(12) STANDARD PATENT (11) Application No. **AU 2017282817 B2**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Data processing method and device

(51) International Patent Classification(s)
H04L 29/08 (2006.01)

(21) Application No: **2017282817** (22) Date of Filing: **2017.06.07**

(87) WIPO No: **WO17/219857**

(30) Priority Data

(31) Number	(32) Date	(33) Country
201610444320.5	**2016.06.20**	**CN**

(43) Publication Date: **2017.12.28**
(44) Accepted Journal Date: **2020.01.02**

(71) Applicant(s)
Alibaba Group Holding Limited

(72) Inventor(s)
Yang, Zhenkun;Zhao, Yuzhong;Shi, Wenhui

(74) Agent / Attorney
Pizzeys Patent and Trade Mark Attorneys Pty Ltd, PO Box 291, WODEN, ACT, 2606, AU

(56) Related Art
CN 104657382 A
EP 2429134 A1

(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
2017 年 12 月 28 日 (28.12.2017)

WIPO | PCT

(10) 国际公布号
WO 2017/219857 A1

(51) 国际专利分类号:
H04L 29/08 (2006.01)

(21) 国际申请号: PCT/CN2017/087378

(22) 国际申请日: 2017 年 6 月 7 日 (07.06.2017)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201610444320.5 2016年6月20日 (20.06.2016) 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: 阳振坤 (YANG, Zhenkun); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 赵裕众 (ZHAO, Yuzhong); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 师文汇 (SHI, Wenhui); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。

(74) 代理人: 北京三友知识产权代理有限公司 (BEIJING SANYOU INTELLECTUAL PROPERTY AGENCY LTD.); 中国北京市金融街 35 号国际企业大厦 A 座 16 层, Beijing 100033 (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, 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, 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。

(54) **Title:** DATA PROCESSING METHOD AND DEVICE

(54) 发明名称: 一种数据处理方法和设备

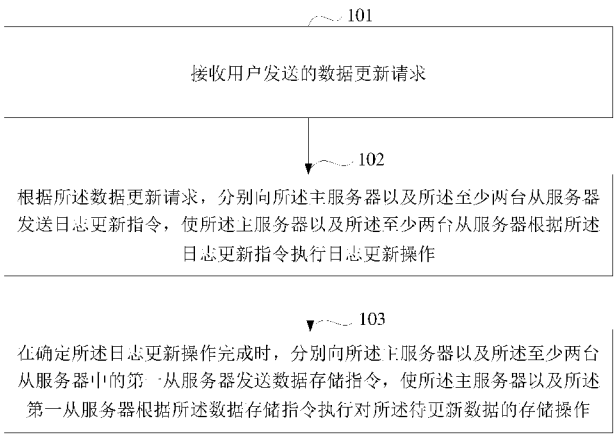

图 1

101 RECEIVING A DATA UPDATE REQUEST SENT BY A USER
102 ACCORDING TO THE DATA UPDATE REQUEST, RESPECTIVELY SENDING A LOG UPDATE INSTRUCTION TO A MASTER SERVER AND AT LEAST TWO SLAVE SERVERS, SO THAT THE MASTER SERVER AND THE AT LEAST TWO SLAVE SERVERS EXECUTE A LOG UPDATE OPERATION ACCORDING TO THE LOG UPDATE INSTRUCTION
103 WHEN IT IS DETERMINED THAT THE LOG UPDATE OPERATION IS COMPLETED, RESPECTIVELY SENDING A DATA STORAGE INSTRUCTION TO THE MASTER SERVER AND A FIRST SLAVE SERVER IN THE AT LEAST TWO SLAVE SERVERS, SO THAT THE MASTER SERVER AND THE FIRST SLAVE SERVER EXECUTE A STORAGE OPERATION ON DATA TO BE UPDATED ACCORDING TO THE DATE STORAGE INSTRUCTION

(57) **Abstract:** Disclosed are a data processing method and device. The method comprises: receiving a data update request sent by a user; according to the data update request, respectively sending a log update instruction to a master server and at least two slave servers; and when it is determined that a log update operation is completed, respectively sending a data storage instruction to the master server and a first slave server in the at least two slave servers. When a data update request is received, a log update instruction can be sent to a master server and slave servers, so that the master server and the slave servers execute a log update operation. When it is determined that the log update operation is completed, a data storage instruction is sent to the master server and one slave server therein. When data in the slave server is lost, the lost data can be recovered by means of a pre-stored log, thereby ensuring the data consistency between the master server and the slave servers. With regard to data to be updated, only the master server and one slave server store same, thereby efficiently reducing the resources consumed during data storage.

WO 2017/219857 A1

[见续页]

(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))。

(57) 摘要：本申请公开了一种数据处理方法和设备，接收用户发送的数据更新请求；根据数据更新请求，分别向主服务器以及至少两台从服务器发送日志更新指令；在确定日志更新操作完成时，分别向主服务器以及至少两台从服务器中的第一从服务器发送数据存储指令。在接收到数据更新请求时，可以向主服务器以及从服务器发送日志更新指令，使主服务器以及从服务器执行日志更新操作，在确定日志更新操作完成时，向主服务器以及其中一个从服务器发送数据存储指令，在从服务器中的数据丢失时，可以通过预先存储的日志恢复丢失的数据，保证主服务器与从服务器之间数据的一致性，针对待更新数据，只有主服务器以及一个从服务器对其进行存储，有效减少数据存储消耗的资源。

DATA PROCESSING METHOD AND DEVICE

[0001] The present application claims priority to Chinese Patent Application No. 201610444320.5, filed on June 20, 2016 and entitled "DATA PROCESSING METHOD AND DEVICE", which is incorporated here by reference in its entirety.

TECHNICAL FIELD

[0002] The present disclosure relates to the field of computer technologies, and in particular, to a data processing method and device.

BACKGROUND

[0003] In a distributed data storage system, one server serves as a primary server and another server serves as a dependent server (hereinafter referred to as a secondary server). The primary server can provide an external data read/write service. The secondary server maintains data synchronization with the primary server, and when the primary server fails, the secondary server can provide the external data read/write service.

[0004] Usually, when receiving and storing data, the primary server can synchronously store the data in the secondary server, so as to ensure data consistency between the primary server and the secondary server. However, in practice, because there is an operation delay in data synchronization between the primary server and the secondary server, some data in the secondary server may be lost, which cannot ensure data consistency between the secondary server and the primary server.

[0005] To ensure data consistency between the primary server and the secondary server, in the existing technology, when the data stored in the primary server is synchronized to the secondary server, the data can be synchronized to a plurality of secondary servers. In other words, a plurality of data copies in the primary server can be stored. When data in one secondary server is lost, the lost data can be obtained from the other secondary servers. As such, when the primary server fails, the plurality of secondary servers can provide an external data read/write service.

[0006] However, because the plurality of data copies in the primary server need to be stored, a relatively large number of resources are needed for data storage.

SUMMARY

[0007] In view of this, implementations of the present application provide a data processing method and device, so as to solve a problem that, a relatively large number of resources are needed for data storage because a plurality of data copies of the same data need to be stored in a distributed data storage system.

[0008] An implementation of the present application provides a data processing method, where the method is applied to a distributed data storage system, the distributed data storage system includes a primary server and at least two secondary servers, and the method includes: receiving a data update request sent by a user, where the data update request includes data updates; sending a log update instruction separately to the primary server and the at least two secondary servers based on the data update request, so that the primary server and the at least two secondary servers perform a log update operation based on the log update instruction; and when it is determined that the log update operation is completed, sending a data storage instruction separately to the primary server and a first secondary server in the at least two secondary servers, so that the primary server and the first secondary server perform, based on the data storage instruction, a storage operation on the data updates.

[0009] An implementation of the present application further provides a data processing method, where the method is applied to a distributed data storage system, the distributed data storage system includes a primary server and at least two secondary servers, and the method includes: receiving, by the primary server, a data update request sent by a user, where the data update request includes data updates; initiating, by the primary server, a log update operation based on the data update request, and sending a log update instruction separately to the at least two secondary servers, so that the at least two secondary servers perform the log update operation based on the log update instruction; and when determining that the log update operation is completed, initiating, by the primary server, a data storage operation, and sending a data storage instruction to a first secondary server in the at least two secondary servers, so that the first secondary server performs, based on the data storage instruction, a storage operation on the data updates.

[0010] An implementation of the present application provides a data processing device, where the device is applied to a distributed data storage system, the distributed data storage system includes a primary server and at least two secondary servers, and the device includes a receiving unit and a sending unit. The receiving unit is configured to receive a data update request sent by a user, where the data update request includes data updates; the sending unit is configured to send a log update instruction separately to the primary server and the at least two secondary servers based on the data update request, so that the primary server and the at least two secondary servers perform a log update operation based on the log update instruction; and when determining that the log update operation is completed, the sending unit is configured to send a data storage instruction separately to the primary server and a first secondary server in the at least two secondary servers, so that the primary server and the first secondary server perform, based on the data storage instruction, a storage operation on the data updates.

[0011] An implementation of the present application further provides a data processing device, where the device is applied to a distributed data storage system, the distributed data storage system includes a primary server and at least two secondary servers, and the device includes a receiving unit and a sending unit. The receiving unit is configured to receive a data update request sent by a user, where the data update request includes data updates; the sending unit is configured to initiate a log update operation based on the data update request, and send a log update instruction to the at least two secondary servers, so that the at least two secondary servers perform the log update operation based on the log update instruction; and when determining that the log update operation is completed, the sending unit is configured to initiate a data storage operation, and send a data storage instruction to a first secondary server in the at least two secondary servers, so that the first secondary server performs, based on the data storage instruction, a storage operation on the data updates.

[0012] At least one of the previously described technical solutions adopted in the implementations of the present application can achieve the following beneficial effects: when receiving the data update request, the distributed data storage system can send a log update instruction separately to the primary server and the secondary servers, so that the primary server and the secondary servers perform a log update operation, and send a data storage instruction to the primary server and one of the secondary servers when it is determined that the log update operation is completed. As

such, when data in the secondary server is lost, the lost data can be restored by using the log prestored in the primary server and the secondary server, so as to ensure data consistency between the primary server and the secondary server, and only the primary server and one of the secondary servers store the data updates, thereby effectively reducing resources needed for data storage.

BRIEF DESCRIPTION OF DRAWINGS

[0013] The accompanying drawings described here are intended to provide a further understanding of the present application, and constitute a part of the present application. The illustrative implementations of the present application and descriptions of the illustrative implementations are intended to describe the present application, and do not constitute limitations on the present application. The accompanying drawings include the following diagrams:

[0014] FIG. 1 is a schematic flowchart illustrating a data processing method, according to an implementation of the present application;

[0015] FIG. 2 is a schematic flowchart illustrating a data processing method, according to an implementation of the present application;

[0016] FIG. 3 is a schematic diagram illustrating a primary server determining procedure, according to an implementation of the present application;

[0017] FIG. 4 is a schematic structural diagram illustrating server troubleshooting, according to an implementation of the present application;

[0018] FIG. 5 is a schematic structural diagram illustrating server troubleshooting, according to an implementation of the present application;

[0019] FIG. 6 is a schematic structural diagram illustrating server troubleshooting, according to an implementation of the present application;

[0020] FIG. 7 is a schematic structural diagram of a server in a local equipment room and a server in a remote equipment room, according to an implementation of the present application;

[0021] FIG. 8 is a schematic structural diagram illustrating a data processing device, according to an implementation of the present application; and

[0022] FIG. 9 is a schematic structural diagram illustrating a data processing device, according to an implementation of the present application.

DESCRIPTION OF IMPLEMENTATIONS

[0023] Usually, in a distributed database storage system, when receiving and storing data updates, a primary server can synchronously store, in a secondary server, the data updates. However, in practice, because the primary server and the secondary server are located in different areas, etc., there is a delay during data synchronization between the primary server and the secondary server. In other words, after the primary server completes storage of the data updates, the secondary server may have not completed storage of the data updates yet. As such, if the primary server fails, the secondary server may lose some data.

[0024] In the existing technology, to ensure that the secondary server does not lose data (to be specific, to ensure data consistency between the primary server and the secondary server), the data in the primary server can be synchronized to a plurality of (at least two) secondary servers. However, compared with the original distributed data storage system, a relatively large number of data copies need to be stored, which easily causes a waste of resources.

[0025] In practice, for the sake of system security, before storing the data updates, the primary server (or the secondary server) first needs to perform log updating, and then the primary server (or the secondary server) then stores the data updates. As such, consistency between the data stored in the primary server (or the secondary server) and the log can be ensured. When the primary server (or the secondary server) fails to store the data, the data can be restored by using the prestored log.

[0026] It can be seen that, in the distributed data storage system, before the primary server synchronizes the data updates, if the primary server and the secondary server first perform log updating and then store the data updates, when the data stored in the secondary server is lost, the lost data can be restored by using the prestored log. As such, log consistency between the primary server and the secondary server ensures data consistency between the primary server and the secondary server, so that the number of copies of the data updates can be reduced when the data is stored.

[0027] Based on the previous idea, in technical solutions provided in the present implementation of the present application, the number of copies of the data can be reduced while the data consistency between the primary server and the secondary server is ensured, thereby reducing the resources needed for data storage.

[0028] To achieve the purpose of the present application, implementations of the

present application provide a data processing method and device. The method is applied to a distributed data storage system, and the distributed data storage system includes a primary server and at least two secondary servers. The method includes: receiving a data update request sent by a user, where the data update request includes data updates; sending a log update instruction separately to the primary server and the at least two secondary servers based on the data update request, so that the primary server and the at least two secondary servers perform a log update operation based on the log update instruction; and when it is determined that the log update operation is completed, sending a data storage instruction separately to the primary server and a first secondary server in the at least two secondary servers, so that the primary server and the first secondary server perform, based on the data storage instruction, a storage operation on the data updates.

[0029] When receiving the data update request, the distributed data storage system can send a log update instruction separately to the primary server and the secondary servers, so that the primary server and the secondary servers perform a log update operation, and send a data storage instruction to the primary server and one of the secondary servers when it is determined that the log update operation is completed. As such, when data in the secondary server is lost, the lost data can be restored by using the log prestored in the primary server and the secondary server, so as to ensure data consistency between the primary server and the secondary server, and only the primary server and one of the secondary servers store the data updates, thereby effectively reducing the resources needed for data storage.

[0030] It is worthwhile to note that the technical solutions provided in the present implementations of the present application can be based on a distributed consistency protocol and applied to a distributed data storage system based on the distributed consistency protocol. The distributed consistency protocol can be the Paxos protocol or other consistency protocols, which is not limited.

[0031] The distributed data storage system based on the distributed consistency protocol can include the primary server and the at least two secondary servers. The primary server can be configured to provide an external data read/write service, and after the primary server fails, the at least two secondary servers can provide the external data read/write service.

[0032] The following clearly and comprehensively describes the technical solutions in the present application with reference to the specific implementations of

the present application and the corresponding accompanying drawings. Apparently, the described implementations are merely some rather than all of the implementations of the present application. Other implementations obtained by a person of ordinary skill in the art based on the implementations of the present application without creative efforts shall fall within the protection scope of the present application.

[0033] The technical solutions provided in the implementations of the present application are described in detail below with reference to the accompanying drawings.

Implementation 1

[0034] FIG. 1 is a schematic flowchart illustrating a data processing method, according to an implementation of the present application. The method is described below. An execution body in the present implementation of the present application can be a distributed data storage system based on a distributed consistency protocol (hereinafter referred to as the distributed data storage system).

[0035] Step 101: Receive a data update request sent by a user.

[0036] The data update request includes data updates.

[0037] In step 101, after initiating a service request and receiving a response message, the user can send the data update request to the distributed data storage system. At this time, the distributed data storage system can receive the data update request sent by the user.

[0038] In the present implementation of the present application, the service request initiated by the user can be a payment request, a friend adding request, or another service request, which is not limited here.

[0039] The user sends the data update request to the distributed data storage system, the data update request can include the data updates, and the data update request is used to request the distributed data storage system to store the data updates.

[0040] For example, the user initiates a payment request on a payment platform, the payment platform can respond to the payment request sent by the user, and after receiving a response message, the user can send the data update request to the distributed data storage system. The data update request includes data that is related to the payment request and that is to be updated, and the data update request is used to request the distributed data storage system to store the data that is related to the

payment request and that is to be updated.

[0041] Step 102: Send a log update instruction separately to the primary server and the at least two secondary servers based on the data update request, so that the primary server and the at least two secondary servers perform a log update operation based on the log update instruction.

[0042] In step 102, when receiving the data update request sent by the user, the distributed data storage system can send the log update instruction separately to the primary server and the at least two secondary servers in the distributed data storage system based on the data request, so that the primary server and the at least two secondary servers perform the log update operation based on the log update instruction.

[0043] It is worthwhile to note that, in practice, before the distributed data storage system receives the data update request, that is, when data stored in the distributed data storage system is empty, the distributed data storage system needs to determine a primary server, and the determined primary server provides the external data read/write service.

[0044] The primary server can be determined by the distributed data storage system based on a random rule, or the primary server can be determined, in an election (the election here can be election held by a plurality of servers based on the distributed consistency protocol) method, from the plurality of servers included in the distributed data storage system, or one server in the distributed data storage system can be selected as the primary server based on an actual case. The method for determining the primary server is not limited here.

[0045] After the primary server is determined, another server in the distributed data storage system can serve as the secondary server. In the present implementation of the present application, there are at least two secondary servers.

[0046] After determining the primary server and the secondary server, the distributed data storage system can send the log update instruction to the primary server when receiving the data update instruction, so that the primary server performs the log update operation based on the log update instruction.

[0047] When sending the log update instruction to the primary server, the distributed data storage system can synchronously send the log update instruction to the at least two secondary servers, so that the at least two secondary servers perform the log update operation based on the log update instruction.

[0048] It is worthwhile to note that, when the log update instruction is sent to the at least two secondary servers, the log update instruction can be sent to each of the at least two secondary servers; or several (at least two) secondary servers can be selected from the at least two secondary servers, and the log update instruction is sent to the selected several secondary servers, which is not limited.

[0049] For example, if the distributed data storage system includes two secondary servers, the log update instruction can be sent separately to the two secondary servers; or if the distributed data storage system includes four secondary servers, the log update instruction can be sent to two of the secondary servers, or the log update instruction can be sent to any three of the secondary servers, or the log update instruction can be further sent separately to the four secondary servers, which is not limited.

[0050] In the present implementation of the present application, to provide example description, the log update instruction can be sent to each of the at least two secondary servers.

[0051] When receiving the log update instruction, the primary server and the at least two secondary servers can perform the log update operation based on the log update instruction.

[0052] Step 103: When it is determined that the log update operation is completed, send a data storage instruction separately to the primary server and a first secondary server in the at least two secondary servers, so that the primary server and the first secondary server perform, based on the data storage instruction, a storage operation on the data updates.

[0053] In step 103, when determining that the log update operation is completed, the distributed data storage system can select one of the at least two secondary servers as the first secondary server, and send the data storage instruction separately to the primary server and the first secondary server, so that the primary server and the first secondary server store the data updates.

[0054] In the present implementation of the present application, at least in the following two cases, it can be determined that the log update operation is completed. The two cases are as follows:

[0055] The first case: when it is determined that the primary server completes the log update operation, it is determined that the log update operation is completed.

[0056] The second case: when it is determined that the primary server and at least

one secondary server in the at least two secondary servers complete the log update operation, it is determined that the log update operation is completed.

[0057] In the first case, in practice, because the primary server is configured to provide the external data read/write service, when determining whether the log update operation is completed, the distributed data storage system needs to first determine whether the primary server completes the log update operation, and can determine that the log update operation is completed when determining that the primary server completes the log update operation.

[0058] In the second case, to ensure log consistency between the primary server and the secondary server, after it is determined that the primary server completes the log update operation, it can be further determined whether the at least two secondary servers complete the log update operation. At this time, if the at least one secondary server in the at least two secondary servers completes the log update operation, it can indicate log consistency between the primary server and the secondary server. In other words, it can be determined that the log update operation is completed.

[0059] As such, after it is determined that the log update operation is completed, one of the at least two secondary servers can be selected as the first secondary server, and the data storage instruction is sent separately to the primary server and the first secondary server, so that the primary server and the first secondary server perform the storage operation on the data updates.

[0060] In the present implementation of the present application, the data storage instruction can be sent separately to the primary server and the first secondary server in the at least two secondary servers based on the previous two cases.

[0061] In the first case, the data storage instruction is sent separately to the primary server and the first secondary server in the at least two secondary servers, which includes: when it is determined that the primary server completes the log update operation, sending the data storage instruction to the primary server, and after the data storage instruction is sent to the primary server and it is determined that the at least one server in the at least two secondary servers completes the log update operation, sending the data storage instruction to the first secondary server in the at least two secondary servers.

[0062] When it is determined that the primary server completes the log update operation, the data storage instruction can be sent to the primary server, and the primary server can receive the data storage instruction, and perform, based on the data

storage instruction, the storage operation on the data updates.

[0063] At this time, it can be further determined whether the at least two secondary servers complete the log update operation. If at least one secondary server completes the log update operation, one of the at least two secondary servers can be selected as the first secondary server, and the data storage instruction is sent to the first secondary server.

[0064] In the second case, the data storage instruction is sent separately to the primary server and the first secondary server in the at least two secondary servers, which includes: when it is determined that the primary server and the at least one secondary server in the at least two secondary servers complete the log update operation, sending the data storage instruction separately to the primary server and the first secondary server in the at least two secondary servers.

[0065] When it is determined that the primary server and at least one secondary server in the at least two secondary servers complete the log update operation, one of the at least two secondary servers can be selected as the first secondary server, and the data storage instruction is sent separately to the primary server and the first secondary server.

[0066] It is worthwhile to note that, in the earlier described two cases, when one of the at least two secondary servers is selected as the first secondary server, one secondary server can be randomly selected as the first secondary server from servers that receive the log update operation instructions, or one secondary server can be selected as the first secondary server based on an actual case, which is not limited.

[0067] After the distributed data storage system sends the data storage instruction separately to the primary server and the first secondary server in the at least two secondary servers, the primary server and the first secondary server can receive the data storage instruction, and perform, based on the data storage instruction, the storage operation on the data updates.

[0068] In practice, some secondary servers in the distributed data storage system that do not perform the storage operation on the data updates can be predetermined. As such, when receiving the data update request, the distributed data storage system cannot send the data storage instruction to these determined secondary servers.

[0069] Optionally, the method further includes: determining a second secondary server from the at least two secondary servers, and when it is determined that the second secondary server completes the log update operation, sending a null-operation

instruction to the second secondary server, where the null-operation instruction is used to instruct the second secondary server to perform no operation.

[0070] In the present implementation of the present application, when being sent, the log update instruction can be sent separately to the at least two secondary servers. However, when it is determined that the log update operation is completed, the data storage instruction is sent to one of the at least two secondary servers. Therefore, the second secondary server can be determined from the at least two secondary servers, and when it is determined that the second secondary server completes the log update operation, the null-operation instruction is sent to the second secondary server, so that the second secondary server performs no operation. In other words, the second secondary server does not store the data updates. As such, the number of copies of the data updates can be reduced.

[0071] It is worthwhile to note that "second" in the second secondary server and "first" in the earlier described first secondary server are merely intended to distinguish different secondary servers, and do not have any other special meaning.

[0072] In the present implementation of the present application, when being determined, the second secondary server can be determined while the primary server is determined, or the second secondary server can be determined while the log update instruction is sent, or the second secondary server can be determined while the data storage instruction is sent, which is not limited.

[0073] As such, the at least two secondary servers can include one first secondary server and at least one second secondary server. For example, if the distributed data storage system includes two secondary servers, one first secondary server and one second secondary server are included; or if the distributed data storage system includes three secondary servers, one first secondary server and two second secondary servers are included.

[0074] When sending the data storage instruction, the distributed data storage system sends the data storage instruction to the primary server and the first secondary server only. As such, the primary server and the first secondary server store the data updates, and the second secondary server does not store the data updates, thereby reducing the resources needed for data storage.

[0075] In addition, in the present implementation of the present application, when any server (the primary server or the secondary server) in the distributed data storage system fails, the distributed data storage system can provide the external data

read/write service while ensuring the data consistency between the primary server and the secondary server.

[0076] In the distributed data storage system, there are at least the following several fault cases:

[0077] The first fault case: the primary server fails.

[0078] The second fault case: the first secondary server fails.

[0079] The third fault case: the second secondary server fails.

[0080] Specific to the previous three fault cases, the following separately describes how the distributed data storage system provides the external data service while ensuring data consistency.

[0081] In the first fault case and the second fault case, the distributed data storage system performs the following operation: when it is determined that the primary server or the first secondary server fails, determining a third secondary server, and sending a data synchronization instruction to the third secondary server, so that the third secondary server completes, based on the data synchronization instruction, data synchronization with a server that does not fail and that stores the data.

[0082] In the first fault case, details are as follows:

[0083] When it is determined that the primary server fails, because the first secondary server stores data, the first secondary server can serve as a new primary server, and the first secondary server provides the external data read/write service. At this time, if a data loss occurs in the first secondary server compared with the primary server, the lost data can be restored by using the log stored in the first secondary server and/or the log stored in the second secondary server.

[0084] In addition, data stored in the first secondary server further needs to be backed up. Therefore, a third secondary server needs to be determined, and the data stored in the first secondary server is backed up.

[0085] When the third secondary server is determined, one secondary server can be selected as the third secondary server from other secondary servers included in the at least two secondary servers, or a new server can be determined as the third secondary server, which is not limited.

[0086] After the third secondary server is determined, a data synchronization instruction can be sent to the third server, so that the third secondary server completes, based on the data synchronization instruction, data synchronization with a server that does not fail and that stores the data. Here, the server that does not fail and that stores

the data can be the first secondary server, and the third secondary server completes the data synchronization with the first secondary server.

[0087] It is worthwhile to note that "third" in the third secondary server is intended for distinguishing from the first secondary server and the second secondary server, and has no other special meaning.

[0088] Optionally, to ensure log consistency among the first secondary server, the second secondary server, and the third secondary server based on the distributed consistency protocol, a log synchronization instruction can be further sent to the third secondary server, so that the third secondary server performs a log synchronization operation based on the log stored in the first secondary server.

[0089] In the second fault case, details are as follows:

[0090] When it is determined that the first secondary server fails, the primary server can still provide the external data service. However, because the first secondary server fails, a new secondary server needs to be determined, and data stored in the primary server is backed up.

[0091] The method for determining the new secondary server is the same as the earlier described method for determining the third secondary server. Details are omitted here for simplicity.

[0092] After the new secondary server is determined, a data synchronization instruction can be sent to the new secondary server, so that the new secondary server completes, based on the data synchronization instruction, data synchronization with a server that does not fail and that stores the data. Here, the server that does not fail and that stores the data can be the primary server, and the new secondary server completes the data synchronization with the primary server.

[0093] Optionally, to ensure log consistency among the primary server, the second secondary server, and the new secondary server based on the distributed consistency protocol, a log synchronization instruction can be further sent to the new secondary server, so that the new secondary server performs a log synchronization operation based on the log stored in the primary server.

[0094] In the third fault case, when it is determined that the second secondary server fails, determining a fourth secondary server, and sending a log synchronization instruction to the fourth secondary server, so that the fourth secondary server completes, based on the log synchronization instruction, log synchronization with a server that does not fail.

[0095] When it is determined that the second secondary server fails, the primary server can still provide the external data read/write service, and the first secondary server can back up the data stored in the primary server. However, at this time, to ensure log consistency between the primary server and the first secondary server based on the distributed consistency protocol, a fourth secondary server needs to be determined.

[0096] The method for determining the fourth secondary server is the same as the earlier described method for determining the third secondary server. Details are omitted here for simplicity.

[0097] After the fourth secondary server is determined, a log synchronization instruction can be sent to the fourth secondary server, so that after receiving the log synchronization instruction, the fourth secondary server completes, based on the log synchronization instruction, log synchronization with the server that does not fail.

[0098] Herein, the server that does not fail can be another server other than the second secondary server, or can be the primary server, the first secondary server, or another server that completes the log update operation, which is not limited.

[0099] "Fourth" in the fourth secondary server is intended for distinguishing from the first secondary server, the second secondary server, and the third secondary server, and has no other special meaning.

[0100] It is worthwhile to note that, because the second secondary server stores no data, after the fourth secondary server is determined, no data synchronization instruction needs to be sent to the fourth secondary server.

[0101] In the technical solutions provided in the present implementation of the present application, when the data update request is received, the log update instruction can be sent separately to the primary server and the secondary servers, so that the primary server and the secondary servers perform the log update operation, and the data storage instruction is sent to the primary server and one of the secondary servers when it is determined that the log update operation is completed. As such, when data in the secondary server is lost, the lost data can be restored by using the log prestored in the primary server and the secondary server, so as to ensure data consistency between the primary server and the secondary server, and only the primary server and one of the secondary servers store the data updates, thereby effectively reducing the resources needed for data storage.

Implementation 2

[0102] FIG. 2 is a schematic flowchart illustrating a data processing method, according to an implementation of the present application. The method is described below. An execution body of the present implementation of the present application can be a primary server in a distributed data storage system based on a consistency protocol.

[0103] Step 201: The primary server receives a data update request sent by a user.

[0104] The data update request includes data updates.

[0105] In step 201, after initiating a service request and receiving a response message, the user can send the data update request to the primary server in the distributed data storage system. At this time, the primary server in the distributed data storage system can receive the data update request sent by the user.

[0106] In the present implementation of the present application, the primary server is determined when the data stored in the distributed data storage system is empty. The method for determining the primary server is the same as the method described in the previous Implementation 1. Details are omitted here for simplicity.

[0107] Step 202: The primary server initiates a log update operation based on the data update request, and sends a log update instruction to the at least two secondary servers, so that the at least two secondary servers perform the log update operation based on the log update instruction.

[0108] In step 202, when receiving the data update request sent by the user, the primary server can initiate the log update operation, and send the log update instruction to the at least two secondary servers in the distributed data storage system based on the data request, so that the at least two secondary servers perform the log update operation based on the log update instruction.

[0109] When receiving the data update request, the primary server can initiate the log update operation based on the data update request, and further send the log update instruction to the at least two secondary servers based on the data update request; or when receiving the data update request, the primary server can send the log update instruction to the at least two secondary servers based on the data update request, and further initiate the log update operation based on the data update request, which is not limited.

[0110] After the primary server sends the log update instruction to the at least two

secondary servers, the at least two secondary servers can receive the log update instruction, and perform the log update operation based on the log update instruction.

[0111] Step 203: When determining that the log update operation is completed, the primary server initiates a data storage operation, and sends a data storage instruction to a first secondary server in the at least two secondary servers, so that the first secondary server performs, based on the data storage instruction, a storage operation on the data updates.

[0112] In step 203, when determining that the log update operation is completed, the primary server can initiate the data storage operation, select one of the at least two secondary servers as a first secondary server, and send the data storage instruction to the first secondary server, so that the primary server and the first secondary server store the data updates.

[0113] The primary server can determine, at least in the following two cases, that the log update operation is completed. The two cases are as follows:

[0114] The first case: when it is determined that the primary server completes the log update operation, it is determined that the log update operation is completed.

[0115] The second case: when it is determined that the primary server and at least one secondary server in the at least two secondary servers complete the log update operation, it is determined that the log update operation is completed.

[0116] After determining that the log update operation is completed, the primary server can initiate the data storage operation, so that the primary server provides an external data read/write service. At this time, the primary server can further select one of the at least two secondary servers as a first secondary server, and send the data storage instruction to the first secondary server, so that the first secondary server performs, based on the data storage instruction, the data storage operation on the data updates.

[0117] The primary server can initiate the data storage operation before sending the data storage instruction, or can initiate the data storage operation after sending the data storage instruction, which is not limited.

[0118] As such, after receiving the data update request, the primary server can send the log update instruction to the first secondary server and the second secondary server, and when determining that the log update operation is completed, the primary server performs the data storage operation, and sends the data storage instruction to the first secondary server. As such, when data in the first secondary server is lost, the

lost data can be restored based on the log prestored in the primary server, and/or the log prestored in the first secondary server, and/or the log prestored in the second secondary server, so as to ensure data consistency between the primary server and the first secondary server. In addition, only the primary server and the first secondary server store the data updates, thereby reducing the resources needed for data storage.

Implementation 3

[0119] A distributed data storage system includes three servers (one primary server and two secondary servers) is used as example description of technical solutions provided in the present implementation of the present application below. An execution body of the present implementation of the present application can be a primary server.

[0120] Step 1: Determine the primary server, a first secondary server, and a second secondary server from the distributed data storage system.

[0121] When data stored in the distributed data storage system is empty, the primary server and the two secondary servers can be determined from the distributed data storage system. For ease of distinguishing, the two secondary servers can be respectively referred to as the first secondary server and the second secondary server.

[0122] In the present implementation of the present application, the primary server can store the data updates, and provide an external data read/write service. The first secondary server can store the data updates, and the second secondary server does not store the data updates.

[0123] In the present implementation of the present application, a sequence of determining the primary server, the first secondary server, and the second secondary server is not limited.

[0124] Example description of a determining sequence of the second secondary server, the primary server, and the first secondary server is used below.

[0125] First, the second secondary server is determined.

[0126] When being determined, the second secondary server can be determined based on a distributed consistency protocol, namely, the Paxos protocol, in a method of election held by the three servers included in the distributed data storage system.

[0127] When the second secondary server is determined, one server can be randomly selected as the second secondary server or the second secondary server is

determined based on an actual case. Here, a method for determining the second secondary server is not limited.

[0128] Second, the primary server is determined.

[0129] After the second secondary server is determined, the primary server can be determined based on the distributed consistency protocol, namely, the Paxos protocol, in a method of election held by the second secondary server and remaining two servers in the distributed data storage system.

[0130] It is worthwhile to note that, when the primary server is determined in the method of election, neither the second secondary server nor the other two servers can elect the second secondary server as the primary server.

[0131] Referring to FIG. 3, FIG. 3 is a schematic diagram illustrating a primary server determining procedure, according to an implementation of the present application.

[0132] It can be seen from FIG. 3 that after the second secondary server C is determined in the method of election, server A, server B, and the second secondary server C can determine the primary server in the method of election. Server A can elect server A or server B as the primary server, but cannot elect the second secondary server C as the primary server. Server B can elect server B or server A as the primary server, but cannot elect the second secondary server C as the primary server. The second secondary server C cannot elect the second secondary server C as the primary server, but can elect server A or server B as the primary server.

[0133] When determining the primary server, one of the two servers other than the second secondary server can be randomly selected as the primary server, or one server can be selected as the primary server based on an actual case, which is not limited.

[0134] Finally, the first secondary server is determined.

[0135] After the second secondary server and the primary server are determined from the three servers included in the distributed data storage system, the remaining server can serve as the first secondary server.

[0136] Step 2: The primary server receives a data update request sent by a user.

[0137] The data update request includes data updates.

[0138] Step 3: The primary server sends a log update instruction to the first secondary server and the second secondary server, so that the first secondary server and the second secondary server perform a log update operation based on the log update instruction.

[0139] After the primary server sends the log update instruction to the first secondary server and the second secondary server, the first secondary server and the second secondary server can receive the log update instruction. At this time, the primary server, the first secondary server, and the second secondary server can perform log updating based on the distributed consistency protocol, namely, the Paxos protocol. When it is determined that the primary server and at least one of the first secondary server and the second secondary server complete the log update operation, it is determined that the log update operation is completed.

[0140] Step 4: The primary server sends a data storage instruction to the first secondary server, so that the first secondary server performs, based on the data storage instruction, a storage operation on the data updates.

[0141] When determining that the log update operation is completed, the primary server can perform the storage operation on the data updates, and send the data storage instruction to the first secondary server. The first secondary server can receive the data storage instruction. At this time, if the first secondary server has completed the log update operation, the first secondary server can perform, based on the data storage instruction, the storage operation on the data updates; or if the first secondary server has not completed the log update operation, the first secondary server needs to continue performing the log update operation, and when completing the log update operation, the first secondary server can perform, based on the received data storage instruction, the storage operation on the data updates.

[0142] As such, when receiving the data update request, the primary server can send the log update instruction to the first secondary server and the second secondary server, and when determining that the log update operation is completed, the primary server performs the data storage operation, and sends the data storage instruction to the first secondary server. As such, only the primary server and the first secondary server store the data updates, thereby reducing the resources needed for data storage.

[0143] In the present implementation of the present application, when any one of the primary server, the first secondary server, and the second secondary server fails, the distributed data storage system can provide an external data read/write service while ensuring data consistency.

[0144] When the primary server fails, details are as follows:

[0145] As shown in FIG. 4, when the primary server fails, the primary server cannot continue providing the read/write service for a user, and at this time, the first

secondary server B can be determined as a new primary server, and server D is determined as the new first secondary server. The first secondary server B can provide the data read/write service for the user, and maintain log synchronization with the second secondary server C. Server D obtains a data copy through duplication from the first secondary server B to complete data synchronization with the first secondary server B, and server D performs the log synchronization operation to complete log synchronization with the first secondary server B.

[0146] When the first secondary server fails, details are as follows:

[0147] As shown in FIG. 5, when the first secondary server fails, primary server A can continue providing the data read/write service for the user. At this time, server E can be determined as the new first secondary server. Server E obtains a data copy through duplication from primary server A to complete data synchronization with primary server A, and server E performs the log synchronization operation to complete log synchronization with primary server A.

[0148] When the second secondary server fails, details are as follows:

[0149] As shown in FIG. 6, when the second secondary server fails, primary server A can continue providing the data read/write service for the user. At this time, server F can be determined as the new second secondary server, and server F performs the log synchronization operation to complete log synchronization with primary server A.

[0150] In practice, there can be remote distributed data storage system. In other words, for the same data, in addition to local storage of a data copy, remote storage of data copies is needed, so as to enhance stability of the distributed data storage system. For the remote distributed data storage system, to reduce the resources needed for data storage, at least three servers can be used locally, the primary server stores the data updates, and the other secondary servers store only the log (that is, the other secondary servers perform only the log update operation); and at least three servers are used remotely, the primary server stores a data copy of data in the local primary server, and the other secondary servers store only the log (that is, the other secondary servers perform only the log update operation).

[0151] As shown in FIG. 7, in a local equipment room, three servers labeled A, B, and C can be used, server A is the primary server and stores the data updates, and server B and server C are the secondary servers and perform only the log update operation, but do not store the data updates. In a remote equipment room, servers D, E,

and F can be used, server D is the primary server and stores a data copy of data stored in server A in the local equipment room, and server E and server F perform only the log update operation, but do not store the data updates.

[0152] When server A fails, server D can provide the external data read/write service.

[0153] When server D fails, server A can still continue providing the external data read/write service. At this time, a new primary server needs to be determined in the remote equipment room. The new primary server can complete data synchronization and log synchronization with server A.

[0154] When any one of server B, server C, server E, and server F fails, server A can still continue providing the external read/write service.

[0155] FIG. 8 is a schematic structural diagram illustrating a data processing device, according to an implementation of the present application. The device is applied to the distributed data storage system, the distributed data storage system includes a primary server and at least two secondary servers, and the device includes a receiving unit 81, a sending unit 82, and a determining unit 83.

[0156] The receiving unit 81 is configured to receive a data update request sent by a user, where the data update request includes data updates.

[0157] The sending unit 82 is configured to send a log update instruction separately to the primary server and the at least two secondary servers based on the data update request, so that the primary server and the at least two secondary servers perform a log update operation based on the log update instruction.

[0158] When determining that the log update operation is completed, the sending unit 82 is configured to send a data storage instruction separately to the primary server and a first secondary server in the at least two secondary servers, so that the primary server and the first secondary server perform, based on the data storage instruction, a storage operation on the data updates.

[0159] The sending unit 82 determines that the log update operation is completed, which includes: when it is determined that the primary server completes the log update operation, determining that the log update operation is completed; or when it is determined that the primary server and at least one secondary server in the at least two secondary servers complete the log update operation, determining that the log update operation is completed.

[0160] The sending unit 82 sends the data storage instruction separately to the

primary server and the first secondary server in the at least two secondary servers, which includes: when it is determined that the primary server completes the log update operation, sending the data storage instruction to the primary server, and after the data storage instruction is sent to the primary server and it is determined that the at least one server in the at least two secondary servers completes the log update operation, sending the data storage instruction to the first secondary server in the at least two secondary servers; or when it is determined that the primary server and the at least one secondary server in the at least two secondary servers complete the log update operation, sending the data storage instruction separately to the primary server and the first secondary server in the at least two secondary servers.

[0161] Optionally, the data processing device further includes the determining unit 83.

[0162] The determining unit 83 is configured to determine a second secondary server from the at least two secondary servers, and send a null-operation instruction to the second secondary server when determining that the second secondary server completes the log update operation, where the null-operation instruction is used to instruct the second secondary server to perform no operation.

[0163] Optionally, when determining that the primary server or the first secondary server fails, the determining unit 83 is configured to determine a third secondary server, and send a data synchronization instruction to the third secondary server, so that the third secondary server completes, based on the data synchronization instruction, data synchronization with a server that is not faulty and that stores the data.

[0164] Optionally, when determining that the second secondary server fails, the determining unit 83 is configured to determine a fourth secondary server, and send a log synchronization instruction to the fourth secondary server, so that the fourth secondary server completes, based on the log synchronization instruction, log synchronization with the server that is not faulty.

[0165] FIG. 9 is a schematic structural diagram illustrating a data processing device, according to an implementation of the present application. The device is applied to the distributed data storage system, the distributed data storage system includes a primary server and at least two secondary servers, and the device includes a receiving unit 91 and a sending unit 92.

[0166] The receiving unit 91 is configured to receive a data update request sent by

a user, where the data update request includes data updates.

[0167] The sending unit 92 is configured to initiate a log update operation based on the data update request, and send a log update instruction to the at least two secondary servers, so that the at least two secondary servers perform the log update operation based on the log update instruction.

[0168] When determining that the log update operation is completed, the sending unit 92 is configured to initiate a data storage operation, and send a data storage instruction to a first secondary server in the at least two secondary servers, so that the first secondary server performs, based on the data storage instruction, a storage operation on the data updates.

[0169] A person skilled in the art should understand that the implementations of the present application can be provided as a method, a system, or a computer program product. Therefore, the present application can use hardware only implementations, software only implementations, or implementations with a combination of software and hardware. In addition, the present application can use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a magnetic disk memory, a CD-ROM, an optical memory, etc.) that include computer-usable program code.

[0170] The present application is described with reference to the flowcharts and/or block diagrams of the method, the device (system), and the computer program product according to the implementations of the present application. It should be understood that computer program instructions can be used to implement each procedure and/or each block in the flowcharts and/or the block diagrams and a combination of a procedure and/or a block in the flowcharts and/or the block diagrams. These computer program instructions can be provided for a general-purpose computer, a dedicated computer, a built-in processor, or a processor of another programmable data processing device to generate a machine, so that the instructions executed by the computer or the processor of another programmable data processing device are used to generate an apparatus for implementing a function specified in one or more procedures in the flowcharts and/or in one or more blocks in the block diagrams.

[0171] These computer program instructions can be stored in a computer readable memory that can instruct the computer or the another programmable data processing device to work in a specific way, so that the instructions stored in the computer

readable memory are used to generate an artifact that includes an instruction apparatus. The instruction apparatus implements a function specified in one or more procedures in the flowcharts and/or in one or more blocks in the block diagrams.

[0172] These computer program instructions can be loaded onto the computer or another programmable data processing device, so that a series of operations and steps are performed on the computer or another programmable device, thereby generating computer-implemented processing. Therefore, the instructions executed on the computer or another programmable device provide steps for implementing a function specified in one or more procedures in the flowcharts and/or in one or more blocks in the block diagrams.

[0173] In a typical configuration, a computing device includes one or more processors (CPUs), an input/output interface, a network interface, and a memory.

[0174] The memory may include a non-persistent memory, a RAM, a non-volatile memory, and/or another form in a computer readable medium, for example, a ROM or a flash memory (flash RAM). The memory is an example of the computer readable medium.

[0175] The computer readable medium includes persistent, non-persistent, movable, and unmovable media that can implement information storage by using any method or technology. Information can be a computer readable instruction, a data structure, a program module, or other data. A computer storage medium includes but is not limited to a phase-change random access memory (PRAM), a static RAM (SRAM), a dynamic RAM (DRAM), a random access memory (RAM) of another type, a read-only memory (ROM), an electrically erasable programmable read-only memory (EEPROM), a flash memory or another memory storage device, a compact disc read-only memory (CD-ROM), a digital versatile disc (DVD), or another optical memory, a cassette, a magnetic disk, or another magnetic storage device or any other non-transmission medium. The computer storage medium can be configured to store information accessible to a computing device. Based on the description in the present specification, the computer readable medium does not include computer readable transitory media such as a modulated data signal and a carrier.

[0176] It is further worthwhile to note that, the term "include", "comprise", or any other variant is intended to cover non-exclusive inclusion, so that a process, a method, a commodity, or a device that includes a series of elements not only includes these elements, but also includes other elements that are not expressly listed, or further

includes elements inherent to such process, method, commodity, or device. An element preceded by "includes a ..." does not, without more constraints, exclude the existence of additional identical elements in the process, method, commodity, or device that includes the element.

[0177] A person skilled in the art should understand that the implementations of the present application can be provided as a method, a system, or a computer program product. Therefore, the present application can use hardware only implementations, software only implementations, or implementations with a combination of software and hardware. In addition, the present application can use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a magnetic disk memory, a CD-ROM, an optical memory, etc.) that include computer-usable program code.

[0178] The previous descriptions are merely implementations of the present application, and are not intended to limit the present application. A person skilled in the art can make various modifications and changes to the present application. Any modifications, equivalent replacements, improvements, etc. made within the spirit and principle of the present application shall fall within the protection scope of the claims of the present application.

2017282817 25 Nov 2019

CLAIMS

1. A method for processing data, the method comprising:

receiving a data update request sent by a user, wherein the data update request comprises data updates (101);

sending a log update instruction separately to a primary server and at least two secondary servers based on the data update request, wherein the primary server and the at least two secondary servers perform a log update operation based on the log update instruction (102);

determining whether the log update operation is completed;

in response to determining that the log update operation is completed, sending a data storage instruction separately to the primary server and a first secondary server in the at least two secondary servers, so that the primary server and the first secondary server perform, based on the data storage instruction, a storage operation on the data updates (103);

determining a second secondary server from the at least two secondary servers;

determining that the second secondary server completes the log update operation; and

in response to determining that the second secondary server completes the log update operation, sending a null-operation instruction to the second secondary server, wherein the null-operation instruction is used to instruct the second secondary server to perform no operation.

2. The method according to claim 1, wherein determining whether the log update operation is completed comprises:

determining whether the primary server completed the log update operation; or

determining whether the primary server and at least one secondary server in the at least two secondary servers completed the log update operation.

3. The method according to claim 2, wherein sending the data storage instruction separately to the primary server and the first secondary server in the at least two secondary servers comprises:

in response to determining that the primary server completed the log update operation, sending the data storage instruction to the primary server, and after the data storage instruction is sent to the primary server, determining whether the at least one secondary server in the at least two secondary servers completed the log update operation, sending the data storage instruction to the first secondary server in the at least two secondary servers; or

in response to determining that the primary server and the at least one secondary server in

the at least two secondary servers completed the log update operation, sending the data storage instruction separately to the primary server and the first secondary server in the at least two secondary servers.

4. The method according to claim 1 or 3, wherein the second secondary server is configured to, upon receiving the null-operation instruction, not perform a storage operation on the data updates.

5. The method according to claim 4, wherein the method further comprises:

in response to determining that the primary server or the first secondary server failed, determining a third secondary server, and sending a data synchronization instruction to the third secondary server to complete, based on the data synchronization instruction, data synchronization with a server that is not faulty and that stores the data.

6. The method according to claim 4, wherein the method further comprises:

in response to determining that the second secondary server failed, determining a fourth secondary server, and sending a log synchronization instruction to the fourth secondary server to complete, based on the log synchronization instruction, a log synchronization with a server that is not failed.

7. The method according to any one of claims 1 to 6, wherein the primary server is in communication with an external computing device to perform external read or write operations associated with the data updates.

8. The method according to any one of claims 1 to 7, wherein the data update request comprises a payment request.

9. The method according to any one of claims 1 to 8, wherein the primary server is determined based on a random rule or based on a distributed consistency protocol.

10. A device for processing data, the device comprising a plurality of modules configured to perform the method of any one of claims 1 to 9.

11 Jan 2019 20172822817

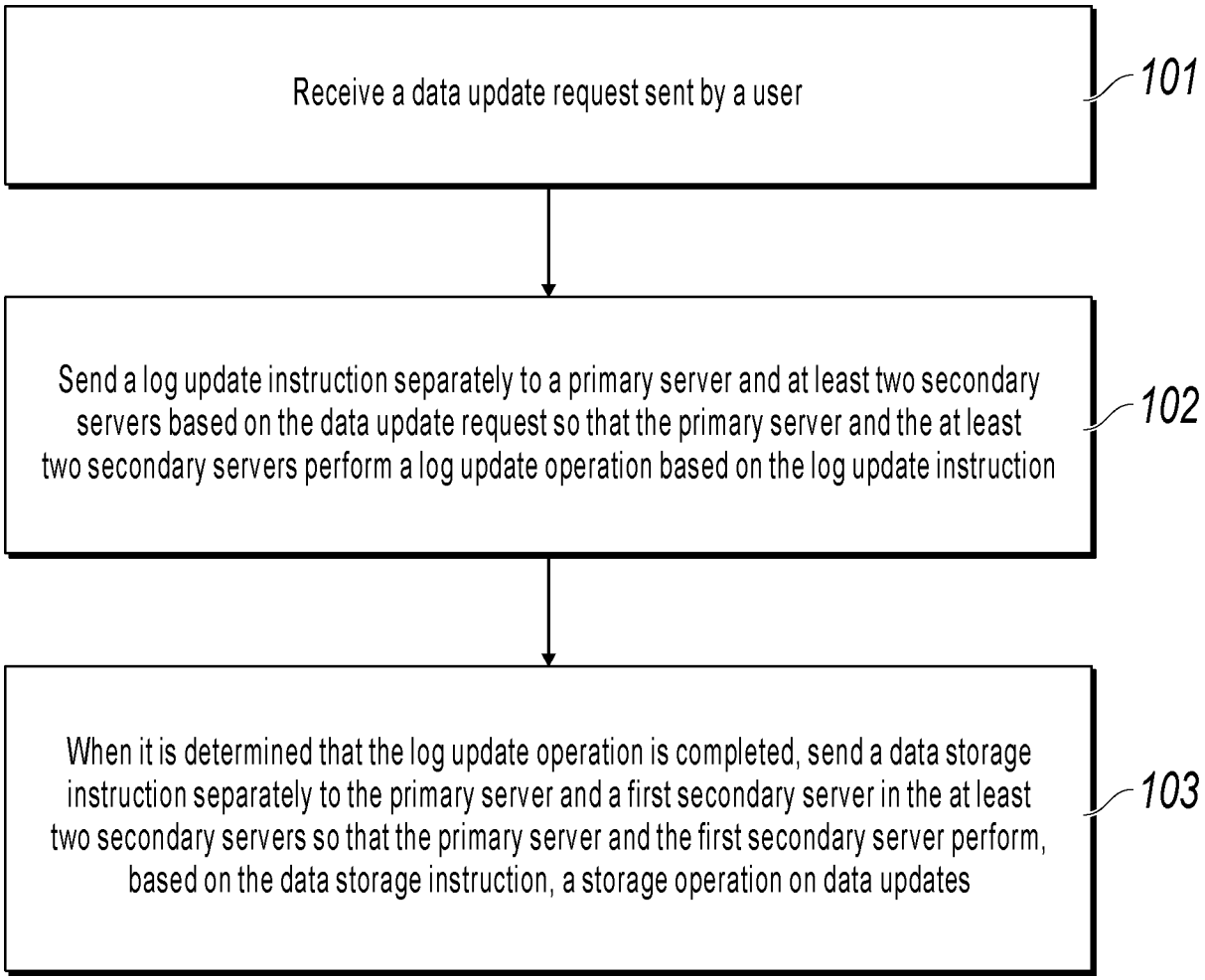

FIG. 1

11 Jan 2019 2017282817

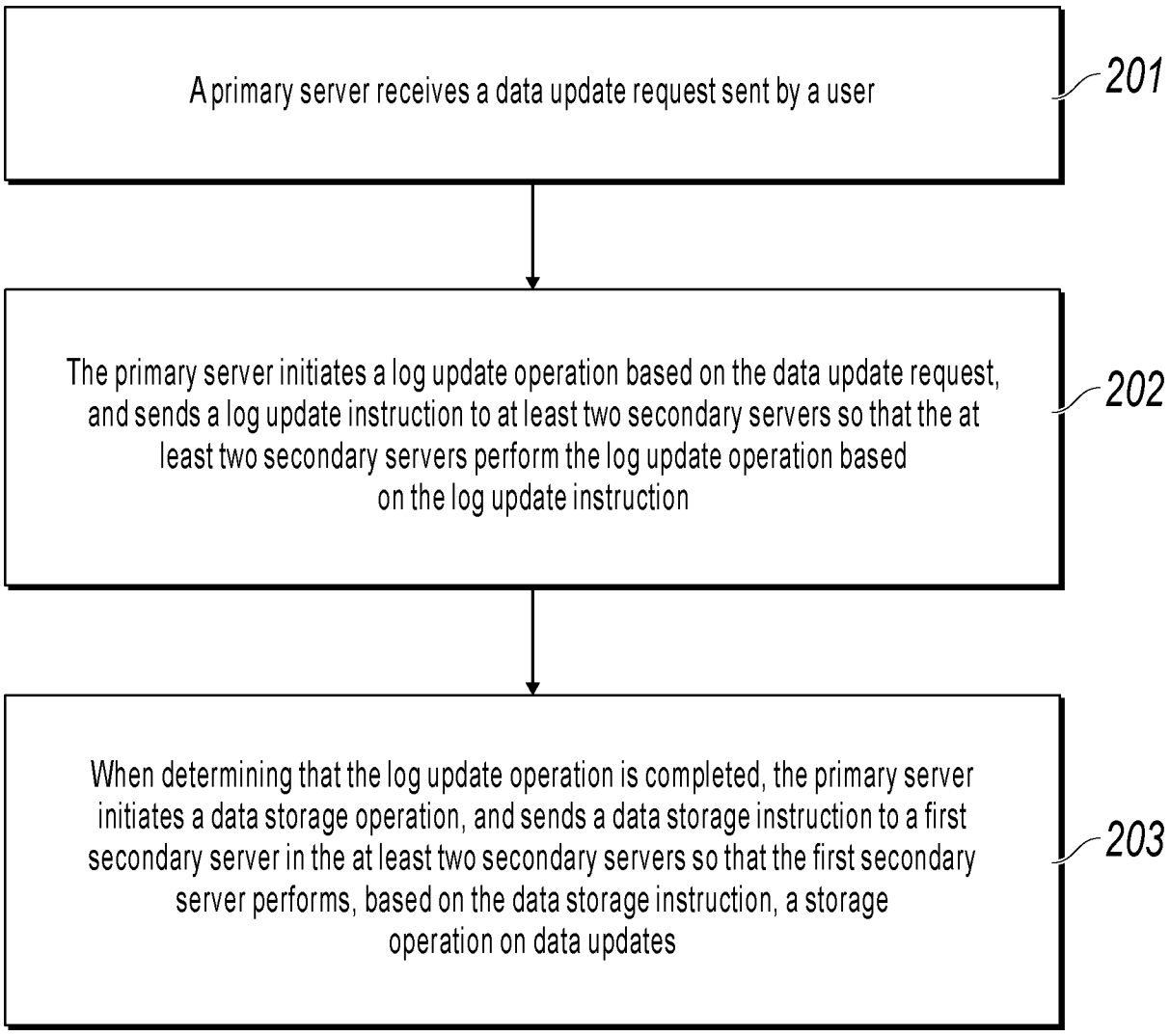

FIG. 2

2017282817 11 Jan 2019

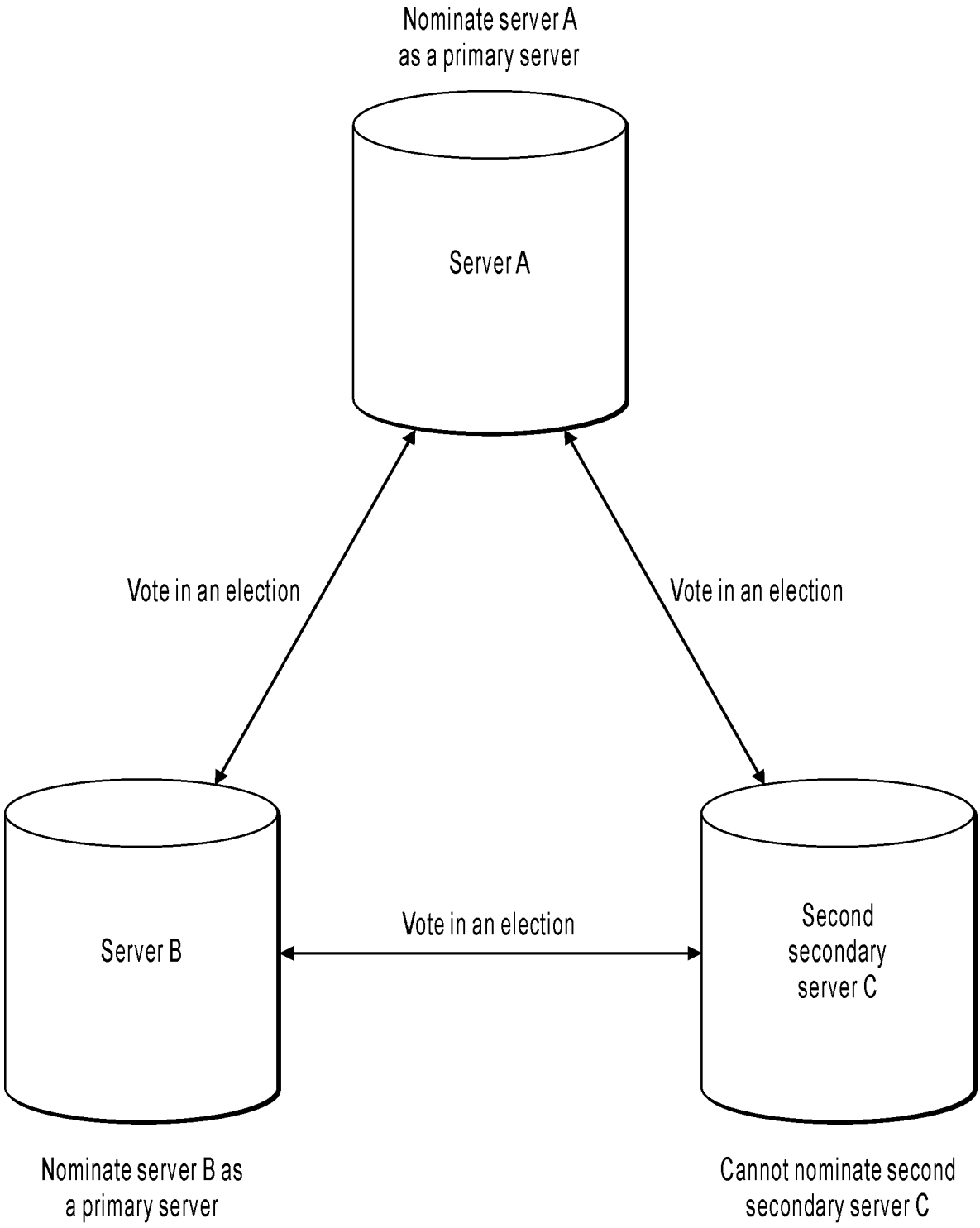

FIG. 3

11 Jan 2019 2017282817

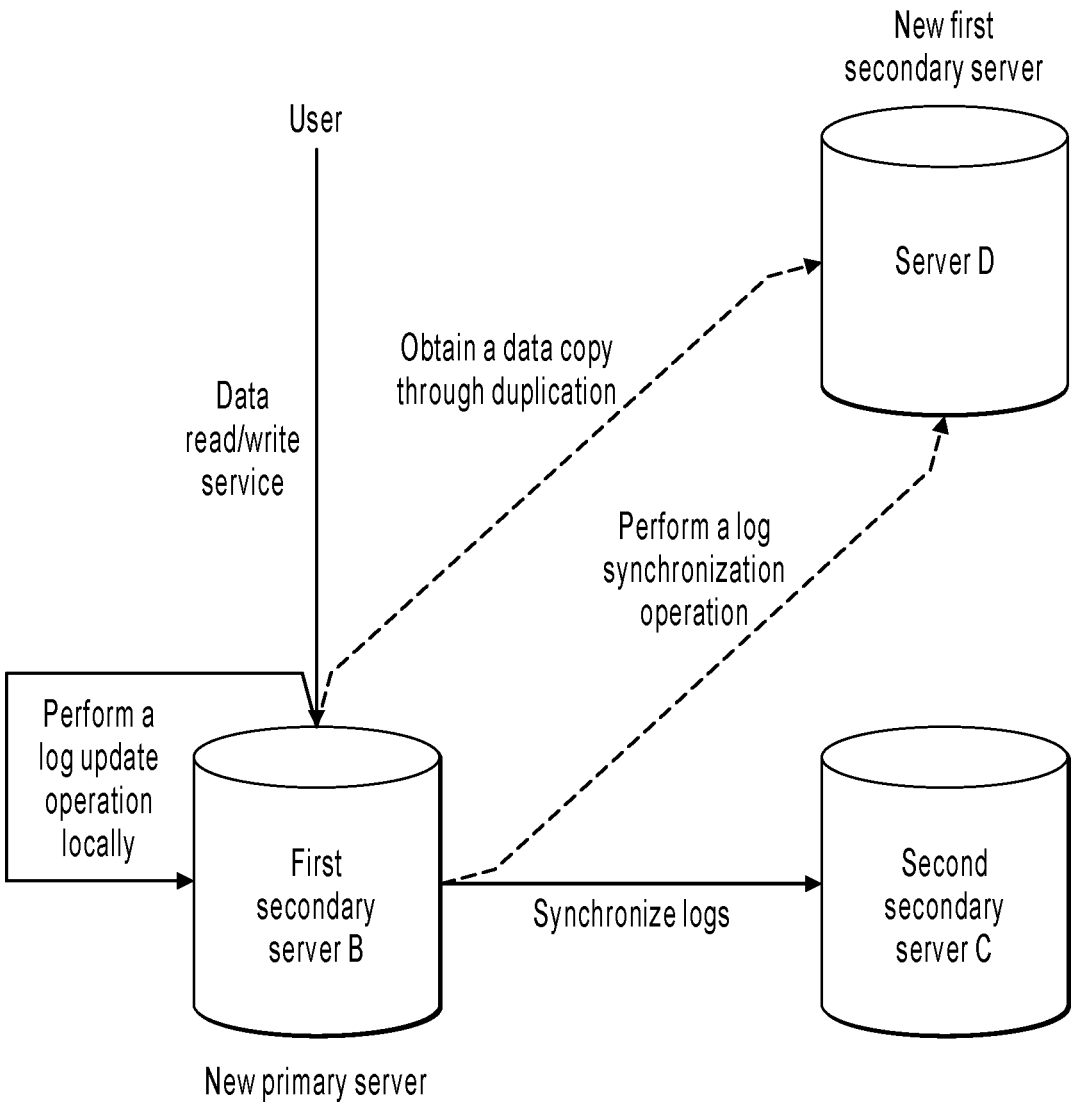

FIG. 4

2017282817 11 Jan 2019

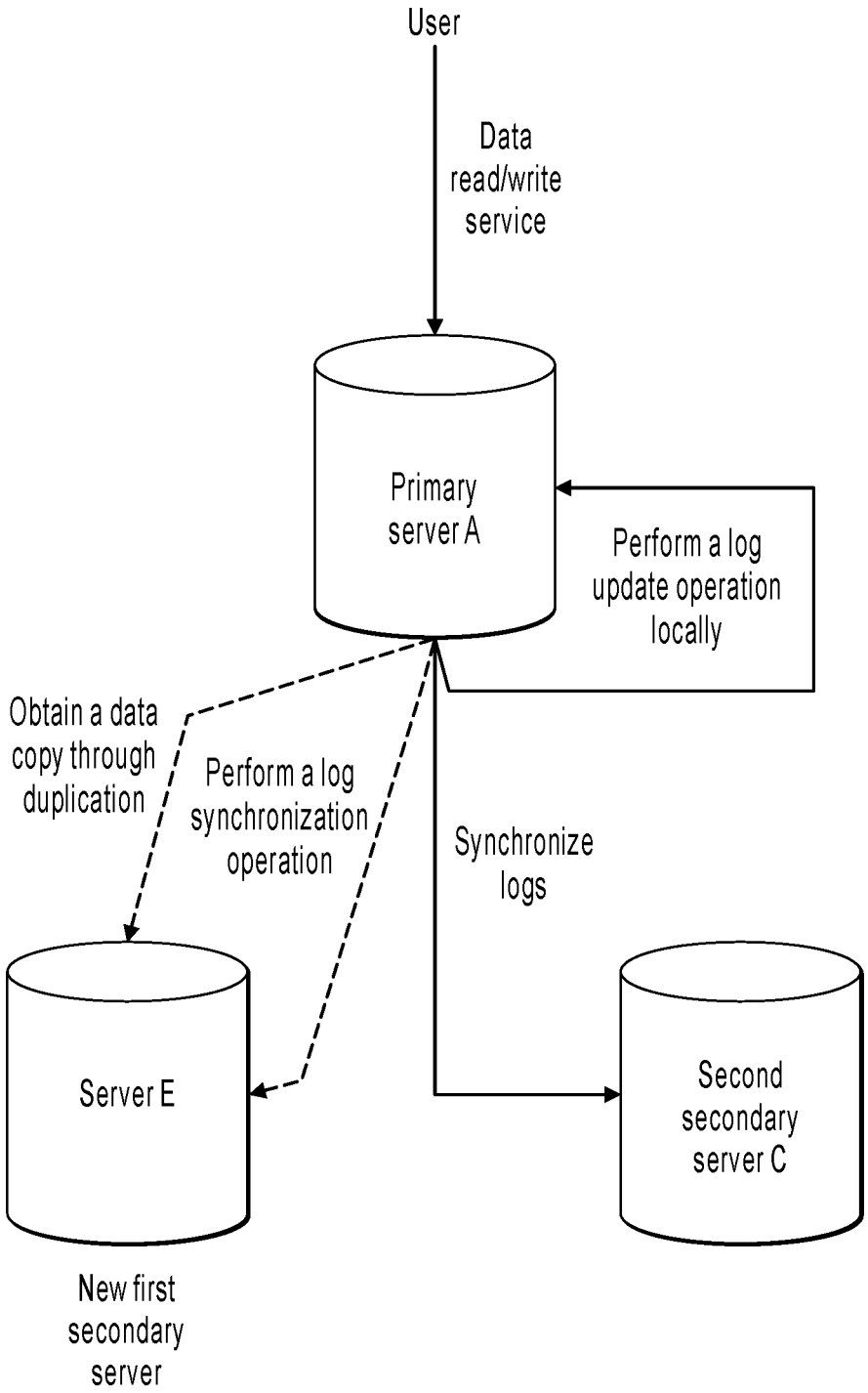

FIG. 5

11 Jan 2019 20172822817

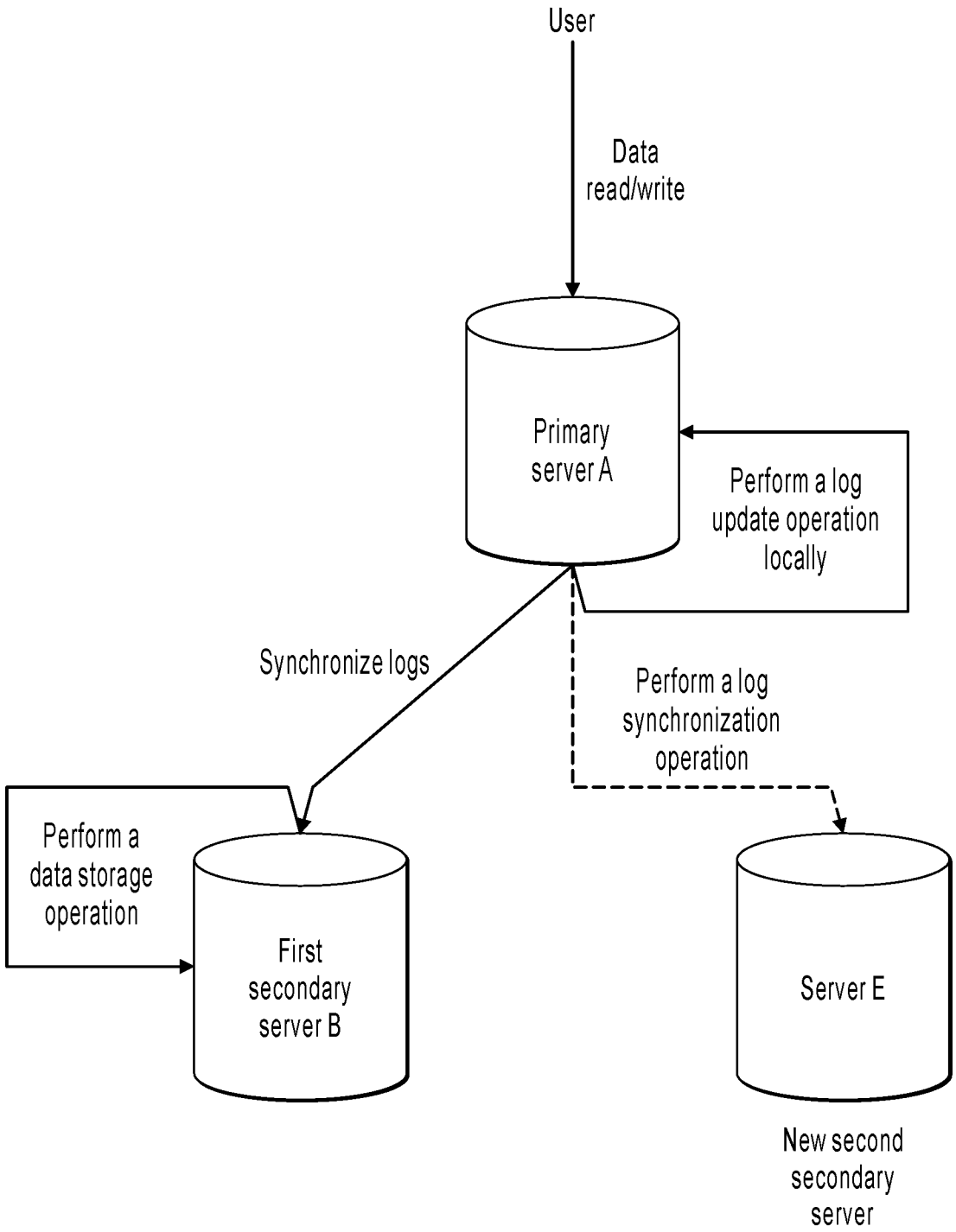

FIG. 6

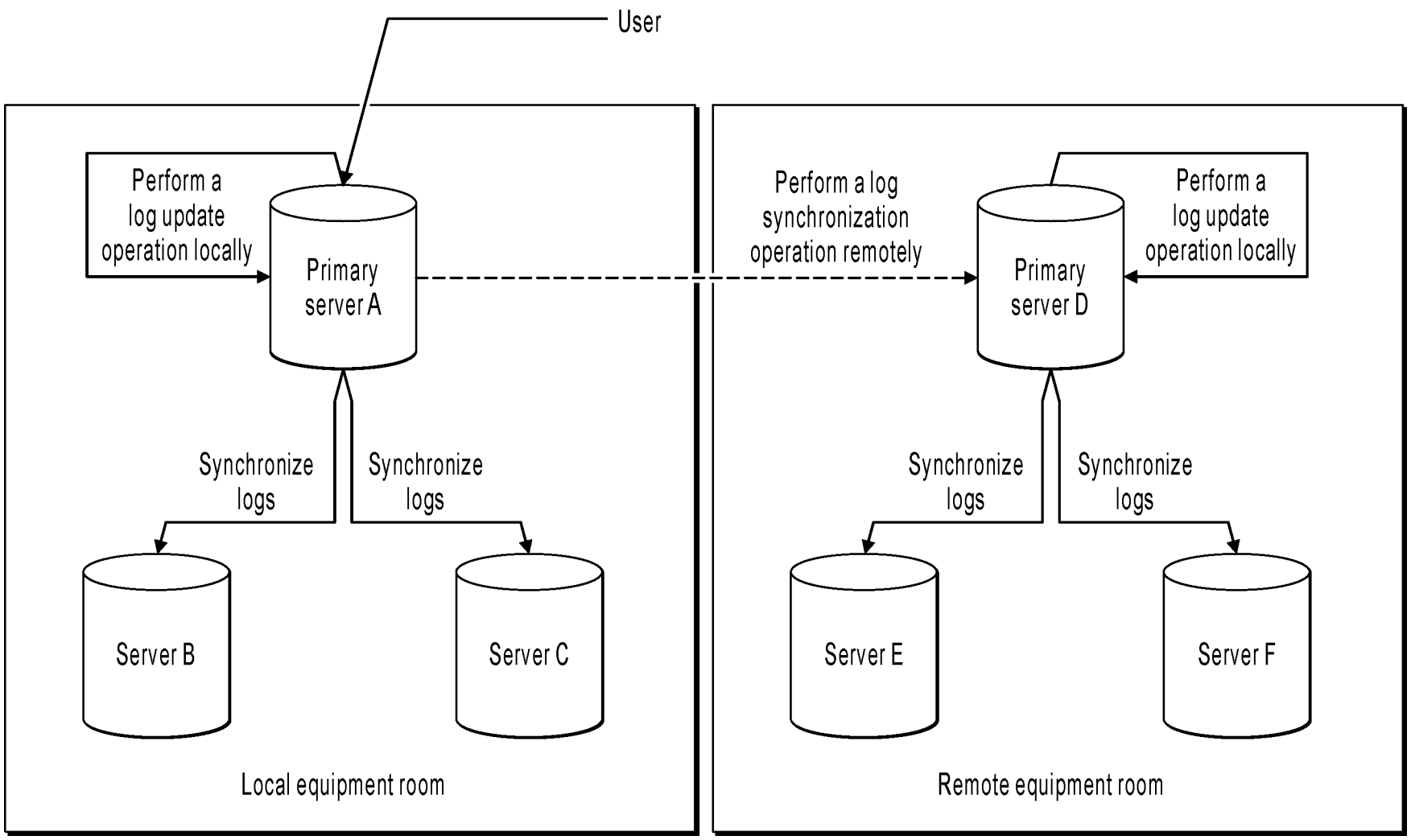
User
Perform a log update operation locally
Primary server A
Synchronize logs
Synchronize logs
Server B
Server C
Local equipment room
Perform a log synchronization operation remotely
Primary server D
Perform a log update operation locally
Synchronize logs
Synchronize logs
Server E
Server F
Remote equipment room

FIG. 7

11 Jan 2019 2017282817

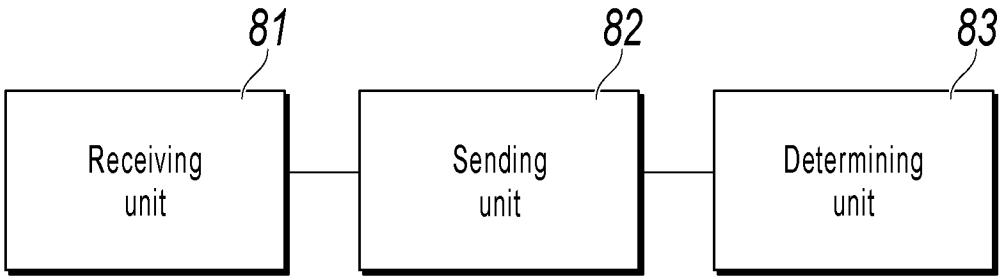

FIG. 8

2017282817 11 Jan 2019

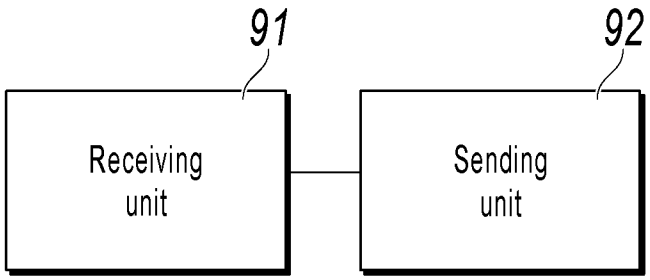

FIG. 9