目录具有一个拆分值

[0093] 拆库规则:正如上面三级属性的说明,每个目录会根据数据量、变更频率、读取频率产生一个拆分值,在拆库时,保证每个代码库的拆分值相等的规则进行。每个拆分值与拆分数目近似成反比。

[0094] 本申请自动化实现扫描代码及log(日志)获取数据量、变更频率、读取频率,并进行计算得出拆分值,并根据已知要拆的数量,进行定制化拆库。

[0095] 下面描述一个简单实现单元实例:该实例只是作为原理简述用,实际拆分远比这里复杂的多。

[0096] |-- 公司A

[0097] | |--e(10,5,10)

[0098] | `--f(5,15,10)

[0099] |-- 公司B

[0100] | |--g(3,2,8)

[0101] | `--h(20,3,2)

[0102] 上表是一个简单的目录树形结构,后面“()”中的属性分别为数据量、变更频率、读取频率。本次模拟采用 $pc=2/3$, $pd=1/3$,计算得, $G(\text{公司Ae})=200/3$, $G(\text{公司Af})=200/3$, $G(\text{公司Bg})=36/3$, $G(\text{公司Bh})=160/3$,上一层级目录的拆分为 $G(\text{公司A})=G(\text{公司Ae})+G(\text{公司Af})=400/3$, $G(\text{公司B})=G(\text{公司Bg})+G(\text{公司Bh})=196/3$ 。本次拆分可看出,公司B目录的拆分值与公司A的子目录相近,所以拆分结构为公司Ae、公司Af、公司B三个目录分别作为代码库进行拆分。

[0103] 代码库处理模块12,用于将代码库迁移至开放服务器20。优选的是,所述代码库处理模块12进一步包括:

[0104] 处理子模块121,用于将操作系统级的代码进行适配、二进制化集成并测试。

[0105] 第一检查子模块122,用于若测试通过后,将代码进行安全检查。即扫描共建用的代码库中是否存在不想开放的源码文件,若存在不想开放的源码文件则禁止开放。

[0106] 迁移子模块123,用于若代码安全检查通过后,将代码库迁移至开放服务器20。

[0107] 本申请提供一种开放服务器20,所述开放服务器20分别与源码服务器10和合作方服务器30通信连接,源码服务器10和合作方服务器30能够访问开放服务器20,开放服务器20和合作方服务器30不能访问源码服务器10。所述开放服务器20至少包括:

[0108] 代码存储模块21,用于接收并存储源码服务器10根据操作系统级的代码和预定的开库规则所定制化建立的多个代码库,供合作方服务器30在开放服务器20上进行代码共建。

[0109] 优选的是,所述开放服务器20进一步包括:

[0110] 第二检查模块22,用于对存储到开放服务器20中的代码库中的代码进行代码安全检查。

[0111] 代码库开放模块23,用于若代码安全检查通过后,将开放服务器20中的代码库进行开放。

[0112] 所述合作方服务器30,用于在开放服务器20上进行代码共建。优选的是,开放服务器20为所述合作方服务器30提供指定端口的访问权限。更好的是,开放服务器20创建授权账号,授权账号对应开放服务器20的指定端口的访问权限。合作方服务器30通过授权账号访问开放服务器20的指定端口进行代码共建。

[0113] 本申请采用网域隔离服务器、点对点单向授权共建方法。所谓网域隔离是指在开放服务器20与源码服务器10间采取网域隔离(具体可采用物理隔离或网络隔离均可);单向访问,即服务器间只能单向访问,源码服务器10可访问开放服务器20,开放服务器20不可访问源码服务器10;单向授权,即开放服务器20只对需共建方提供单独认证及端口,使其具有访问该端口指定码段的权限。

[0114] 本申请关键技术点包括:1)基于操作系统级代码库定制化技术及拆库规则;2)基于操作系统级代码共建网域隔离服务器点对点式代码交互方法。本申请具有如下技术效果:

[0115] 安全性高:采取服务器点对点直连的方式,避免了第三方的介入,大幅度减少代码中途泄露的可能性;采取单向授权的指定端口共建模式,即只有授权服务器才能获取服务器的该端口代码,加大了对于服务器安全性;开放服务器20采取单向网络隔离的模式,即使开放服务器20被攻破,仍然不会影响到其他服务器群。

[0116] 高效交互:共建公司之间在同一台开放服务器20上进行研发,每次提交的log(日志)彼此可查看,提交具有即时性,即对于研发人员来说实现效果与在同公司研发效果类似,即时提交即时更新,且节省不必要的成本。

[0117] 定制化技术及拆库规则:由于采取共建于同台服务器的模式,又必须满足每个公司对于操作系统代码库树结构的特殊需求,本申请基于库深度及广度进行解析,定制化批量自动建库。满足各方需求,增强移植性及完善共建效果,提高效率减小成本。同时,本申请除了满足个性化需求,还采取了算法对于拆库进行优化,大幅度提高代码库本身的使用效率。

[0118] 图3是本申请代码共建方法的流程图,代码共建方法应用如图1或图2所示的代码共建系统100中,所述代码共建方法包括步骤有:

[0119] 步骤S301,源码服务器10根据操作系统级的代码和预定的开库规则,定制化建立多个代码库。

[0120] 通常,每个操作系统版本约300个代码库。优选的是,源码服务器10根据开库规则,对操作系统级的代码包进行库深度及库广度扫描,建立起包括多个代码库的代码库结构,生成索引文件,并将代码包中文件批量创建并上传到代码库中。所述开库规则根据源码服务器10和/或合作服务器对代码库结构的指定需求制定,从而满足各个公司对于操作系统代码库结构的指定需求。

[0121] 步骤S302,源码服务器10将代码库迁移至开放服务器20。

[0122] 步骤S303,合作方服务器30在开放服务器20上进行代码共建。

[0123] 这里,合作方服务器30直接在开放服务器20上将代码修改,源码服务器10可以直接获取修改后的代码。优选的是,开放服务器20为合作方服务器30提供指定端口的访问权限。更好的是,开放服务器20创建授权账号,授权账号对应开放服务器20的指定端口的访问权限。合作方服务器30通过授权账号访问开放服务器20的指定端口进行代码共建。

[0124] 图4是本申请优选代码共建方法的流程图,代码共建方法应用如图2所示的代码共建系统100中,所述代码共建方法包括步骤有:

[0125] 步骤S401,源码服务器10根据操作系统级的代码和预定的开库规则,定制化建立多个代码库。

[0126] 优选的是,源码服务器10根据开库规则,对操作系统级的代码包进行库深度及库广度扫描,建立起包括多个代码库的代码库结构,生成索引文件,并将代码包中文件批量创建并上传到代码库中。所述开库规则根据源码服务器10和/或合作服务器对代码库结构的指定需求制定,从而满足各个公司对于操作系统代码库结构的指定需求。

[0127] 步骤S402,扫描代码及日志获取每个代码库的数据量、变更频率和读取频率。

[0128] 步骤S403,根据每个代码库的数据量、变更频率、读取频率和预定的代码库拆分值算法,计算出代码库的拆分值。

[0129] 优选的是,所述代码库拆分值算法为:

[0130] 代码库的拆分值 $G = p_c * (S * VC) + p_d * (S * VD)$,其中 p_c 和 p_d 分别为加权值, S 为当前代码库的数据量, VC 为当前代码库的数据变更频率, VD 为当前代码库的数据读取频率。

[0131] 步骤S404,根据预定的需求库总数,以及保证每个代码库的拆分值相近的拆分规则,对代码库进行拆库。

[0132] 步骤S405,将操作系统级的代码进行适配、二进制化集成并测试。

[0133] 步骤S406,若测试通过后,将代码进行安全检查。

[0134] 即扫描代码库中是否存在不想开放的源码文件,若存在不想开放的源码文件则禁止开放。

[0135] 步骤S407,若代码安全检查通过后,将代码库迁移至开放服务器20。

[0136] 步骤S408,开放服务器20对代码库中的代码进行代码安全检查。

[0137] 即扫描代码库中是否存在不想开放的源码文件,若存在不想开放的源码文件则禁止开放。

[0138] 步骤S409,若代码安全检查通过后,将开放服务器20中的代码库进行开放。

[0139] 步骤S410,合作方服务器30在开放服务器20上进行代码共建。

[0140] 优选的是,开放服务器20为合作方服务器30提供指定端口的访问权限。更好的是,开放服务器20创建授权账号,授权账号对应开放服务器20的指定端口的访问权限。合作方服务器30通过授权账号访问开放服务器20的指定端口进行代码共建。

[0141] 综上所述,本申请提供操作系统级的代码共建技术方案,由源码服务器根据操作系统级的代码和预定的开库规则,定制化建立多个代码库,并将所述代码库迁移至开放服务器,合作方服务器在开放服务器上进行代码共建。借此,本申请通过定制化建立代码库,能够满足多个公司对于操作系统代码库结构的指定需求,可增强移植性及完善共建效果,提高效率并减小成本;并且,由于本申请采用服务器点对点单向授权共建方法,大幅度减少

代码中途泄露的可能性,提高了代码共建的安全性;另外,由于本申请中多个公司之间在同一台开放服务器上进行研发,其实现效果与在同公司研发效果类似,因此交互更加高效。

[0142] 当然,本申请还可有其它多种实施例,在不背离本申请精神及其实质的情况下,熟悉本领域的技术人员当可根据本申请作出各种相应的改变和变形,但这些相应的改变和变形都应属于本申请所附的权利要求的保护范围。

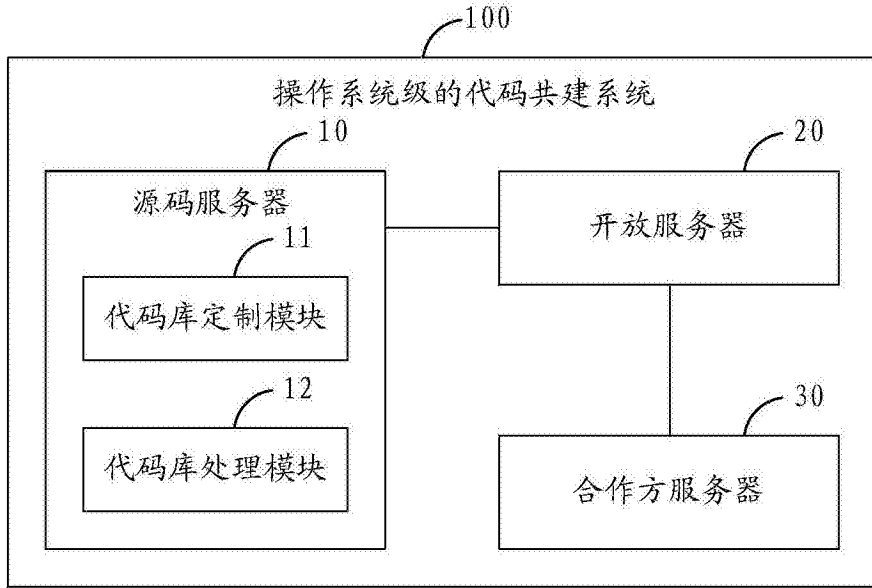


图1

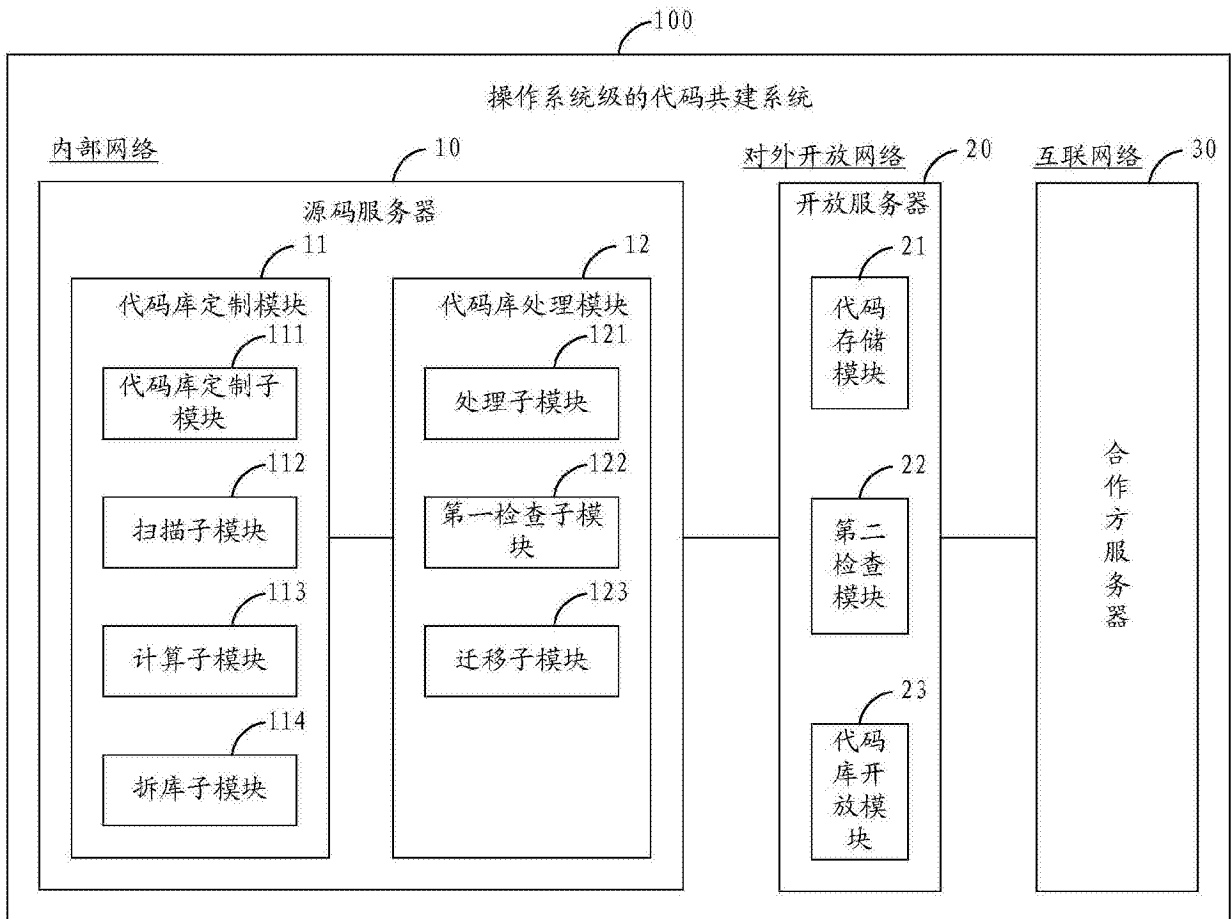


图2

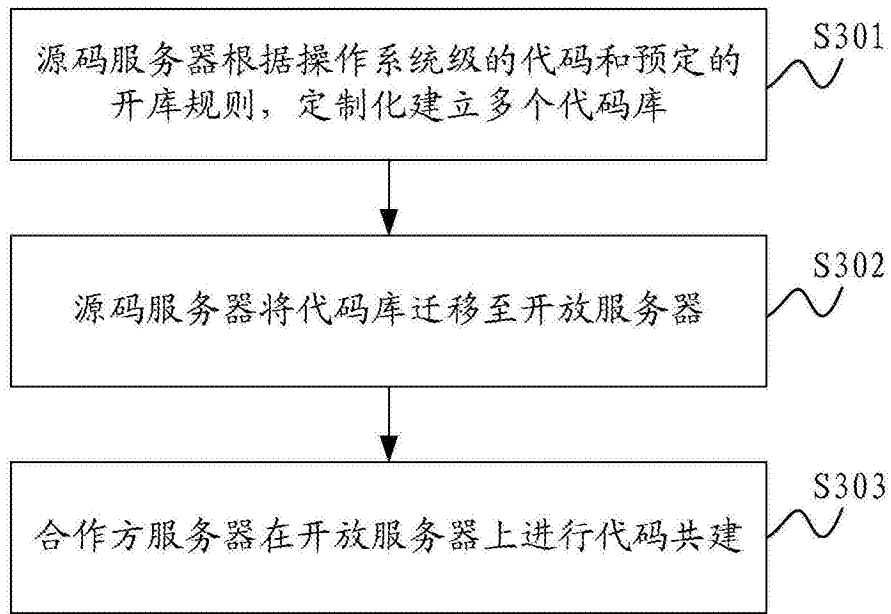


图3

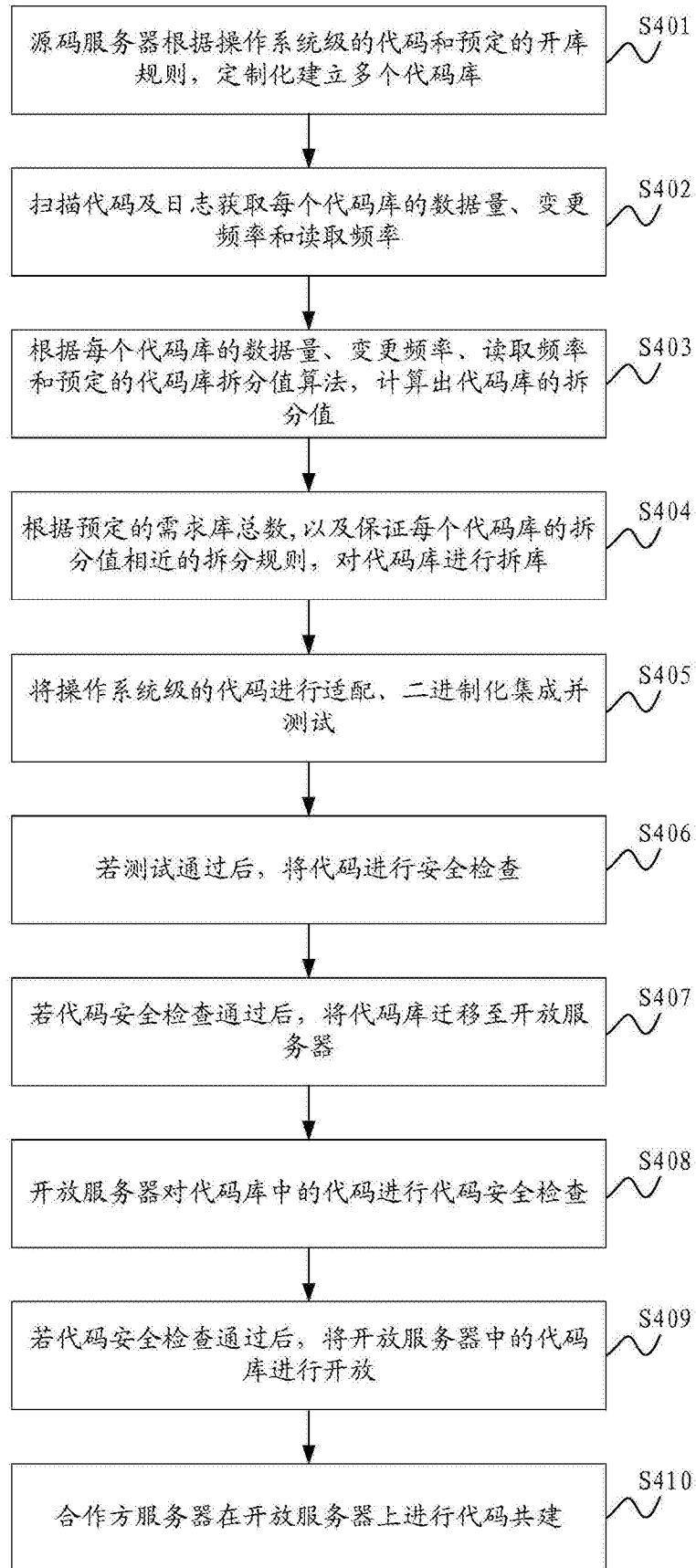


图4