

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2019 年 8 月 1 日 (01.08.2019)



(10) 国际公布号
WO 2019/144809 A1

(51) 国际专利分类号:
H04L 29/08 (2006.01)

(21) 国际申请号: PCT/CN2019/071136

(22) 国际申请日: 2019 年 1 月 10 日 (10.01.2019)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201810065040.2 2018 年 1 月 23 日 (23.01.2018) CN

(71) 申请人: 阿里巴巴集团控股有限公司 (**ALIBABA GROUP HOLDING LIMITED**) [—/CN]; 开曼群岛
大开曼资本大厦一座四层 847 号邮箱,
Grand Cayman (KY)。

(72) 发明人: 张杨 (**ZHANG, Yang**); 中国浙江省杭州市
市余杭区文一西路 969 号 3 号楼 5 楼 阿里巴巴
集团法务部, Zhejiang 311121 (CN)。

(74) 代理人: 北京博思佳知识产权代理有限公司 (**BEIJING BESTIPR INTELLECTUAL PROPERTY LAW CORPORATION**); 中国北京市
海淀区上地三街 9 号嘉华大厦 B 座 409,
Beijing 100085 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家
保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG,
BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU,
CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB,
GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS,
JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK,
LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX,
MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL,

(54) **Title:** SERVICE UPDATE METHOD AND DEVICE, SYSTEM

(54) 发明名称: 一种服务更新方法及装置、系统

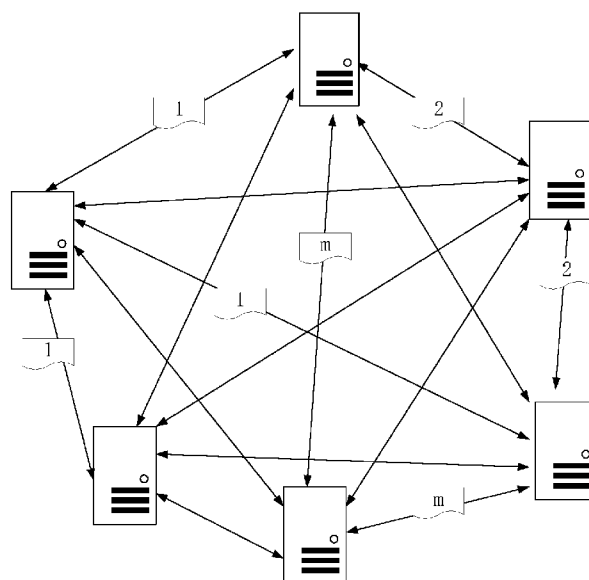


图 4c

(57) **Abstract:** The embodiments of the present application provide a service update method and device, and a system. The method comprises: in a plurality of service nodes on which service updating is to be performed, acquiring a plurality of pieces of sub-update data from the plurality of service nodes other than a first service node; combining the plurality of pieces of sub-update data acquired, to obtain service update data; and performing service updating on the first service node according to the combined service update data. On the basis of pre-dividing service update data for performing service updating on service nodes into plurality of pieces of sub-update data, the plurality of service nodes on which service updating is to be performed commonly participating in diffusion of each sub-update data, and the data used by any service node for service update being obtained by collecting all the sub-update data



WO 2019/144809 A1

PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,
SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, ZA, ZM, ZW。

- (84) 指定国(除另有指明, 要求每一种可提供的地区
保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ,
NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM,
AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,
CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,
IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

- 包括国际检索报告(条约第21条(3))。

from other associated service nodes and then combining same, the present invention effectively uses computing resources and network resources of the plurality of service nodes, improves the service update efficiency of the multi-service nodes, and implements service updates on each service node quickly, efficiently and securely.

(57) 摘要: 本申请实施例提供了一种服务更新方法及装置、系统, 该方法包括: 在待服务更新的多个服务节点中, 从第一服务节点以外的多个服务节点中获取多个子更新数据; 将获取到的多个子更新数据进行合并, 得到服务更新数据; 根据合并得到的服务更新数据, 对第一服务节点进行服务更新。基于预先将用于服务节点进行服务更新的服务更新数据分割为多个子更新数据, 多个待服务更新的服务节点共用参与各子更新数据的扩散, 任一服务节点用于服务更新的数据是从其他关联服务节点中收集所有子更新数据后再合并得到的, 有效利用多个服务节点的计算资源和网路资源, 提高了多服务节点的服务更新效率, 实现快速、高效、安全地对各服务节点进行服务更新。

一种服务更新方法及装置、系统

技术领域

[01] 本申请涉及服务更新技术领域，尤其涉及一种服务更新方法及装置、系统。

背景技术

5 [02] 目前，随着互联网、云计算产业的快速发展，互联网或云计算厂商需要频繁部署、更新产品功能和服务。在大型互联网产品中，对客户端提供的功能和服务往往由成百上千台服务节点上的各类软件参与实现的。产品功能和服务的每次部署均需要更新所有服务节点上的相关软件或配置。

10 [03] 当前，相关技术中提供了一种服务更新方法，主要为：首先，将更新服务时所需的服务更新数据上传至指定客户机，即用于更新各服务节点上的相关软件或配置的设备；然后，该指定客户机通过网络依次将服务更新数据传输至多个待服务更新的服务节点，服务节点接收到服务更新数据后进行本地服务更新，在整个服务更新的过程中，只有指定客户机参与服务更新数据的分发，各服务节点均等待指定客户机将服务更新数据传输至本地。另外，在整个服务更新的过程中，一旦指定客户机出现单点故障，将导致服务
15 更新数据无法正常发布而暂停，进而导致出现线上多个服务节点的服务版本不一致或功能异常的情况。

[04] 由此可知，采用现有技术中的服务更新方法，对多个服务节点进行服务更新的过程中，存在对指定客户机依赖性高、服务更新耗费时间长、更新效率低的问题。

发明内容

20 [05] 本申请实施例的目的是提供一种服务更新方法及装置、系统，基于预先将用于服务节点进行服务更新的服务更新数据分割为多个子更新数据，多个待服务更新的服务节点共用参与各子更新数据的扩散，任一服务节点用于服务更新的数据是从其他关联服务节点中收集所有子更新数据后再合并得到的，有效利用多个服务节点的计算资源和网路资源，提高了多服务节点的服务更新效率，实现快速、高效、安全地对各服务节点进行服
25 务更新。

[06] 为解决上述技术问题，本申请实施例是这样实现的：

[07] 本申请实施例提供了一种服务更新方法，包括：

[08] 在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；

[09] 将获取到的所述多个子更新数据进行合并，得到所述服务更新数据；

5 [10] 根据合并得到的所述服务更新数据，对所述第一服务节点进行服务更新。

[11] 本申请实施例提供了一种服务更新装置，包括：

[12] 子数据获取模块，用于在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；

10 [13] 更新数据生成模块，用于将获取到的所述多个子更新数据进行合并，得到所述服务更新数据；

[14] 服务更新模块，用于根据合并得到的所述服务更新数据，对所述第一服务节点进行服务更新。

[15] 本申请实施例提供了一种服务更新系统，包括：待服务更新的多个服务节点，两两

15 服务节点之间通信连接，每个服务节点设置有上述服务更新装置。

[16] 本申请实施例提供了一种服务更新设备，包括：

[17] 处理器；以及

[18] 被安排成存储计算机可执行指令的存储器，所述可执行指令在被执行时使所述处理器：

20 [19] 在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；

[20] 将获取到的所述多个子更新数据进行合并，得到所述服务更新数据；

[21] 根据合并得到的所述服务更新数据，对所述第一服务节点进行服务更新。

[22] 本申请实施例提供了一种存储介质，用于存储计算机可执行指令，所述可执行指令在被执行时实现以下流程：

25

[23] 在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个

子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；

[24]将获取到的所述多个子更新数据进行合并，得到所述服务更新数据；

[25]根据合并得到的所述服务更新数据，对所述第一服务节点进行服务更新。

[26]本申请实施例中的服务更新方法及装置、系统，在待服务更新的多个服务节点中，

- 5 从第一服务节点以外的多个服务节点中获取多个子更新数据；将获取到的多个子更新数据进行合并，得到服务更新数据；根据合并得到的服务更新数据，对第一服务节点进行服务更新。基于预先将用于服务节点进行服务更新的服务更新数据分割为多个子更新数据，多个待服务更新的服务节点共用参与各子更新数据的扩散，任一服务节点用于服务更新的数据是从其他关联服务节点中收集所有子更新数据后再合并得到的，有效利用多
10 个服务节点的计算资源和网路资源，提高了多服务节点的服务更新效率，实现快速、高效、安全地对各服务节点进行服务更新。

附图说明

[27]为了更清楚地说明本申请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
15 申请中记载的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

[28]图 1 为本申请实施例提供的服务更新的应用场景示意图；

[29]图 2 为本申请实施例提供的服务更新方法的第一种流程示意图；

[30]图 3 为本申请实施例提供的服务更新方法的第二种流程示意图；

- 20 [31]图 4a 为本申请实施例提供的服务更新方法中子更新数据扩散的实现原理示意图之一；

[32]图 4b 为本申请实施例提供的服务更新方法中子更新数据扩散的实现原理示意图之二；

- 25 [33]图 4c 为本申请实施例提供的服务更新方法中子更新数据扩散的实现原理示意图之三；

[34]图 4d 为本申请实施例提供的服务更新方法中子更新数据扩散的实现原理示意图之四；

[35]图 5 为本申请实施例提供的服务更新方法的第三种流程示意图;

[36]图 6 为本申请实施例提供的服务更新装置的第一种模块组成示意图;

[37]图 7 为本申请实施例提供的服务更新装置的第二种模块组成示意图;

[38]图 8 为本申请实施例提供的服务更新系统的结构示意图;

5 [39]图 9 为本申请实施例提供的服务更新设备的结构示意图。

具体实施方式

[40]为了使本技术领域的人员更好地理解本申请中的技术方案,下面将结合本申请实施例中的附图,对本申请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申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,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,都应当属于本申请保护的范围。

[41]本申请实施例提供了一种服务更新方法及装置、系统,基于预先将用于服务节点进行服务更新的服务更新数据分割为多个子更新数据,多个待服务更新的服务节点共用参与各子更新数据的扩散,任一服务节点用于服务更新的数据是从其他关联服务节点中收集所有子更新数据后再合并得到的,有效利用多个服务节点的计算资源和网路资源,提高了多服务节点的服务更新效率,实现快速、高效、安全地对各服务节点进行服务更新。

[42]图 1 为本申请实施例提供的服务更新的应用场景示意图,如图 1 所示,在图 1 中涉及多个待服务更新的服务节点,该服务节点为互联网服务或云计算服务的托管者,可以是物理机、虚拟容器、或服务器软件等,在子更新数据初次分发阶段,即首次将由服务更新数据分割得到的多个子更新数据分发给多个服务节点的过程,每个子更新数据被分发到至少一个服务节点中;随之进入子更新数据的扩散阶段,随着子更新数据的不断扩散,每个子更新数据所在的服务节点的数量越来越多,每个服务节点存储的子更新数据的个数也越来越多,任一服务节点收集所有子更新数据后,合并得到完整的服务更新数据,并利用该服务更新数据完成本地服务更新。其中,在需要对产品功能和服务进行更新部署时,先更新待服务更新的所有服务节点上的相关软件或配置,在服务更新过程中,具有跨机房、分布式去中心化、容错机制、系统无关、硬件无关的特点,充分利用服务节点之间、机房之间的网路资源和计算资源,保证子更新数据快速、高效、安全、可靠地在多服务节点中扩散。最终,每个待服务更新的服务节点均合并得到一份完整的服务

更新数据，从而实现多服务节点的服务更新。

[43]图 2 为本申请实施例提供的服务更新方法的第一种流程示意图，图 2 中的方法能够由图 1 中的服务节点 100 执行，适用于互联网服务和云计算服务的服务更新部署，如图 2 所示，该方法至少包括以下步骤：

5 [44]S201，在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据，其中，该多个子更新数据为预先由服务更新数据分割后得到的，第一服务节点为待服务更新的多个服务节点中的任一服务节点，由于任一服务节点不仅参与子更新数据的扩散，且自身也需要进行服务更新，因此，在子更新数据由一个服务节点扩散到其他服务节点的过程中，传输的子更新数据是服务更新数据分割后得到的数据
10 子块的数据副本。其中，第一服务节点获取到的多个子更新数据可以来自第一服务节点以外的部分服务节点，也可以是来自第一服务节点以外的全部服务节点。

[45]S202，将获取到的多个子更新数据进行合并，得到用于服务更新的服务更新数据，具体的，获取到所有的子更新数据后，将多个子更新数据还原成分割前的服务更新数据。

[46]S203，根据合并得到的服务更新数据，对第一服务节点进行服务更新，具体的，在
15 由多个子更新数据合并得到完整的服务更新数据后，如果本次服务更新为原应用服务的升级，则利用该服务更新数据替代与待升级服务相关的原服务数据，如果本次服务更新为新增应用服务，则将该服务更新数据添加至原服务数据，再重启本服务节点服务。

[47]例如，针对支付宝应用中新增的“发红包赚赏金”功能，需要将实现“发红包赚赏金”应用服务的服务更新数据分割为多个子更新数据，再采用步骤 S201 至 S203 使得各
20 提供该应用服务的服务节点获取到所有子更新数据并合并为服务更新数据，各服务节点利用该服务更新数据更新本地应用服务，进而对客户端提供相应的应用服务。

[48]本申请实施例中，基于预先将用于服务节点进行服务更新的服务更新数据分割为多个子更新数据，多个待服务更新的服务节点共用参与各子更新数据的扩散，任一服务节点用于服务更新的数据是从其他关联服务节点中收集所有子更新数据后再合并得到的，
25 有效利用多个服务节点的计算资源和网路资源，提高了多服务节点的服务更新效率，实现快速、高效、安全地对各服务节点进行服务更新。

[49]其中，如图 3 所示，在 S201 在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据之前，还包括：

[50]S204, 接收服务更新请求, 其中, 该服务更新请求携带有由服务更新数据分割得到的所有子更新数据的数据标识; 具体的, 每个子更新数据对应于一个数据标识, 例如, 可以将数据序号作为子更新数据的数据标识, 以服务更新数据被分割成 m 个子更新数据为例, 首个子更新数据的数据标识为 1, 最后一个子更新数据的数据标识为 m 。

5 [51]对应的, 上述 S201 在待服务更新的多个服务节点中, 从第一服务节点以外的多个服务节点中获取多个子更新数据, 具体包括:

[52]S2011, 在待服务更新的多个服务节点中, 根据接收到的服务更新请求, 从第一服务节点以外的多个服务节点中分别获取与各数据标识对应的子更新数据。

10 [53]在具体实施时, 如图 4a 所示, 在待服务更新的 N 个服务节点中, 将服务更新数据上传至任一服务节点 (如, 服务节点 k), 该服务节点接收到服务更新数据后, 将服务更新数据分割为多个子更新数据 (如, 分割为 m 个子更新数据); 在待服务更新的多个服务节点中, 如图 4b 所示, 选取目标数量的服务节点作为首发服务节点 (如, 首发服务节点的数量为 x , $x \leq N$), 将多个子更新数据分发给多个首发服务节点 (如, 将 m 个子更新数据分发给 x 个首发服务节点), 此时, 针对每个首发服务节点, 该首发服务节点至少接收到一个子更新数据, 针对每个子更新数据, 该子更新数据至少在一个首发服务节点上存在, 此过程为子更新数据初次分发阶段, 其中, 如果第一服务节点属于首发服务节点, 则接收服务节点 k 的服务更新请求; 如果第一服务节点不属于首发服务节点, 则接收服务节点 k 或任一首发服务节点的服务更新请求。随之进入子更新数据的扩散阶段, 如图 4c 所示, 各子更新数据在多个服务节点之间进行扩散, 最终, 每个服务节点获取到所有子更新数据, 如图 4d 所示, 将获取到的多个子更新数据进行合并, 得到用于服务更新的服务更新数据。

[54]具体的, 针对第一服务节点属于首发服务节点的情况, 上述 S204 接收服务更新请求, 具体包括:

25 [55]接收初次分发的至少一个子更新数据和更新数据分割信息, 其中, 该更新数据分割信息包括: 由服务更新数据分割得到的所有子更新数据的数据标识, 且每个子更新数据均被分发到至少一个服务节点。

[56]其中, 如果第一服务节点参与子更新数据初次分发阶段, 第一服务节点接收到的服务更新请求中可以同时携带至少一个子更新数据和更新数据分割信息, 第一服务节点接收到初次分发的至少一个子更新数据后, 参与后续的子更新数据的扩散阶段, 将本地已

存储的子更新数据传输至需要该子更新数据的服务节点,以及从其他服务节点中获取本地不存在的子更新数据。

[57]针对第一服务节点不属于首发服务节点的情况,第一服务节点直接进入子更新数据的扩散阶段,第一服务节点接收到的服务更新请求中可以只携带更新数据分割信息,根据该更新数据分割信息向其他服务节点获取本地不存在的子更新数据,同时获取到子更新数据后,如果接收到子更新数据获取请求,则将本地已存储的子更新数据传输至需要该子更新数据的服务节点。

[58]进一步的,考虑到初次分发时可能存在数据丢包、或者服务节点未将接收到的子更新数据扩散到其他服务节点时出现异常,导致子更新数据无法正常扩散到所有服务节点,为了确保各待服务更新的服务节点均能够获取到所有子更新数据,基于此,由服务更新数据分割得到的每个子更新数据均被分发到至少两个服务节点,即将同一子更新数据分发给两个首发服务节点。

[59]具体的,在子更新数据初次分发阶段,即将多个子更新数据分发给多个首发服务节点的过程中,将同一子更新数据分发给至少两个首发服务节点,此时,在待服务更新的多个服务节点中,每个子更新数据存储于至少两个服务节点中,这样即使某一服务节点出现异常,仍可以从另一服务节点中获取子更新数据,保证了各子更新数据均能够正常扩散到所有服务节点,进而确保各待服务更新的服务节点均能够获取到所有子更新数据,提高了服务更新数据发布成功率。

[60]其中,如图5所示,上述S2011在待服务更新的多个服务节点中,根据接收到的服务更新请求,从第一服务节点以外的多个服务节点中分别获取与各数据标识对应的子更新数据,具体包括:

[61]S20111,分别获取第一服务节点以外的多个服务节点的子更新数据存储信息,其中,该子更新数据存储信息包括:服务节点上已存储的子更新数据的数据标识。

[62]S20112,针对服务更新请求中每个数据标识,判断第一服务节点是否存在与该数据标识对应的子更新数据;具体的,按照预设顺序,逐一查询本地是否存在各数据标识对应的子更新数据,如果不存在,则向已存储有该子更新数据的服务节点获取。

[63]若否,则执行S20113,根据各子更新数据存储信息和本地不存在的子更新数据对应的目标数据标识,选取一个服务节点作为目标服务节点,以及从该目标服务节点中获取目标数据标识对应的子更新数据。

[64]其中，第一服务节点获取到除自身以外的各服务节点的子更新数据存储信息后，可以确定各子更新数据分布在哪些服务节点中，进而有针对性地从目标服务节点中获取本地不存在的子更新数据，直到本地存储有所有数据标识对应的子更新数据，具体的，第一服务接节点可以同时从至少两个不同的服务节点中获取不同的数据标识对应的子更新数据，并且，所有服务节点能够同时获取各自所需的子更新数据，将数据处理负载和数据传输负载分配到所有服务节点上，大大节省了服务更新数据获取时间，提高了多服务节点的服务更新效率和成功率。

[65]进一步的，考虑到在确定目标服务节点的过程中，如果待服务更新的服务节点的数量比较多，那么获取到的子更新数据存储信息的数量也较多，为了提高目标服务节点确定效率，基于此，上述 S20113 根据各子更新数据存储信息和本地不存在的子更新数据对应的目标数据标识，选取一个服务节点作为目标服务节点，具体包括：

[66]根据各子更新数据存储信息，确定子更新数据分布信息，其中，该子更新数据分布信息包括：服务更新请求中各数据标识对应的子更新数据所在的服务节点的信息。

[67]根据子更新数据分布信息和本地不存在的子更新数据对应的目标数据标识，选取一个服务节点作为目标服务节点。

[68]具体的，在接收到除自身以外的多个服务节点的子更新数据存储信息后，先对接收到的多个子更新数据存储信息进行汇总，得到子更新数据分布信息，在子更新数据分布信息中，直接查询到各数据标识与存储该数据标识对应的子更新数据的服务节点的信息之间的对应关系，例如，数据标识 1—服务节点 A\B\C，即服务节点 A、服务节点 B、服务节点 C 中存储有数据标识 1 对应的子更新数据，如果需要获取数据标识 1 对应的子更新数据，此时可以从服务节点 A、服务节点 B、服务节点 C 中选择一个作为目标服务节点，这样一目了然得知目标数据标识对应的子更新数据分布在哪些服务节点中，快速定位到目标服务节点。

[69]进一步的，为了更加充分利用各服务节点的计算资源和网路资源，避免出现某个服务节点负载率很高，而其他服务节点负载率很低的情况，达到多服务节点的负载均衡的效果，基于此，上述根据子更新数据分布信息和本地不存在的子更新数据对应的目标数据标识，选取一个服务节点作为目标服务节点，具体包括：

[70]基于子更新数据分布信息，将已存储目标数据标识对应的子更新数据的多个服务节点作为备选服务节点，其中，该目标数据标识为本地不存在的子更新数据对应的数据标

识。

[71]根据各备选服务节点的负载率,将负载率最低的备选服务节点确定为目标服务节点。

[72]具体的,在选取目标服务节点的过程中,参考备选服务节点的负载情况,在多个服务节点都存在目标数据标识对应的子更新数据的情况下,将负载率最低的备选服务节点
5 确定为目标服务节点,这样能够实现更加充分利用各服务节点的计算资源和网路资源,达到多服务节点的负载均衡的效果,进而更进一步地提高多服务节点的服务更新效率。

[73]具体的,在从目标服务节点中获取目标数据标识对应的子更新数据之后,还包括:

[74]若子更新数据获取失败,则按照负载率由低至高的顺序重新选择下一个目标服务节点,并从重新选择的目标服务节点中获取目标数据标识对应的子更新数据。

10 [75]其中,针对只有一个服务节点存储有目标数据标识对应的子更新数据的情况,需要等待该目标服务节点恢复后,继续从该目标服务节点中获取所需的子更新数据。为了避免该情况的发生,在具体实施时,在子更新数据初次分发阶段,将同一子更新数据分发给至少两个首发服务节点,可以保证至少两个服务节点存储有目标数据标识对应的子更新数据;另外,即使在子更新数据初次分发阶段,将同一子更新数据只分发给一个首发
15 服务节点,而随着子更新数据的不断扩散,也能够保证至少两个服务节点存储有目标数据标识对应的子更新数据,提供了数据获取失败冗余机制,提高了服务更新数据发布成功率。

[76]针对至少两个服务节点存储有目标数据标识对应的子更新数据的情况,第一服务节点从某一目标服务节点获取子更新数据失败后,则变更服务节点重新获取子更新数据,
20 具体的,在服务更新过程中,若子更新数据获取失败,则说明目标服务节点出现异常,此时,第一服务节点重新选取下一个目标服务节点。

[77]另外,当出现异常的目标服务节点恢复正常后,该目标服务节点继续执行:分别获取本服务节点以外的多个服务节点的子更新数据存储信息;针对服务更新请求中每个数据标识,判断本地是否存在与该数据标识对应的子更新数据;若否,则根据最新版本的
25 各子更新数据存储信息和该数据标识,选取一个服务节点作为目标服务节点,以及从该目标服务节点中获取数据标识对应的子更新数据,直到获取到所有子更新数据。

[78]其中,针对获取子更新数据存储信息的过程可以通过如下两种方式获得,具体为:

[79]第一种方式:每个服务节点的子更新数据存在更新时,主动向其他服务节点广播更

新后的子更新数据存储信息，具体的，上述 S20111 分别获取第一服务节点以外的多个服务节点的子更新数据存储信息，具体包括：

[80]接收第一服务节点以外的多个服务节点广播的子更新数据存储信息，其中，该子更新数据存储信息为第一服务节点以外的多个服务节点在获取到新的子更新数据且更新本地的子更新数据存储信息后得到的。

[81]第二种方式：每个服务节点接收到其他服务节点的子数据存储信息获取请求后，才向相应的服务节点发送最新版本的子更新数据存储信息，具体的，上述 S20111 分别获取第一服务节点以外的多个服务节点的子更新数据存储信息，具体包括：

[82]向第一服务节点以外的各服务节点发送子数据存储信息获取请求；

10 [83]接收第一服务节点以外的各服务节点发送的本地最新版本的子更新数据存储信息，其中，该子更新数据存储信息为第一服务节点以外的各服务节点响应于子数据存储信息获取请求发送的。

[84]其中，针对合并子更新数据的过程，具体的，上述 S202 将获取到的多个子更新数据进行合并，得到用于服务更新的服务更新数据，具体包括：

15 [85]按照数据标识由小到大的顺序，对获取到的多个子更新数据进行排序；

[86]根据排序结果，将各子更新数据组合得到用于服务更新的服务更新数据。

[87]其中，针对每个服务节点而言，该服务节点不仅向其他服务节点获取子更新数据，还需要基于其他服务节点的子更新数据获取请求，向相应的服务节点发送子更新数据，基于此，上述方法还包括：

20 [88]接收第一服务节点以外的各服务节点发送的子更新数据获取请求；

[89]将与接收到的获取请求对应的子更新数据发送至相应的服务节点，以使接收到该子更新数据的服务节点判断本地是否存储有所有子更新数据，如果是，则进行本地服务更新，即第一服务节点也参与子更新数据的扩散。

25 [90]其中，针对每个服务节点而言，该服务节点不仅向其他服务节点获取子更新数据存储信息，还需要向其他服务节点发送本地最新版本的子更新数据存储信息，基于此，上述方法还包括：

[91]将本地最新版本的子更新数据存储信息发送至相应的服务节点，以使接收到该子更

新数据存储信息的服务节点确定目标服务节点，并从该目标服务节点获取子更新数据。

[92]具体的，针对发送子更新数据存储信息的过程可以通过如下两种方式实现，具体为：

[93]第一种方式：第一服务节点的子更新数据存在更新时，主动向其他服务节点广播更新后的子更新数据存储信息，具体的，上述将本地最新版本的子更新数据存储信息发送至相应的服务节点，具体包括：

[94]在获取到新的子更新数据后，更新本地的子更新数据存储信息；

[95]将更新后的子更新数据存储信息广播至第一服务节点以外的多个服务节点。

[96]第二种方式：第一服务节点接收到其他服务节点的子数据存储信息获取请求后，才向该服务节点发送最新版本的子更新数据存储信息，具体的，上述将本地最新版本的子更新数据存储信息发送至相应的服务节点，具体包括：

[97]接收第一服务节点以外的任一服务节点发送子数据存储信息获取请求；

[98]将本地最新版本的子更新数据存储信息传输至发送子数据存储信息获取请求的服务节点。

[99]本申请实施例中的服务更新方法，在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据；将获取到的多个子更新数据进行合并，得到服务更新数据；根据合并得到的服务更新数据，对第一服务节点进行服务更新。基于预先将用于服务节点进行服务更新的服务更新数据分割为多个子更新数据，多个待服务更新的服务节点共用参与各子更新数据的扩散，任一服务节点用于服务更新的数据是从其他关联服务节点中收集所有子更新数据后再合并得到的，有效利用多个服务节点的计算资源和网路资源，提高了多服务节点的服务更新效率，实现快速、高效、安全地对各服务节点进行服务更新。

[100] 对应上述图 2 至图 5 描述的服务更新方法，基于相同的技术构思，本申请实施例还提供了一种服务更新装置，图 6 为本申请实施例提供的服务更新装置的第一种模块组成示意图，该装置用于执行图 2 至图 5 描述的服务更新方法，如图 6 所示，该装置包括：

[101] 子数据获取模块 601，用于在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；

[102] 更新数据生成模块 602, 用于将获取到的所述多个子更新数据进行合并, 得到所述服务更新数据;

[103] 服务更新模块 603, 用于根据合并得到的所述服务更新数据, 对所述第一服务节点进行服务更新。

5 [104] 可选地, 如图 7 所示, 上述装置还包括:

[105] 更新请求接收模块 604, 用于在从第一服务节点以外的多个服务节点中获取多个子更新数据之前, 接收服务更新请求, 其中, 所述服务更新请求携带有由服务更新数据分割得到的所有子更新数据的数据标识;

[106] 所述子数据获取模块 601, 具体用于:

10 [107] 根据所述服务更新请求, 从第一服务节点以外的多个服务节点中分别获取与各所述数据标识对应的子更新数据。

[108] 可选地, 所述更新请求接收模块 604, 具体用于:

[109] 接收初次分发的至少一个子更新数据和更新数据分割信息, 其中, 所述更新数据分割信息包括: 由服务更新数据分割得到的所有子更新数据的数据标识, 且每个子更新数据均被分发到至少一个服务节点。

15 [110] 可选地, 由服务更新数据分割得到的每个子更新数据均被分发到至少两个服务节点。

[111] 可选地, 所述子数据获取模块 601, 进一步具体用于:

[112] 分别获取第一服务节点以外的多个服务节点的子更新数据存储信息, 其中, 所述子更新数据存储信息包括: 服务节点上已存储的子更新数据的数据标识;

[113] 针对所述服务更新请求中每个数据标识, 判断所述第一服务节点是否存在与该数据标识对应的子更新数据;

[114] 若否, 则根据各所述子更新数据存储信息和所述数据标识, 选取一个服务节点作为目标服务节点, 以及从所述目标服务节点中获取所述数据标识对应的子更新数据。

25 [115] 可选地, 所述子数据获取模块 601, 更进一步具体用于:

[116] 根据各所述子更新数据存储信息, 确定子更新数据分布信息, 其中, 所述子更新数据分布信息包括: 所述服务更新请求中各数据标识对应的子更新数据所在的服务节

点的信息;

[117] 根据所述子更新数据分布信息和所述数据标识, 选取一个服务节点作为目标服务节点。

[118] 可选地, 所述子数据获取模块 601, 再进一步具体用于:

5 [119] 基于所述子更新数据分布信息, 将已存储所述数据标识对应的子更新数据的多个服务节点作为备选服务节点;

[120] 根据各所述备选服务节点的负载率, 将负载率最低的备选服务节点确定为目标服务节点。

[121] 可选地, 上述子数据获取模块 601, 还具体用于:

10 [122] 在从所述目标服务节点中获取所述数据标识对应的子更新数据之后, 若子更新数据获取失败, 则按照负载率由低至高的顺序重新选择下一个目标服务节点, 并从重新选择的所述目标服务节点中获取所述数据标识对应的子更新数据。

[123] 可选地, 所述子数据获取模块 601, 更进一步具体用于:

15 [124] 接收第一服务节点以外的多个服务节点广播的子更新数据存储信息, 其中, 所述子更新数据存储信息为第一服务节点以外的多个服务节点在获取到新的子更新数据且更新本地的子更新数据存储信息后得到的。

[125] 可选地, 所述子数据获取模块 601, 更进一步具体用于:

[126] 向第一服务节点以外的各服务节点发送子数据存储信息获取请求;

20 [127] 接收第一服务节点以外的各服务节点发送的本地最新版本的子更新数据存储信息, 其中, 所述子更新数据存储信息为第一服务节点以外的各服务节点响应于所述子数据存储信息获取请求发送的。

[128] 可选地, 所述更新数据生成模块 602, 具体用于:

[129] 按照数据标识由小到大的顺序, 对所述多个子更新数据进行排序;

[130] 根据排序结果, 将各所述子更新数据组合得到所述服务更新数据。

25 [131] 可选地, 上述装置还包括:

[132] 子更新数据发送模块 605, 用于接收第一服务节点以外的各服务节点发送的子更

新数据获取请求；将与所述获取请求对应的子更新数据发送至相应的服务节点。

[133] 可选地，上述装置还包括：

[134] 存储信息发送模块 606，用于将本地最新版本的子更新数据存储信息发送至相应的服务节点。

5 [135] 本申请实施例中的服务更新装置，在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据；将获取到的多个子更新数据进行合并，得到服务更新数据；根据合并得到的服务更新数据，对第一服务节点进行服务更新。基于预先将用于服务节点进行服务更新的服务更新数据分割为多个子更新数据，多个待
10 服务更新的服务节点共用参与各子更新数据的扩散，任一服务节点用于服务更新的数据是从其他关联服务节点中收集所有子更新数据后再合并得到的，有效利用多个服务节点的
计算资源和网路资源，提高了多服务节点的服务更新效率，实现快速、高效、安全地对各服务节点进行服务更新。

[136] 本发明实施例提供的服务更新装置能够实现上述服务更新方法对应的实施例中的各个过程，为避免重复，这里不再赘述。

15 [137] 进一步地，对应上述图 2 至图 5 所示的方法，基于相同的技术构思，本申请实施例还提供了一种服务更新系统，如图 8 所示，该系统包括：待服务更新的多个服务节点 100，两两服务节点 100 之间通信连接，每个服务节点 100 设置有上述服务更新装置；其中，由服务更新数据分割得到的多个子更新数据（即数据子块 1 至 m）在服务节点 100
之间扩散，以使各服务节点 100 均获取到所有子更新数据。

20 [138] 具体的，任一待服务更新的第一服务节点，用于在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；将获取到的所述多个子更新数据进行合并，得到所述服务更新数据；根据合并得到的所述服务更新数据，对所述第一服务节点进行服务更新。

25 [139] 进一步的，所述第一服务节点，还用于：

[140] 接收服务更新请求，其中，所述服务更新请求携带有由服务更新数据分割得到的所有子更新数据的数据标识；

[141] 根据所述服务更新请求，分别获取与各所述数据标识对应的子更新数据。

[142] 进一步的，所述多个服务节点中任一服务节点（即上述服务节点k），用于：

[143] 接收服务更新数据和数据分割指示；

[144] 根据所述数据分割指示，将所述服务更新数据分割为多个子更新数据；

[145] 从多个待服务更新的服务节点中，选取目标数量的多个初始分配节点，其中，

5 该初始分配节点相当于上述首发服务节点；

[146] 向所述多个初始分配节点发送服务更新请求，其中，所述服务更新请求携带有初次分发的至少一个子更新数据和更新数据分割信息，每个子更新数据被分发给至少一个已选取的初始分配节点。

[147] 进一步的，每个子更新数据被分发给至少两个已选取的初始分配节点。

10 [148] 本申请实施例中的服务更新系统，针对任一服务节点，在待服务更新的多个服务节点中，从本服务节点以外的多个服务节点中获取多个子更新数据；将获取到的多个子更新数据进行合并，得到服务更新数据；根据合并得到的服务更新数据，对第一服务节点进行服务更新。基于预先将用于服务节点进行服务更新的服务更新数据分割为多个子更新数据，多个待服务更新的服务节点共用参与各子更新数据的扩散，任一服务节点
15 用于服务更新的数据是从其他关联服务节点中收集所有子更新数据后再合并得到的，有效利用多个服务节点的计算资源和网路资源，提高了多服务节点的服务更新效率，实现快速、高效、安全地对各服务节点进行服务更新。

[149] 本发明实施例提供的服务更新系统能够实现上述服务更新方法对应的实施例中的各个过程，为避免重复，这里不再赘述。

20 [150] 进一步地，对应上述图2至图5所示的方法，基于相同的技术构思，本申请实施例还提供了一种服务更新设备，该设备用于执行上述的服务更新方法，如图9所示。

[151] 服务更新设备可因配置或性能不同而产生比较大的差异，可以包括一个或一个以上的处理器901和存储器902，存储器902中可以存储有一个或一个以上存储应用程序或数据。其中，存储器902可以是短暂存储或持久存储。存储在存储器902的应用程序
25 序可以包括一个或一个以上模块（图示未示出），每个模块可以包括对服务更新设备中的一系列计算机可执行指令。更进一步地，处理器901可以设置为与存储器902通信，在服务更新设备上执行存储器902中的一系列计算机可执行指令。服务更新设备还可以包括一个或一个以上电源903，一个或一个以上有线或无线网络接口904，一个或一个

以上输入输出接口 905，一个或一个以上键盘 906 等。

[152] 在一个具体的实施例中，服务更新设备包括有存储器，以及一个或一个以上的程序，其中一个或者一个以上程序存储于存储器中，且一个或者一个以上程序可以包括一个或一个以上模块，且每个模块可以包括对服务更新设备中的一系列计算机可执行指令，且经配置以由一个或者一个以上处理器执行该一个或者一个以上程序包含用于进行以下计算机可执行指令：

[153] 在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；

[154] 将获取到的所述多个子更新数据进行合并，得到所述服务更新数据；

10 [155] 根据合并得到的所述服务更新数据，对所述第一服务节点进行服务更新。

[156] 可选地，计算机可执行指令在被执行时，还包含用于进行以下计算机可执行指令：在从第一服务节点以外的多个服务节点中获取多个子更新数据之前，还包括：

[157] 接收服务更新请求，其中，所述服务更新请求携带有由服务更新数据分割得到的所有子更新数据的数据标识；

15 [158] 所述从第一服务节点以外的多个服务节点中获取多个子更新数据，包括：

[159] 根据所述服务更新请求，从第一服务节点以外的多个服务节点中分别获取与各所述数据标识对应的子更新数据。

[160] 可选地，计算机可执行指令在被执行时，所述接收服务更新请求，包括：

20 [161] 接收初次分发的至少一个子更新数据和更新数据分割信息，其中，所述更新数据分割信息包括：由服务更新数据分割得到的所有子更新数据的数据标识，且每个子更新数据均被分发到至少一个服务节点。

[162] 可选地，计算机可执行指令在被执行时，由服务更新数据分割得到的每个子更新数据均被分发到至少两个服务节点。

25 [163] 可选地，计算机可执行指令在被执行时，所述根据所述服务更新请求，从第一服务节点以外的多个服务节点中分别获取与各所述数据标识对应的子更新数据，包括：

[164] 分别获取第一服务节点以外的多个服务节点的子更新数据存储信息，其中，所述子更新数据存储信息包括：服务节点上已存储的子更新数据的数据标识；

[165] 针对所述服务更新请求中每个数据标识,判断所述第一服务节点是否存在与该数据标识对应的子更新数据;

[166] 若否,则根据各所述子更新数据存储信息和所述数据标识,选取一个服务节点作为目标服务节点,以及从所述目标服务节点中获取所述数据标识对应的子更新数据。

5 [167] 可选地,计算机可执行指令在被执行时,所述根据各所述子更新数据存储信息和所述数据标识,选取一个服务节点作为目标服务节点,包括:

[168] 根据各所述子更新数据存储信息,确定子更新数据分布信息,其中,所述子更新数据分布信息包括:所述服务更新请求中各数据标识对应的子更新数据所在的服务节点的信息;

10 [169] 根据所述子更新数据分布信息和所述数据标识,选取一个服务节点作为目标服务节点。

[170] 可选地,计算机可执行指令在被执行时,所述根据所述子更新数据分布信息和所述数据标识,选取一个服务节点作为目标服务节点,包括:

15 [171] 基于所述子更新数据分布信息,将已存储所述数据标识对应的子更新数据的多个服务节点作为备选服务节点;

[172] 根据各所述备选服务节点的负载率,将负载率最低的备选服务节点确定为目标服务节点。

[173] 可选地,计算机可执行指令在被执行时,在从所述目标服务节点中获取所述数据标识对应的子更新数据之后,还包括:

20 [174] 若子更新数据获取失败,则按照负载率由低至高的顺序重新选择下一个目标服务节点,并从重新选择的所述目标服务节点中获取所述数据标识对应的子更新数据。

[175] 可选地,计算机可执行指令在被执行时,所述分别获取第一服务节点以外的多个服务节点的子更新数据存储信息,包括:

25 [176] 接收第一服务节点以外的多个服务节点广播的子更新数据存储信息,其中,所述子更新数据存储信息为第一服务节点以外的多个服务节点在获取到新的子更新数据且更新本地的子更新数据存储信息后得到的。

[177] 可选地,计算机可执行指令在被执行时,所述分别获取第一服务节点以外的多

个服务节点的子更新数据存储信息，包括：

[178] 向第一服务节点以外的各服务节点发送子数据存储信息获取请求；

[179] 接收第一服务节点以外的各服务节点发送的本地最新版本的子更新数据存储信息，其中，所述子更新数据存储信息为第一服务节点以外的各服务节点响应于所述子数
5 据存储信息获取请求发送的。

[180] 可选地，计算机可执行指令在被执行时，所述将获取到的所述多个子更新数据进行合并，得到所述服务更新数据，包括：

[181] 按照数据标识由小到大的顺序，对所述多个子更新数据进行排序；

[182] 根据排序结果，将各所述子更新数据组合得到所述服务更新数据。

10 [183] 可选地，计算机可执行指令在被执行时，还包含用于进行以下计算机可执行指令：

[184] 接收第一服务节点以外的各服务节点发送的子更新数据获取请求；

[185] 将与所述获取请求对应的子更新数据发送至相应的服务节点。

15 [186] 可选地，计算机可执行指令在被执行时，还包含用于进行以下计算机可执行指令：

[187] 将本地最新版本的子更新数据存储信息发送至相应的服务节点。

[188] 本申请实施例中的服务更新设备，在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据；将获取到的多个子更新数据进行合并，得到服务更新数据；根据合并得到的服务更新数据，对第一服务节点进行服务更新。

20 可见，通过本申请实施例中的服务更新设备，基于预先将用于服务节点进行服务更新的服务更新数据分割为多个子更新数据，多个待服务更新的服务节点共用参与各子更新数据的扩散，任一服务节点用于服务更新的数据是从其他关联服务节点中收集所有子更新数据后再合并得到的，有效利用多个服务节点的计算资源和网路资源，提高了多服务节点的服务更新效率，实现快速、高效、安全地对各服务节点进行服务更新。

25 [189] 进一步地，对应上述图 2 至图 5 所示的方法，基于相同的技术构思，本申请实施例还提供了一种存储介质，用于存储计算机可执行指令，一种具体的实施例中，该存储介质可以为 U 盘、光盘、硬盘等，该存储介质存储的计算机可执行指令在被处理器执

行时，能实现以下流程：

[190] 在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；

[191] 将获取到的所述多个子更新数据进行合并，得到所述服务更新数据；

5 [192] 根据合并得到的所述服务更新数据，对所述第一服务节点进行服务更新。

[193] 可选地，该存储介质存储的计算机可执行指令在被处理器执行时，还实现以下流程：在从第一服务节点以外的多个服务节点中获取多个子更新数据之前，还包括：

[194] 接收服务更新请求，其中，所述服务更新请求携带有由服务更新数据分割得到的所有子更新数据的数据标识；

10 [195] 所述从第一服务节点以外的多个服务节点中获取多个子更新数据，包括：

[196] 根据所述服务更新请求，从第一服务节点以外的多个服务节点中分别获取与各所述数据标识对应的子更新数据。

[197] 可选地，该存储介质存储的计算机可执行指令在被处理器执行时，所述接收服务更新请求，包括：

15 [198] 接收初次分发的至少一个子更新数据和更新数据分割信息，其中，所述更新数据分割信息包括：由服务更新数据分割得到的所有子更新数据的数据标识，且每个子更新数据均被分发到至少一个服务节点。

[199] 可选地，该存储介质存储的计算机可执行指令在被处理器执行时，由服务更新数据分割得到的每个子更新数据均被分发到至少两个服务节点。

20 [200] 可选地，该存储介质存储的计算机可执行指令在被处理器执行时，所述根据所述服务更新请求，从第一服务节点以外的多个服务节点中分别获取与各所述数据标识对应的子更新数据，包括：

[201] 分别获取第一服务节点以外的多个服务节点的子更新数据存储信息，其中，所述子更新数据存储信息包括：服务节点上已存储的子更新数据的数据标识；

25 [202] 针对所述服务更新请求中每个数据标识，判断所述第一服务节点是否存在与该数据标识对应的子更新数据；

[203] 若否，则根据各所述子更新数据存储信息和所述数据标识，选取一个服务节点

作为目标服务节点, 以及从所述目标服务节点中获取所述数据标识对应的子更新数据。

[204] 可选地, 该存储介质存储的计算机可执行指令在被处理器执行时, 所述根据各所述子更新数据存储信息和所述数据标识, 选取一个服务节点作为目标服务节点, 包括:

5 [205] 根据各所述子更新数据存储信息, 确定子更新数据分布信息, 其中, 所述子更新数据分布信息包括: 所述服务更新请求中各数据标识对应的子更新数据所在的服务节点的信息;

[206] 根据所述子更新数据分布信息和所述数据标识, 选取一个服务节点作为目标服务节点。

10 [207] 可选地, 该存储介质存储的计算机可执行指令在被处理器执行时, 所述根据所述子更新数据分布信息和所述数据标识, 选取一个服务节点作为目标服务节点, 包括:

[208] 基于所述子更新数据分布信息, 将已存储所述数据标识对应的子更新数据的多个服务节点作为备选服务节点;

[209] 根据各所述备选服务节点的负载率, 将负载率最低的备选服务节点确定为目标服务节点。

15 [210] 可选地, 该存储介质存储的计算机可执行指令在被处理器执行时, 在从所述目标服务节点中获取所述数据标识对应的子更新数据之后, 还包括:

[211] 若子更新数据获取失败, 则按照负载率由低至高的顺序重新选择下一个目标服务节点, 并从重新选择的所述目标服务节点中获取所述数据标识对应的子更新数据。

20 [212] 可选地, 该存储介质存储的计算机可执行指令在被处理器执行时, 所述分别获取第一服务节点以外的多个服务节点的子更新数据存储信息, 包括:

[213] 接收第一服务节点以外的多个服务节点广播的子更新数据存储信息, 其中, 所述子更新数据存储信息为第一服务节点以外的多个服务节点在获取到新的子更新数据且更新本地的子更新数据存储信息后得到的。

25 [214] 可选地, 该存储介质存储的计算机可执行指令在被处理器执行时, 所述分别获取第一服务节点以外的多个服务节点的子更新数据存储信息, 包括:

[215] 向第一服务节点以外的各服务节点发送子数据存储信息获取请求;

[216] 接收第一服务节点以外的各服务节点发送的本地最新版本的子更新数据存储信

息，其中，所述子更新数据存储信息为第一服务节点以外的各服务节点响应于所述子数据存储信息获取请求发送的。

[217] 可选地，该存储介质存储的计算机可执行指令在被处理器执行时，所述将获取到的所述多个子更新数据进行合并，得到所述服务更新数据，包括：

5 [218] 按照数据标识由小到大的顺序，对所述多个子更新数据进行排序；

[219] 根据排序结果，将各所述子更新数据组合得到所述服务更新数据。

[220] 可选地，该存储介质存储的计算机可执行指令在被处理器执行时，还实现以下流程：

[221] 接收第一服务节点以外的各服务节点发送的子更新数据获取请求；

10 [222] 将与所述获取请求对应的子更新数据发送至相应的服务节点。

[223] 可选地，该存储介质存储的计算机可执行指令在被处理器执行时，还实现以下流程：

[224] 将本地最新版本的子更新数据存储信息发送至相应的服务节点。

[225] 本申请实施例中的存储介质存储的计算机可执行指令在被处理器执行时，在待
15 服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据；将获取到的多个子更新数据进行合并，得到服务更新数据；根据合并得到的服务更新数据，对第一服务节点进行服务更新。可见，通过本申请实施例中的存储介质，基于预先将用于服务节点进行服务更新的服务更新数据分割为多个子更新数据，多个待服务更新的服务节点共用参与各子更新数据的扩散，任一服务节点用于服务更新的数据是从
20 其他关联服务节点中收集所有子更新数据后再合并得到的，有效利用多个服务节点的计算资源和网路资源，提高了多服务节点的服务更新效率，实现快速、高效、安全地对各服务节点进行服务更新。

[226] 在 20 世纪 90 年代，对于一个技术的改进可以很明显地区分是硬件上的改进（例如，对二极管、晶体管、开关等电路结构的改进）还是软件上的改进（对于方法流程的改进）。然而，随着技术的发展，当今的很多方法流程的改进已经可以视为硬件电路结构的直接改进。设计人员几乎都通过将改进的方法流程编程到硬件电路中来得到相应的硬件电路结构。因此，不能说一个方法流程的改进就不能用硬件实体模块来实现。例如，
25 可编程逻辑器件（Programmable Logic Device, PLD）（例如现场可编程门阵列（Field

Programmable Gate Array, FPGA))就是这样一种集成电路,其逻辑功能由用户对器件编程来确定。由设计人员自行编程来把一个数字系统“集成”在一片 PLD 上,而不需要请芯片制造厂商来设计和制作专用的集成电路芯片。而且,如今,取代手工地制作集成电路芯片,这种编程也多半改用“逻辑编译器(logic compiler)”软件来实现,它与程序

5 开发撰写时所用的软件编译器相类似,而要编译之前的原始代码也得用特定的编程语言来撰写,此称之为硬件描述语言(Hardware Description Language, HDL),而 HDL 也并非仅有一种,而是有许多种,如 ABEL(Advanced Boolean Expression Language)、AHDL(Altera Hardware Description Language)、Confluence、CUPL(Cornell University Programming Language)、HDCal、JHDL(Java Hardware Description Language)、Lava、

10 Lola、MyHDL、PALASM、RHDL(Ruby Hardware Description Language)等,目前最普遍使用的是 VHDL(Very-High-Speed Integrated Circuit Hardware Description Language)与 Verilog。本领域技术人员也应该清楚,只需要将方法流程用上述几种硬件描述语言稍作逻辑编程并编程到集成电路中,就可以很容易得到实现该逻辑方法流程的硬件电路。

15 [227] 控制器可以按任何适当的方式实现,例如,控制器可以采取例如微处理器或处理器以及存储可由该(微)处理器执行的计算机可读程序代码(例如软件或固件)的计算机可读介质、逻辑门、开关、专用集成电路(Application Specific Integrated Circuit, ASIC)、可编程逻辑控制器和嵌入微控制器的形式,控制器的例子包括但不限于以下微

20 控制器:ARC 625D、Atmel AT91SAM、Microchip PIC18F26K20 以及 Silicone Labs C8051F320,存储器控制器还可以被实现为存储器的控制逻辑的一部分。本领域技术人员也知道,除了以纯计算机可读程序代码方式实现控制器以外,完全可以通过将方法步骤进行逻辑编程来使得控制器以逻辑门、开关、专用集成电路、可编程逻辑控制器和嵌入微控制器等的形式来实现相同功能。因此这种控制器可以被认为是一种硬件部件,而对其内包括的用于实现各种功能的装置也可以视为硬件部件内的结构。或者甚至,可以

25 将用于实现各种功能的装置视为既可以是实现方法的软件模块又可以是硬件部件内的结构。

[228] 上述实施例阐明的系统、装置、模块或单元,具体可以由计算机芯片或实体实现,或者由具有某种功能的产品来实现。一种典型的实现设备为计算机。具体的,计算机例如可以为个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字

30 助理、媒体播放器、导航设备、电子邮件设备、游戏控制台、平板计算机、可穿戴设备

或者这些设备中的任何设备的组合。

[229] 为了描述的方便，描述以上装置时以功能分为各种单元分别描述。当然，在实施本申请时可以把各单元的功能在同一个或多个软件和/或硬件中实现。

5 [230] 本领域内的技术人员应明白，本申请的实施例可提供为方法、系统、或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

10 [231] 本申请是参照根据本申请实施例的方法、设备（系统）、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[232] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

20 [233] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

25 [234] 在一个典型的配置中，计算设备包括一个或多个处理器（CPU）、输入/输出接口、网络接口和内存。

[235] 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（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。

[236] 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暂存电脑可读媒体（transitory media），如调制的数据信号和载波。

5

10

[237] 还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

15

[238] 本领域技术人员应明白，本申请的实施例可提供为方法、系统或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

20

[239] 本申请可以在由计算机执行的计算机可执行指令的一般上下文中描述，例如程序模块。一般地，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本申请，在这些分布式计算环境中，由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中，程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

25

[240] 本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于系统实施例而言，由于其基本相似于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

[241] 以上所述仅为本申请的实施例而已，并不用于限制本申请。对于本领域技术人

员来说, 本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等, 均应包含在本申请的权利要求范围之内。

权利要求书

1、一种服务更新方法，应用于任一待服务更新的第一服务节点，其特征在于，包括：

在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个

5 子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；

将获取到的所述多个子更新数据进行合并，得到所述服务更新数据；

根据合并得到的所述服务更新数据，对所述第一服务节点进行服务更新。

2、根据权利要求 1 所述的方法，其特征在于，在从第一服务节点以外的多个服务
10 节点中获取多个子更新数据之前，还包括：

接收服务更新请求，其中，所述服务更新请求携带有由服务更新数据分割得到的所有子更新数据的数据标识；

所述从第一服务节点以外的多个服务节点中获取多个子更新数据，包括：

15 根据所述服务更新请求，从第一服务节点以外的多个服务节点中分别获取与各所述数据标识对应的子更新数据。

3、根据权利要求 2 所述的方法，其特征在于，所述接收服务更新请求，包括：

接收初次分发的至少一个子更新数据和更新数据分割信息，其中，所述更新数据分割信息包括：由服务更新数据分割得到的所有子更新数据的数据标识，且每个子更新数
20 据均被分发到至少一个服务节点。

4、根据权利要求 3 所述的方法，其特征在于，由服务更新数据分割得到的每个子更新数据均被分发到至少两个服务节点。

25 5、根据权利要求 2 所述的方法，其特征在于，所述根据所述服务更新请求，从第一服务节点以外的多个服务节点中分别获取与各所述数据标识对应的子更新数据，包括：

分别获取第一服务节点以外的多个服务节点的子更新数据存储信息，其中，所述子更新数据存储信息包括：服务节点上已存储的子更新数据的数据标识；

30 针对所述服务更新请求中每个数据标识，判断所述第一服务节点是否存在与该数据标识对应的子更新数据；

若否，则根据各所述子更新数据存储信息和所述数据标识，选取一个服务节点作为目标服务节点，以及从所述目标服务节点中获取所述数据标识对应的子更新数据。

5 6、根据权利要求 5 所述的方法，其特征在于，所述根据各所述子更新数据存储信息和所述数据标识，选取一个服务节点作为目标服务节点，包括：

根据各所述子更新数据存储信息，确定子更新数据分布信息，其中，所述子更新数据分布信息包括：所述服务更新请求中各数据标识对应的子更新数据所在的服务节点的信息；

10 根据所述子更新数据分布信息和所述数据标识，选取一个服务节点作为目标服务节点。

7、根据权利要求 6 所述的方法，其特征在于，所述根据所述子更新数据分布信息和所述数据标识，选取一个服务节点作为目标服务节点，包括：

15 基于所述子更新数据分布信息，将已存储所述数据标识对应的子更新数据的多个服务节点作为备选服务节点；

根据各所述备选服务节点的负载率，将负载率最低的备选服务节点确定为目标服务节点。

20 8、根据权利要求 7 所述的方法，其特征在于，在从所述目标服务节点中获取所述数据标识对应的子更新数据之后，还包括：

若子更新数据获取失败，则按照负载率由低至高的顺序重新选择下一个目标服务节点，并从重新选择的所述目标服务节点中获取所述数据标识对应的子更新数据。

25 9、根据权利要求 5 所述的方法，其特征在于，所述分别获取第一服务节点以外的多个服务节点的子更新数据存储信息，包括：

接收第一服务节点以外的多个服务节点广播的子更新数据存储信息，其中，所述子更新数据存储信息为第一服务节点以外的多个服务节点在获取到新的子更新数据且更新本地的子更新数据存储信息后得到的。

30 10、根据权利要求 5 所述的方法，其特征在于，所述分别获取第一服务节点以外的多个服务节点的子更新数据存储信息，包括：

向第一服务节点以外的各服务节点发送子数据存储信息获取请求；

接收第一服务节点以外的各服务节点发送的本地最新版本的子更新数据存储信息，其中，所述子更新数据存储信息为第一服务节点以外的各服务节点响应于所述子数据存储信息获取请求发送的。

5 11、根据权利要求 2 所述的方法，其特征在于，所述将获取到的所述多个子更新数据进行合并，得到所述服务更新数据，包括：

按照数据标识由小到大的顺序，对所述多个子更新数据进行排序；

根据排序结果，将各所述子更新数据组合得到所述服务更新数据。

10 12、根据权利要求 1 至 11 任一项所述的方法，其特征在于，还包括：

接收第一服务节点以外的各服务节点发送的子更新数据获取请求；

将与所述获取请求对应的子更新数据发送至相应的服务节点。

13、根据权利要求 1 至 11 任一项所述的方法，其特征在于，还包括：

15 将本地最新版本的子更新数据存储信息发送至相应的服务节点。

14、一种服务更新装置，设置于任一待服务更新的第一服务节点，其特征在于，包括：

20 子数据获取模块，用于在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；

更新数据生成模块，用于将获取到的所述多个子更新数据进行合并，得到所述服务更新数据；

25 服务更新模块，用于根据合并得到的所述服务更新数据，对所述第一服务节点进行服务更新。

15、一种服务更新系统，包括：待服务更新的多个服务节点，两两服务节点之间通信连接，每个服务节点设置有如权利要求 14 所述的服务更新装置。

30 16、根据权利要求 15 所述的系统，其特征在于，每个所述服务节点，用于：

接收服务更新请求，其中，所述服务更新请求携带有由服务更新数据分割得到的所有子更新数据的数据标识；

根据所述服务更新请求，分别获取与各所述数据标识对应的子更新数据。

17、根据权利要求 16 所述的系统，其特征在于，所述多个服务节点中任一服务节点，用于：

5 接收服务更新数据和数据分割指示；

根据所述数据分割指示，将所述服务更新数据分割为多个子更新数据；

从多个待服务更新的服务节点中，选取目标数量的多个初始分配节点；

向所述多个初始分配节点发送服务更新请求，其中，所述服务更新请求携带有初次分发的至少一个子更新数据和更新数据分割信息，每个子更新数据被分发给至少一个已

10 选取的初始分配节点。

18、根据权利要求 17 所述的系统，其特征在于，每个子更新数据被分发给至少两个已选取的初始分配节点。

15 19、一种服务更新设备，其特征在于，包括：

处理器；以及

被安排成存储计算机可执行指令的存储器，所述可执行指令在被执行时使所述处理器：

20 在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；

将获取到的所述多个子更新数据进行合并，得到所述服务更新数据；

根据合并得到的所述服务更新数据，对所述第一服务节点进行服务更新。

25 20、一种存储介质，用于存储计算机可执行指令，其特征在于，所述可执行指令在被执行时实现以下流程：

在待服务更新的多个服务节点中，从第一服务节点以外的多个服务节点中获取多个子更新数据，其中，所述多个子更新数据为预先由服务更新数据分割后得到的；

将获取到的所述多个子更新数据进行合并，得到所述服务更新数据；

根据合并得到的所述服务更新数据，对所述第一服务节点进行服务更新。

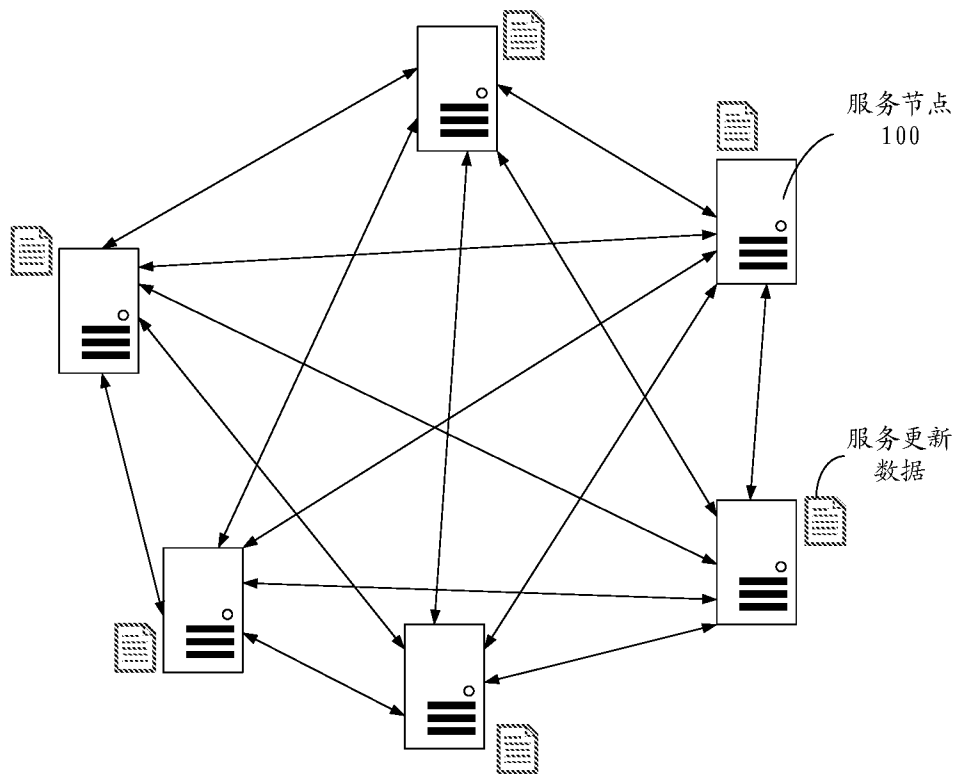


图 1

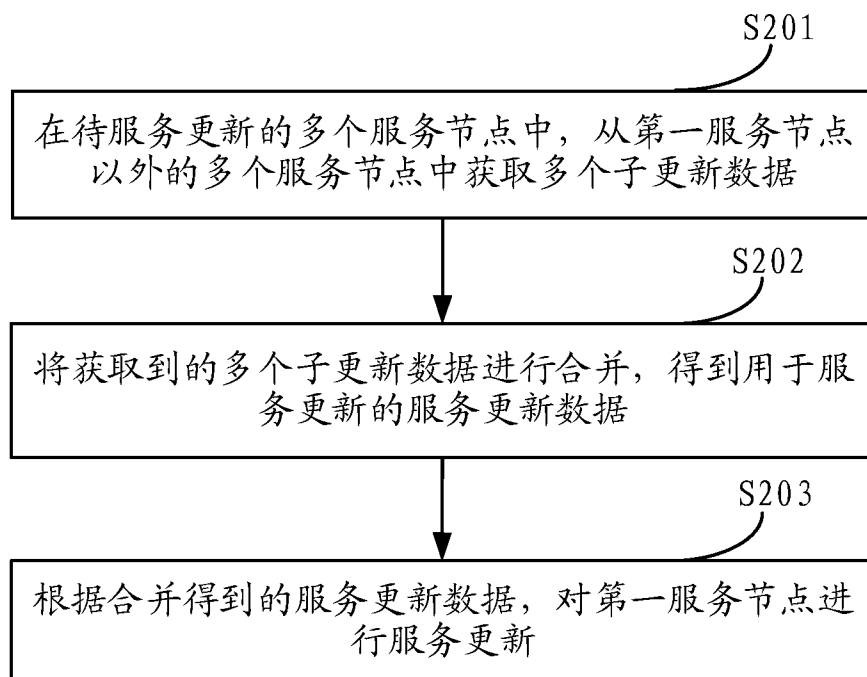


图 2

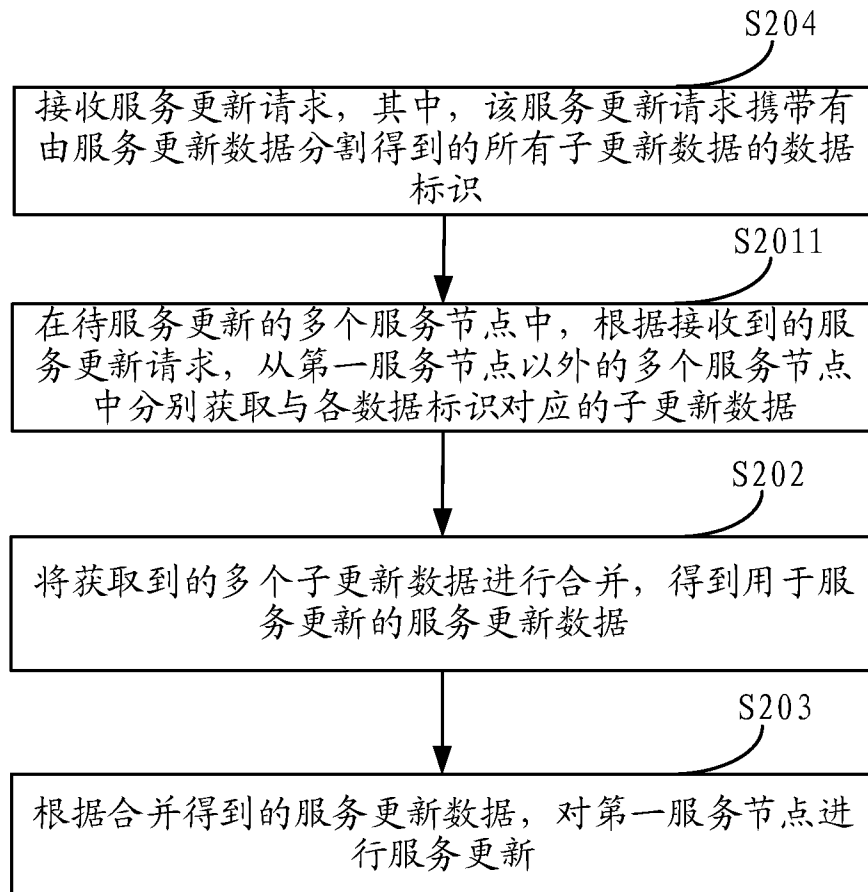


图 3

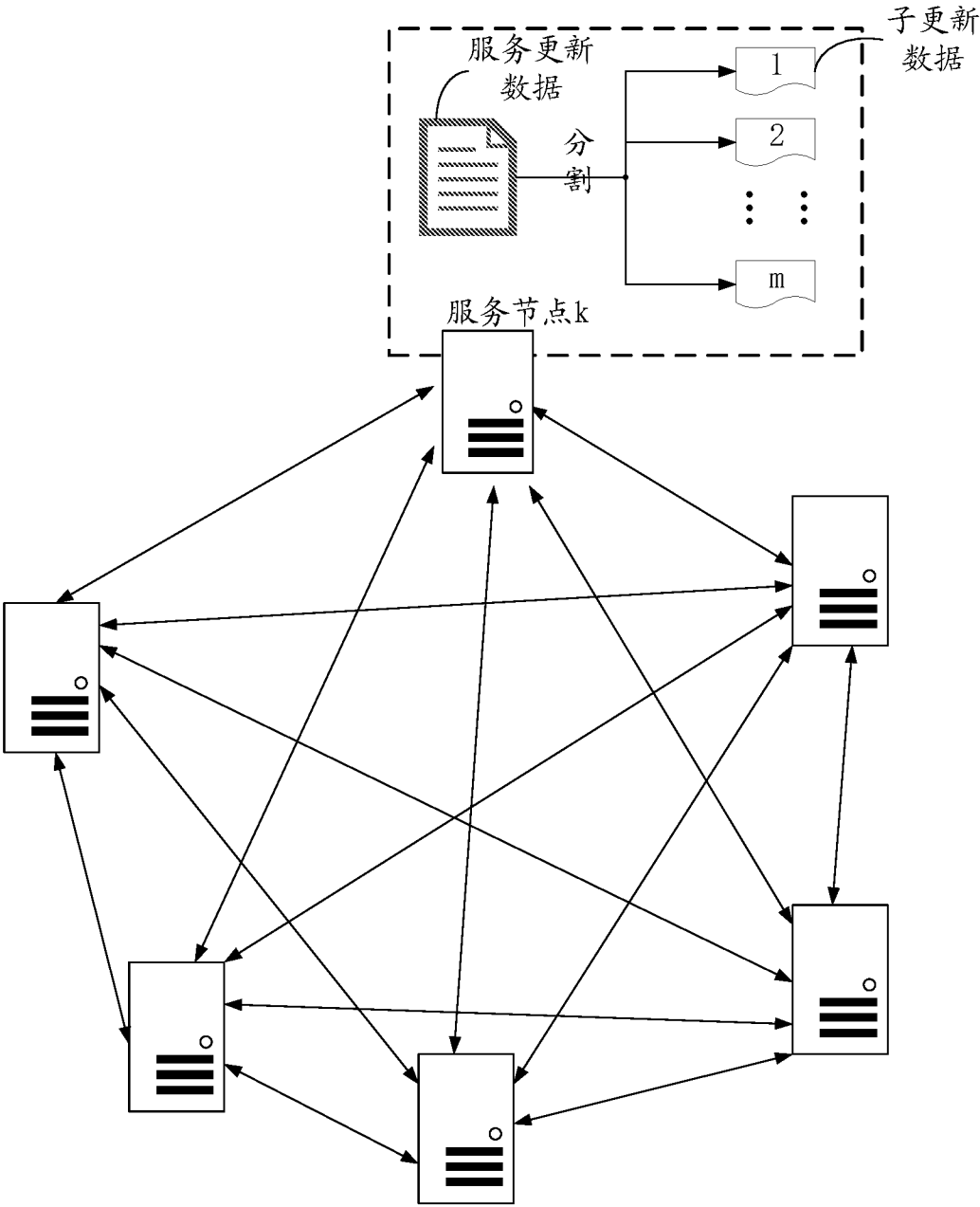


图 4a

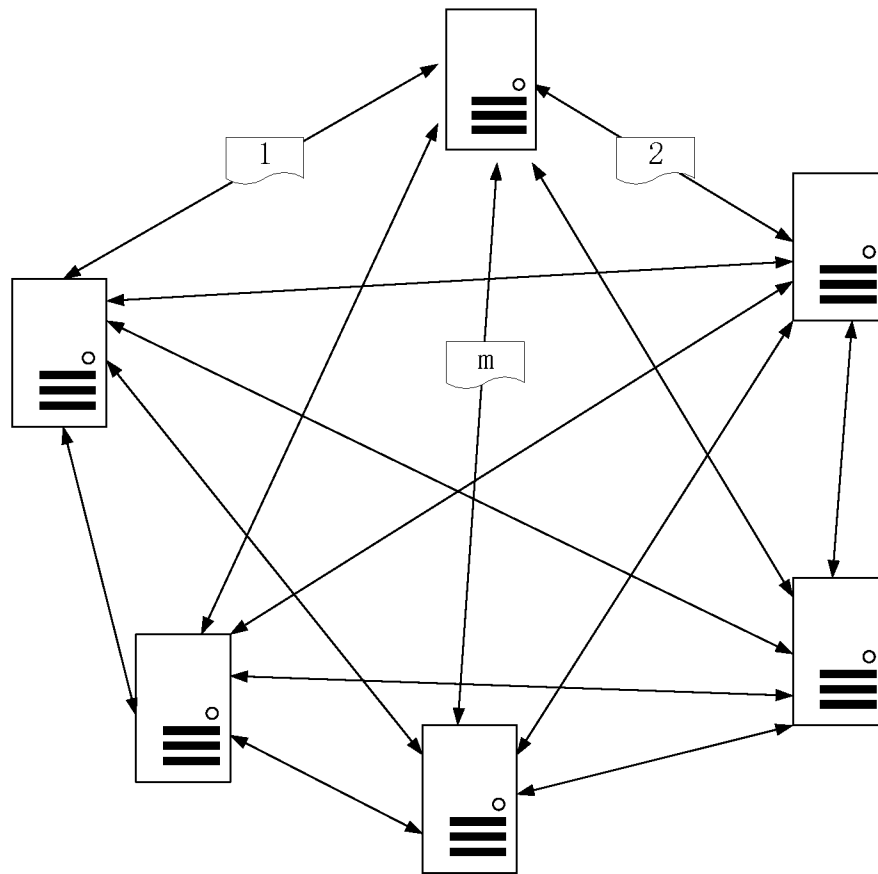


图 4b

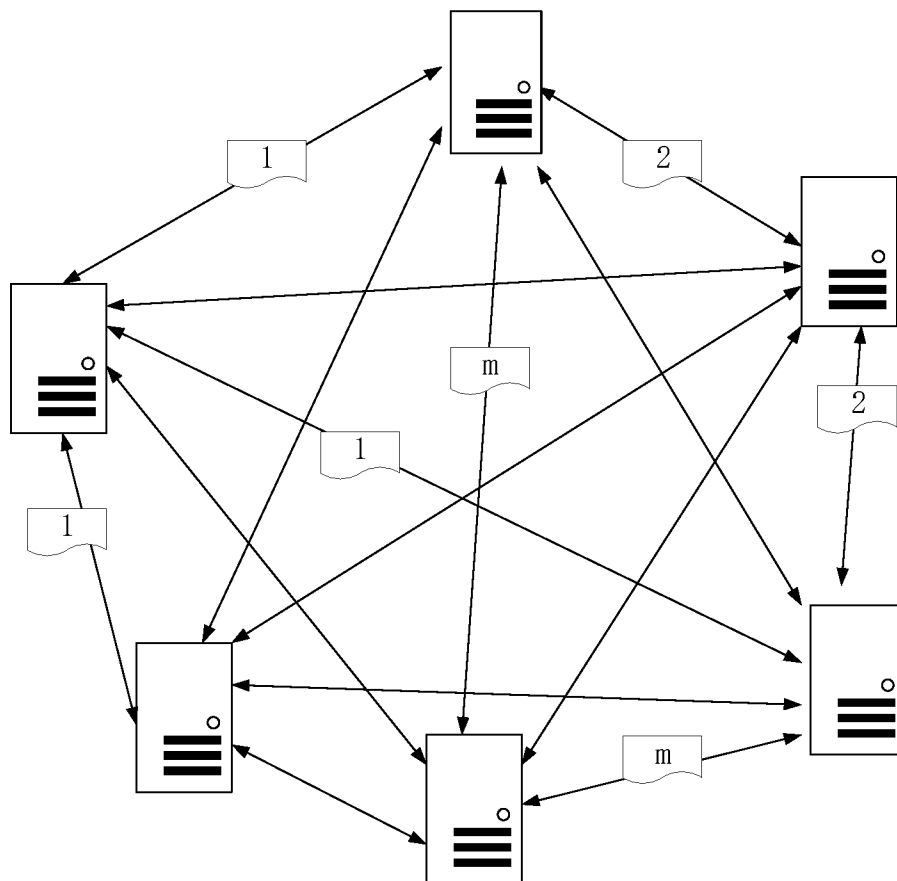


图 4c

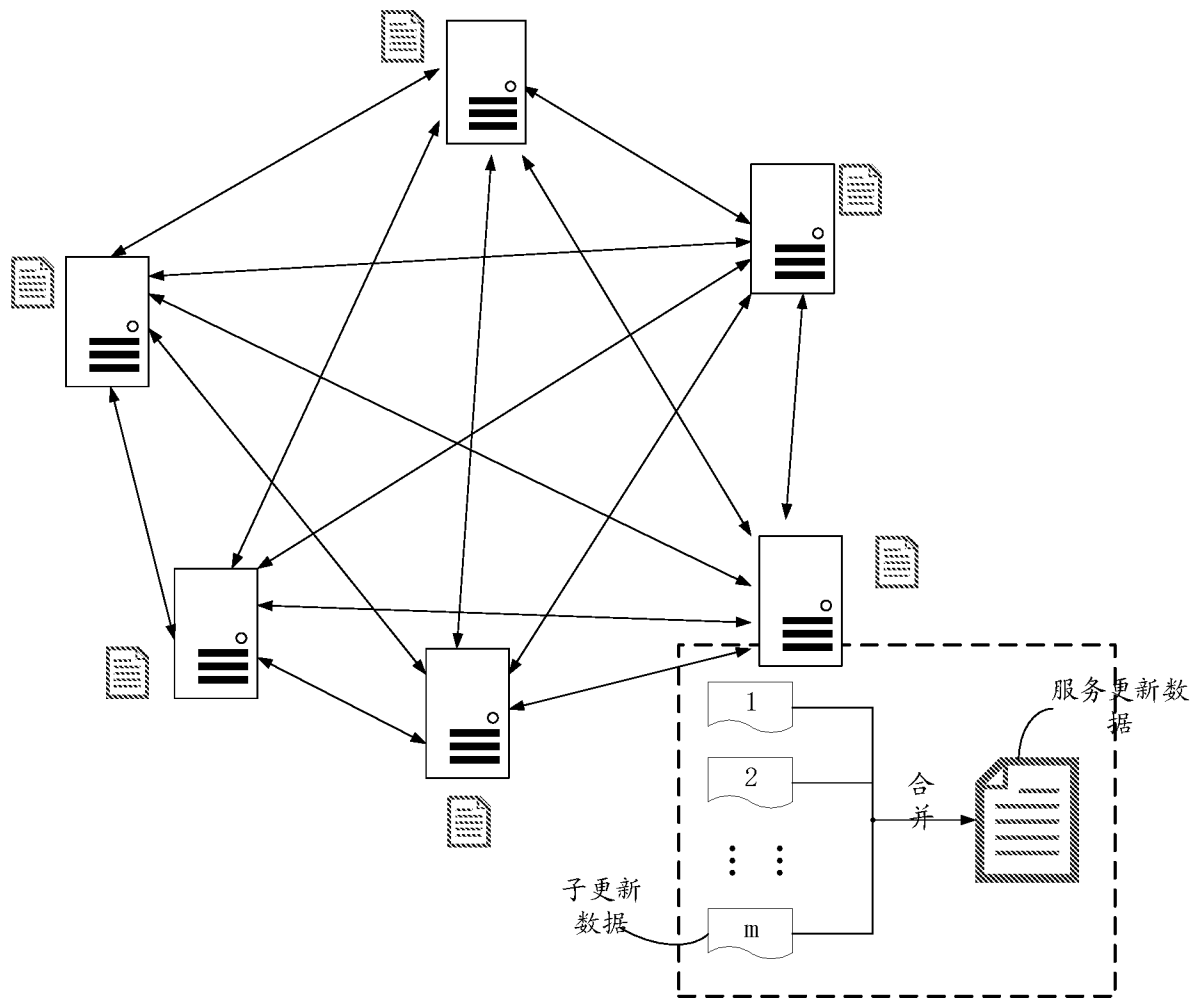


图 4d

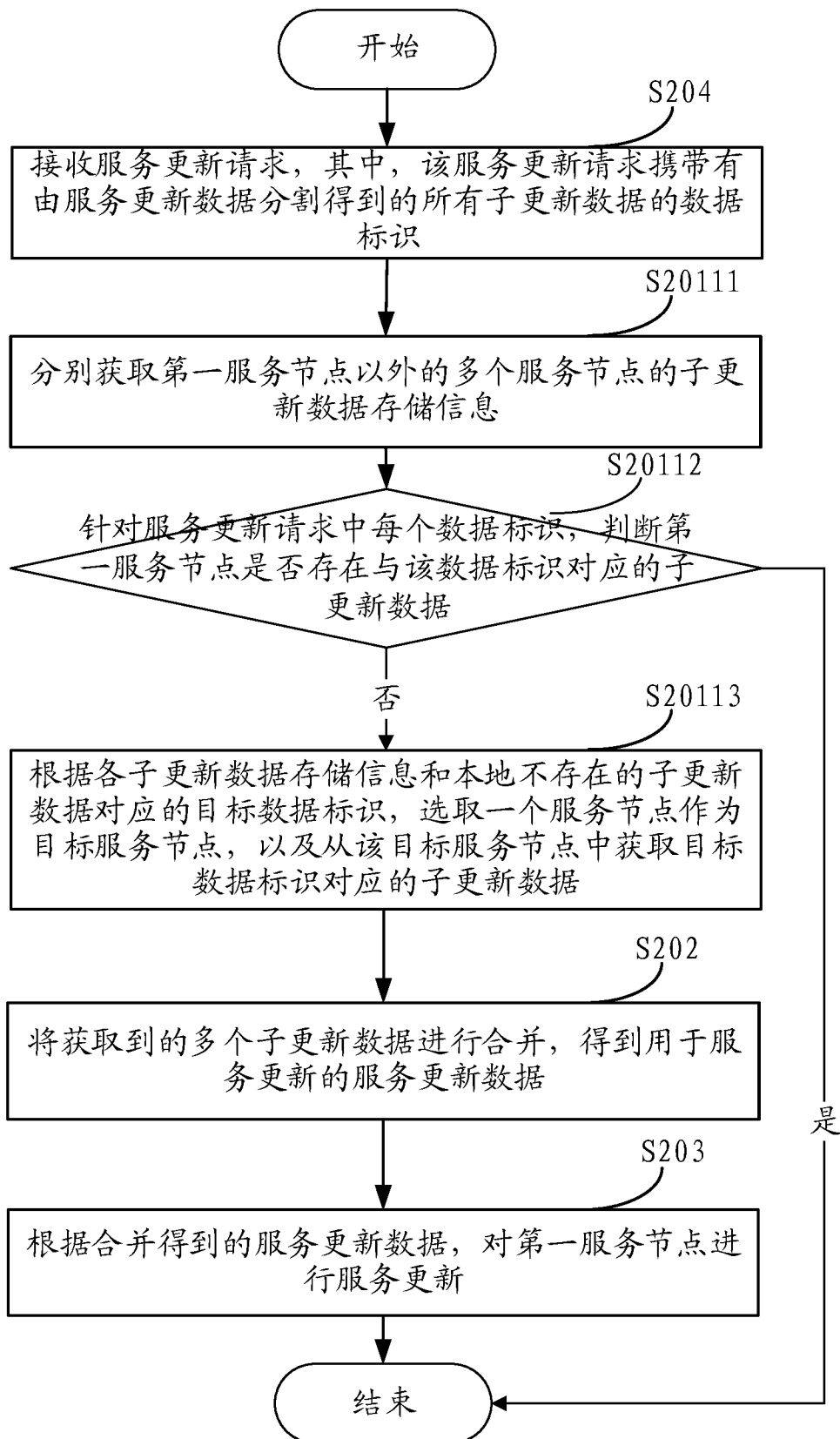


图 5

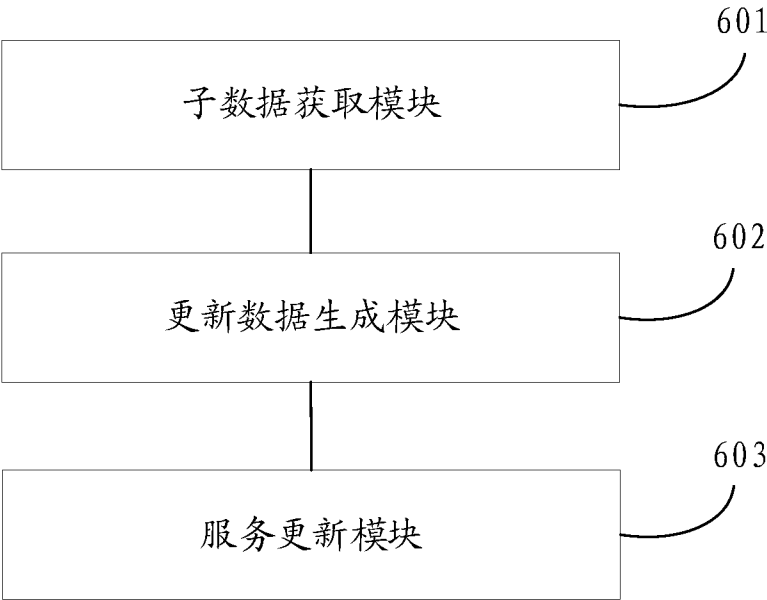


图 6

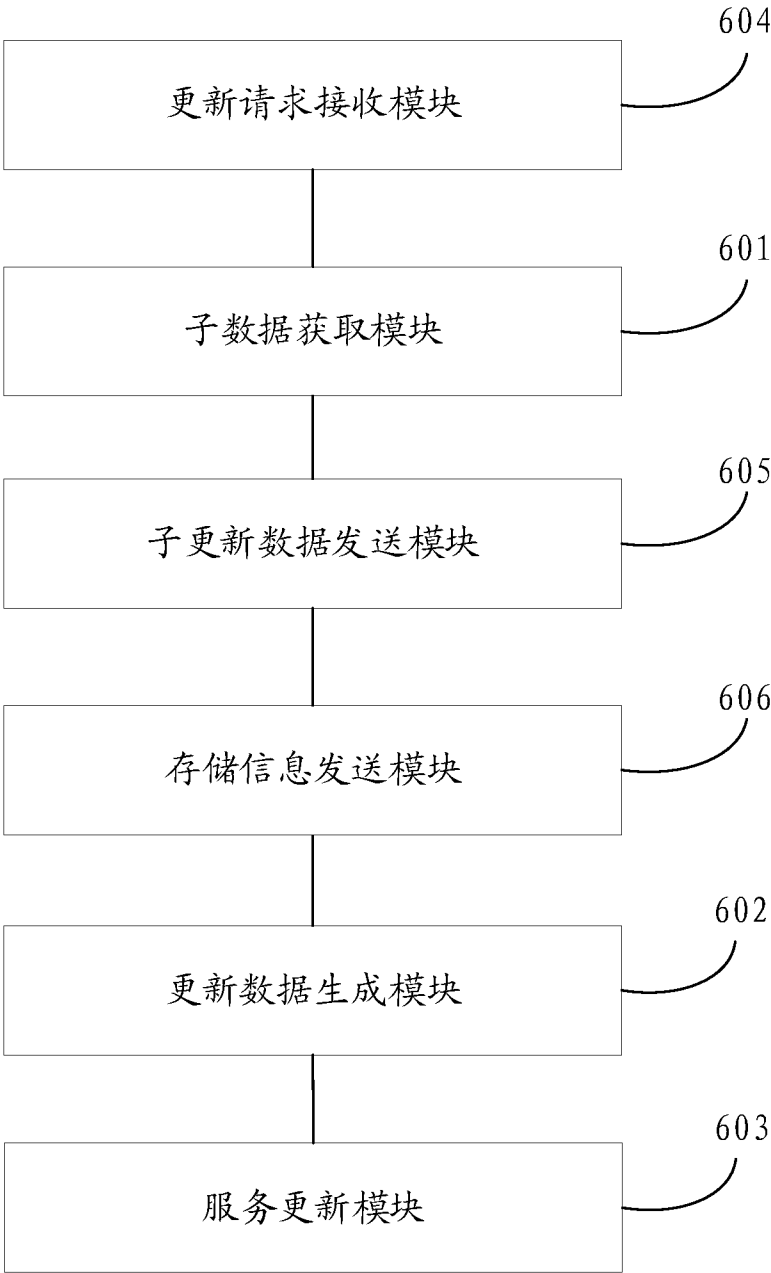


图 7

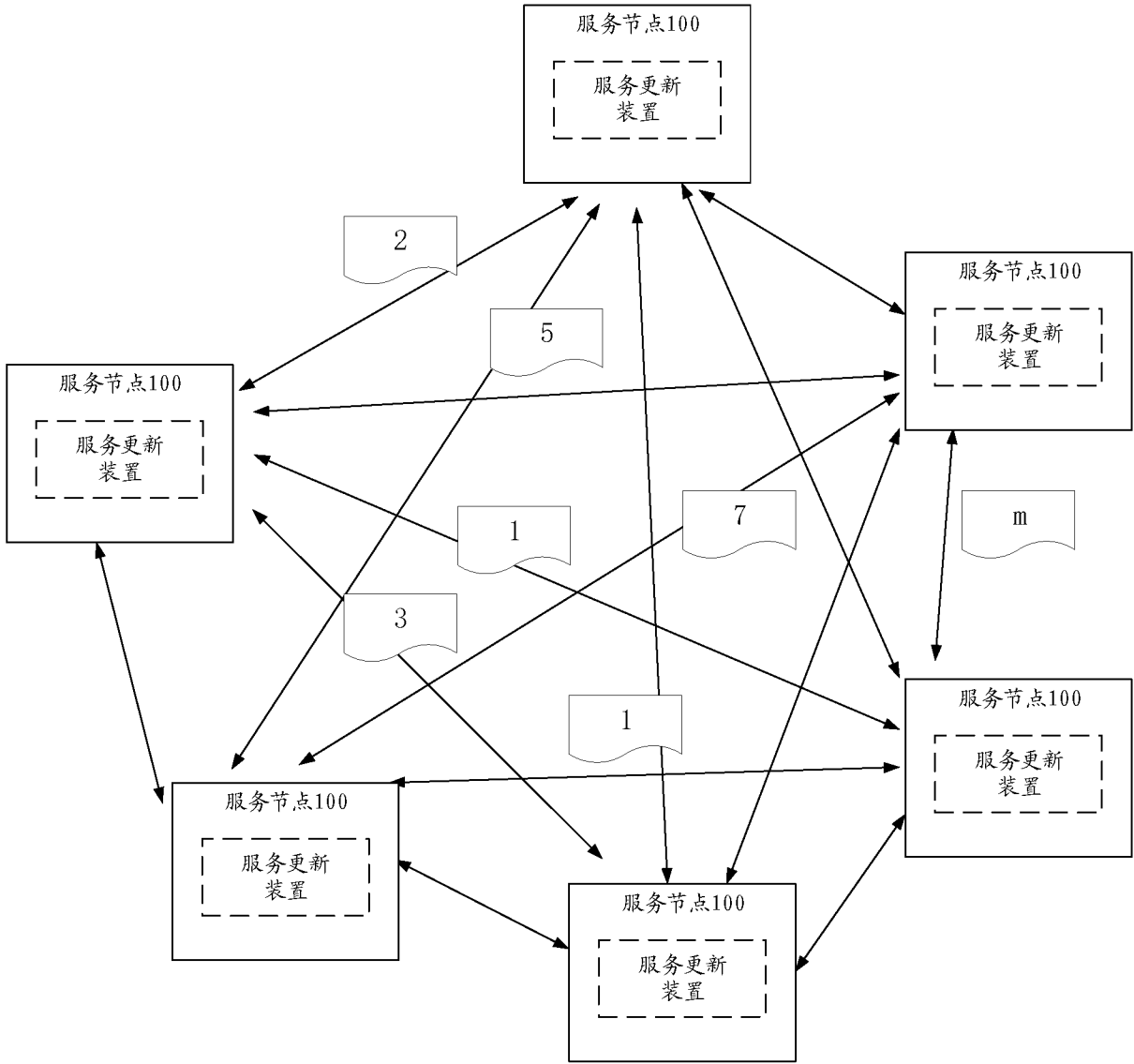


图 8

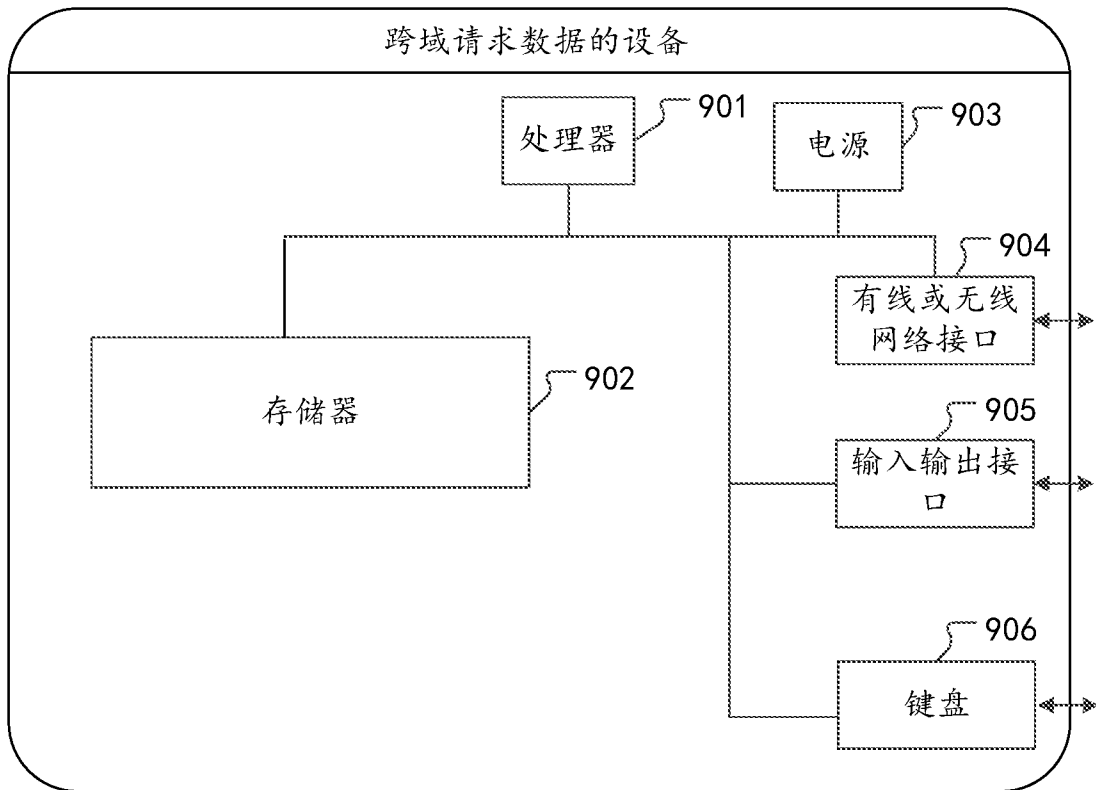


图 9

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2019/071136

A. CLASSIFICATION OF SUBJECT MATTER

H04L 29/08(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS; CNTXT; VEN; CNKI: 更新, 节点, 服务, 内容, 数据, 文件, 分割, 分散, 分裂, update, node, service, data, device

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 108390914 A (ALIBABA GROUP HOLDING LIMITED) 10 August 2018 (2018-08-10) claims 1-20	1-20
X	CN 101374072 A (SIEMENS AG.) 25 February 2009 (2009-02-25) description, page 5, paragraph 1 to page 12, paragraph 3	1-20
A	CN 107315792 A (BEIJING XIAOMI MOBILE SOFTWARE CO., LTD.) 03 November 2017 (2017-11-03) entire document	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

20 March 2019

Date of mailing of the international search report

29 March 2019

Name and mailing address of the ISA/CN

National Intellectual Property Administration, PRC (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing
100088
China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2019/071136

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	108390914	A	10 August 2018	None			
CN	101374072	A	25 February 2009	CN	101374072	B	23 November 2011
CN	107315792	A	03 November 2017	EP	3416055	A1	19 December 2018
				US	2018365200	A1	20 December 2018

国际检索报告

国际申请号

PCT/CN2019/071136

A. 主题的分类

H04L 29/08 (2006.01) i

按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类

B. 检索领域

检索的最低限度文献 (标明分类系统和分类号)

H04L

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用))

CNABS; CNTXT; VEN; CNKI: 更新, 节点, 服务, 内容, 数据, 文件, 分割, 分散, 分裂, update, node, service, data, device

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
PX	CN 108390914 A (阿里巴巴集团控股有限公司) 2018年 8月 10日 (2018 - 08 - 10) 权利要求1-20	1-20
X	CN 101374072 A (西门子子公司) 2009年 2月 25日 (2009 - 02 - 25) 说明书第5页第1段-第12页第3段	1-20
A	CN 107315792 A (北京小米移动软件有限公司) 2017年 11月 3日 (2017 - 11 - 03) 全文	1-20

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2019年 3月 20日

国际检索报告邮寄日期

2019年 3月 29日

ISA/CN的名称和邮寄地址

中国国家知识产权局 (ISA/CN)

中国北京市海淀区蓟门桥西土城路6号 100088

传真号 (86-10)62019451

受权官员

白晶心

电话号码 86-(010)-62412220

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2019/071136

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	108390914	A	2018年 8月 10日	无			
CN	101374072	A	2009年 2月 25日	CN	101374072	B	2011年 11月 23日
CN	107315792	A	2017年 11月 3日	EP	3416055	A1	2018年 12月 19日
				US	2018365200	A1	2018年 12月 20日