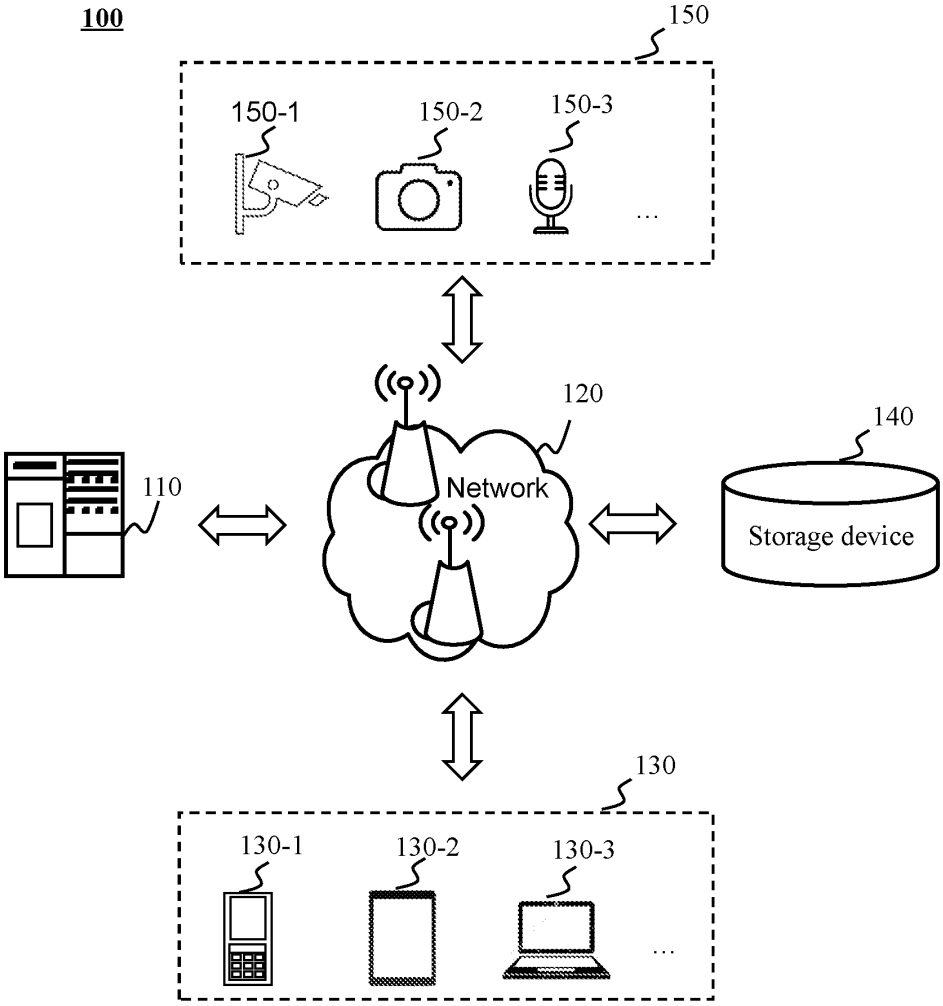

100
150
150-1
150-2
150-3
...
120
140
110
Network
Storage device
130
130-1
130-2
130-3
...

FIG. 1

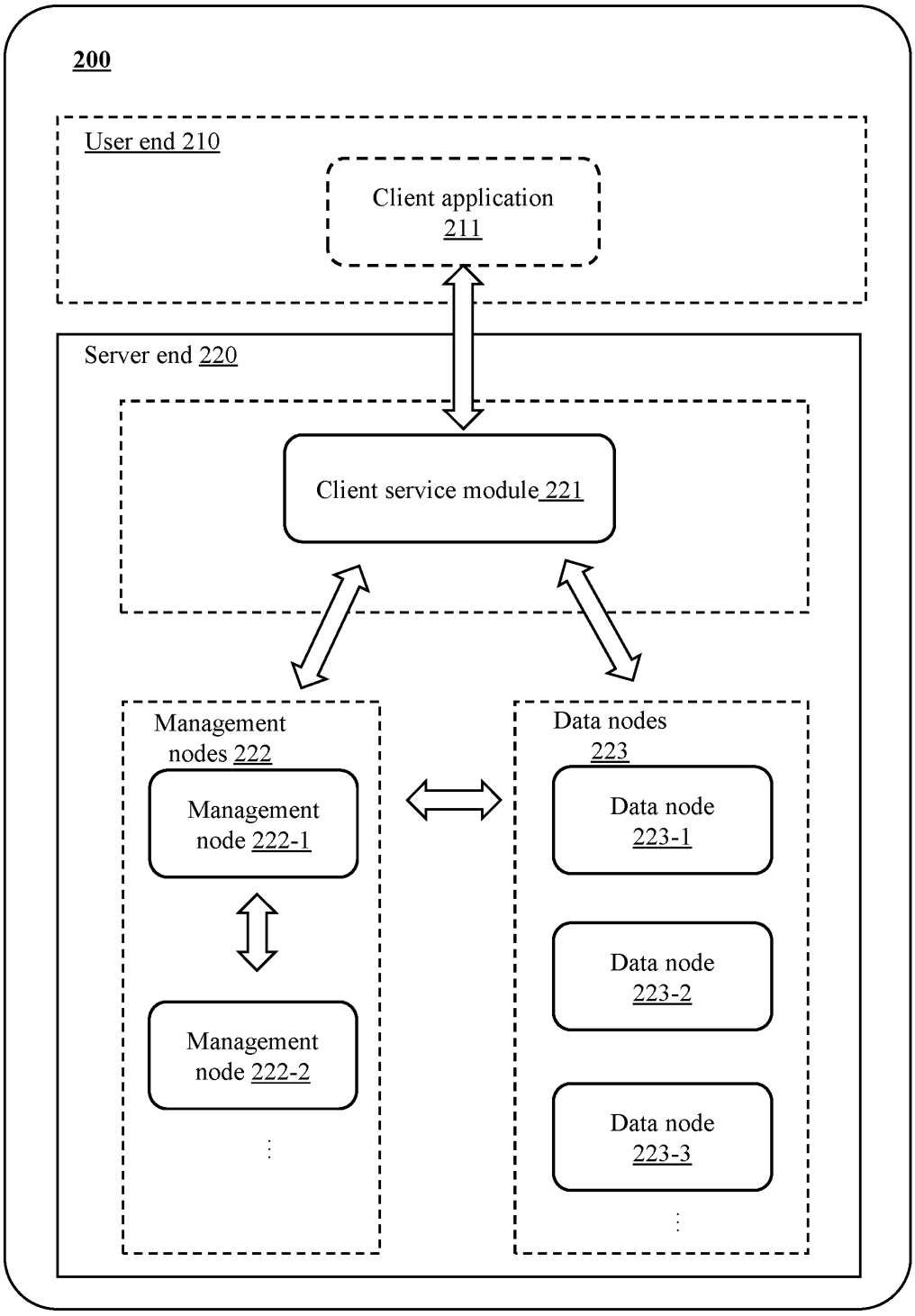

200
User end 210
Client application 211
Server end 220
Client service module 221
Management nodes 222
Management node 222-1
Management node 222-2
Data nodes 223
Data node 223-1
Data node 223-2
Data node 223-3

FIG. 2A

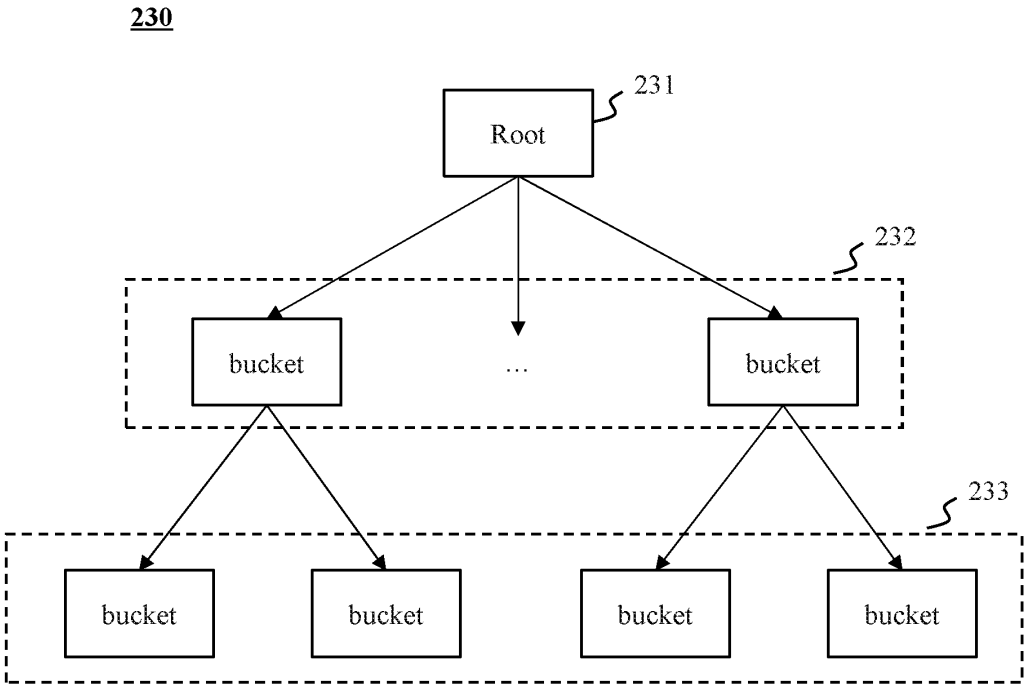

230
231
Root
232
bucket
...
bucket
233
bucket
bucket
bucket
bucket

FIG. 2B

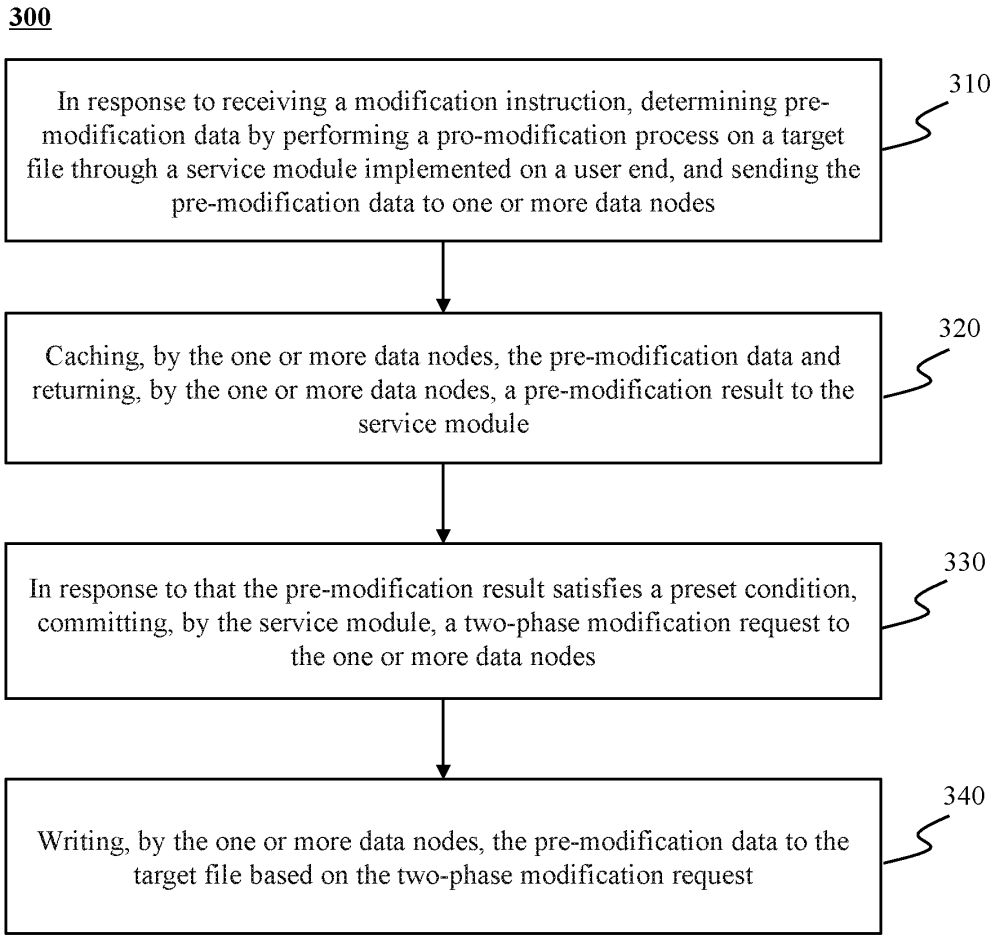

300
In response to receiving a modification instruction, determining pre-modification data by performing a pro-modification process on a target file through a service module implemented on a user end, and sending the pre-modification data to one or more data nodes
310
Caching, by the one or more data nodes, the pre-modification data and returning, by the one or more data nodes, a pre-modification result to the service module
320
In response to that the pre-modification result satisfies a preset condition, committing, by the service module, a two-phase modification request to the one or more data nodes
330
Writing, by the one or more data nodes, the pre-modification data to the target file based on the two-phase modification request
340

FIG. 3A

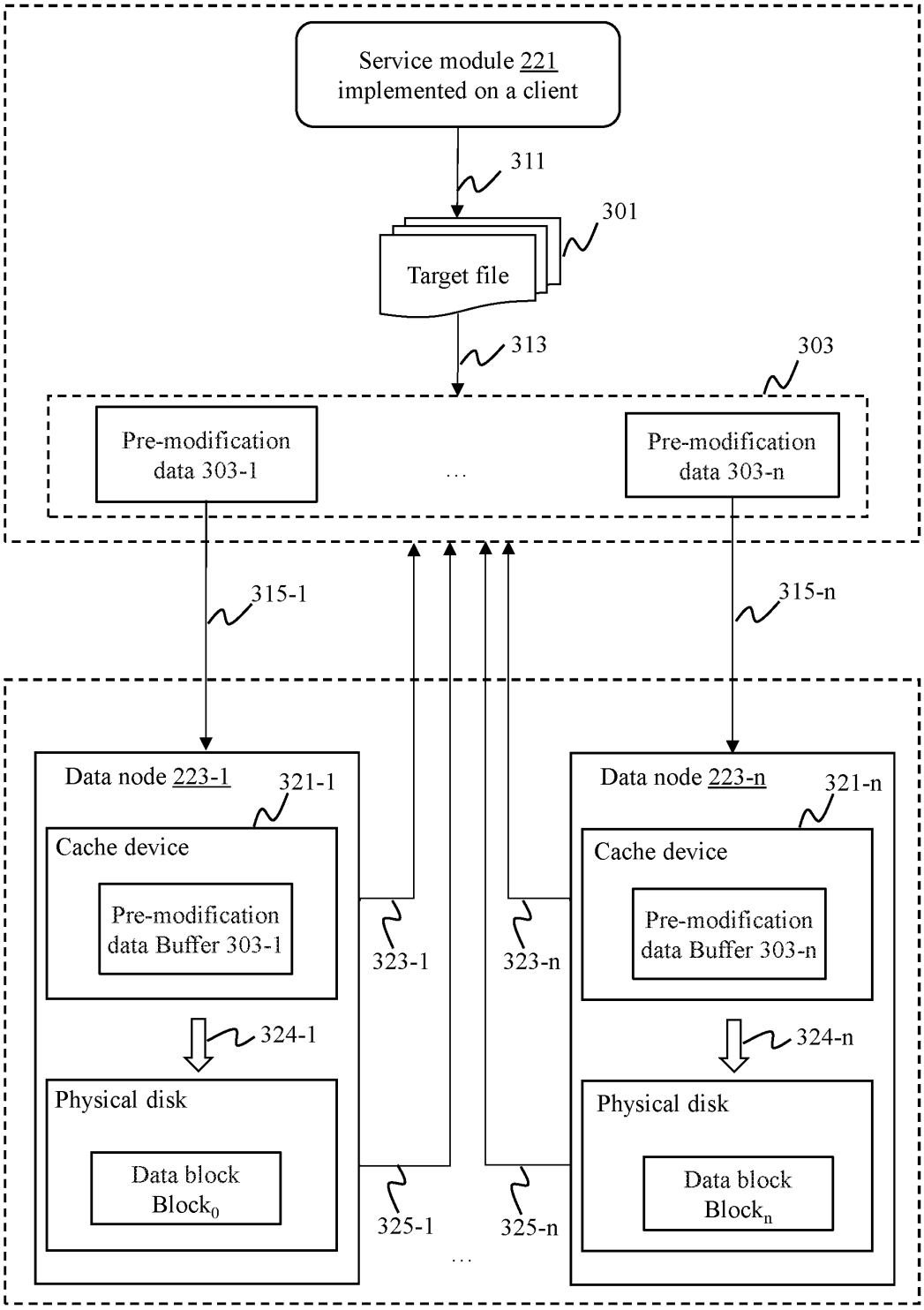

Service module 221 implemented on a client
311
301
Target file
313
303
Pre-modification data 303-1
...
Pre-modification data 303-n
315-1
315-n
Data node 223-1
321-1
Cache device
Pre-modification data Buffer 303-1
323-1
324-1
Physical disk
Data block Block0
325-1
Data node 223-n
321-n
Cache device
Pre-modification data Buffer 303-n
323-n
324-n
Physical disk
Data block Blockn
325-n
...

FIG. 3B

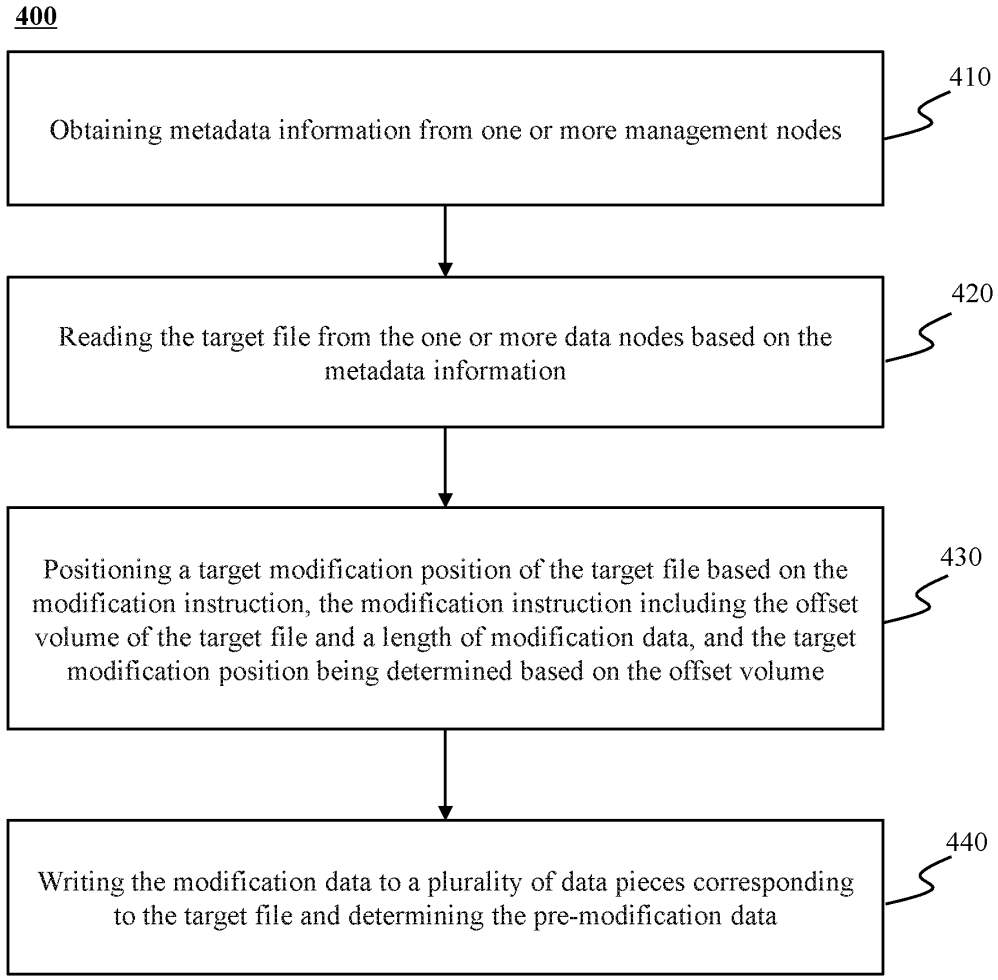

400
Obtaining metadata information from one or more management nodes
410
Reading the target file from the one or more data nodes based on the metadata information
420
Positioning a target modification position of the target file based on the modification instruction, the modification instruction including the offset volume of the target file and a length of modification data, and the target modification position being determined based on the offset volume
430
Writing the modification data to a plurality of data pieces corresponding to the target file and determining the pre-modification data
440

FIG. 4

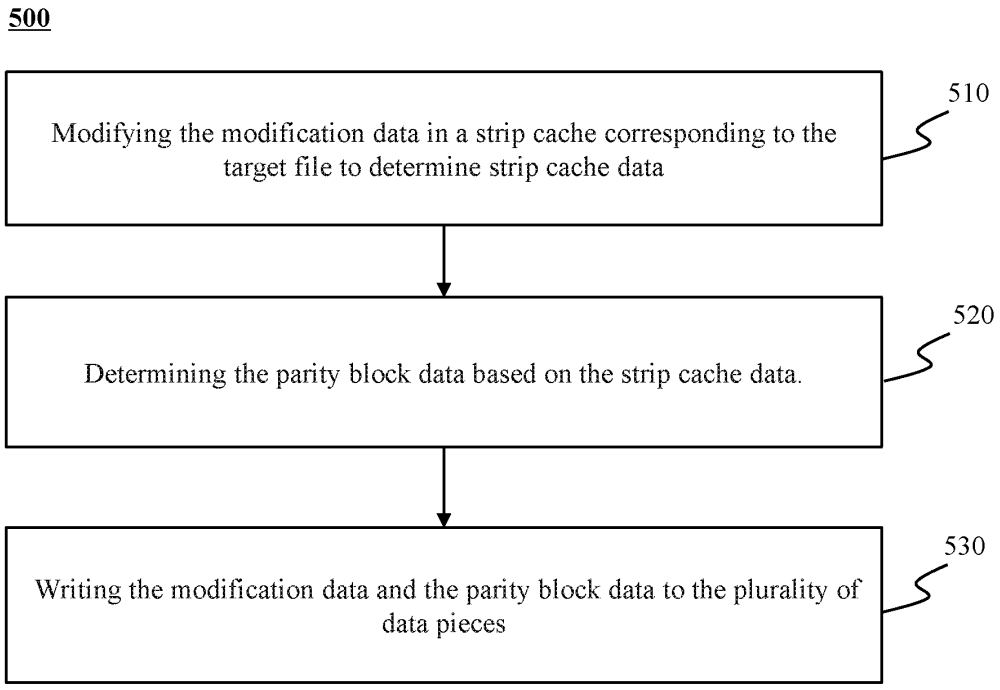

500
Modifying the modification data in a strip cache corresponding to the target file to determine strip cache data
510
Determining the parity block data based on the strip cache data.
520
Writing the modification data and the parity block data to the plurality of data pieces
530

FIG. 5A

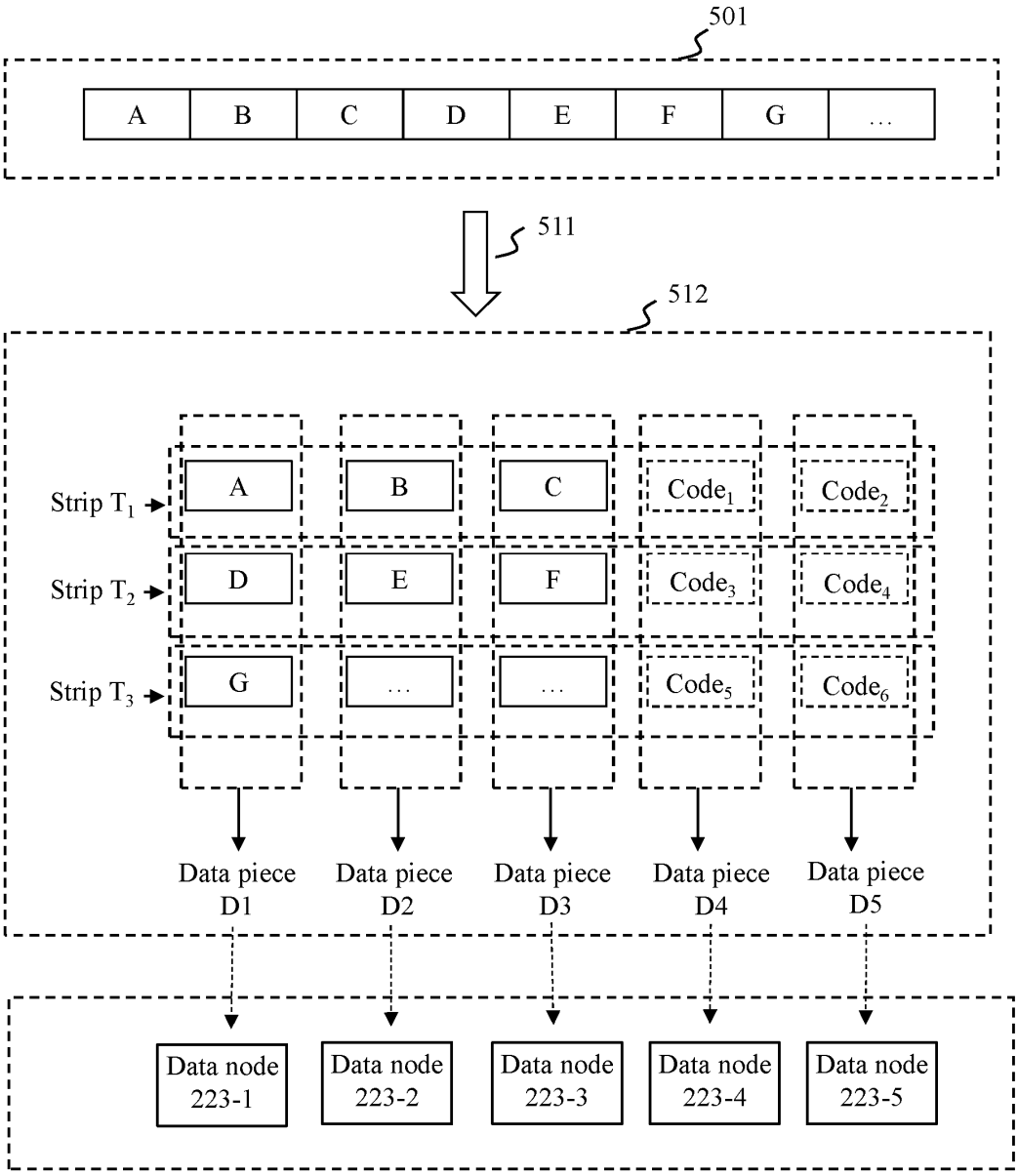

501
A
B
C
D
E
F
G
…
511
512
Strip T1
Strip T2
Strip T3
A
B
C
Code1
Code2
D
E
F
Code3
Code4
G
…
…
Code5
Code6
Data piece D1
Data piece D2
Data piece D3
Data piece D4
Data piece D5
Data node 223-1
Data node 223-2
Data node 223-3
Data node 223-4
Data node 223-5

FIG. 5B

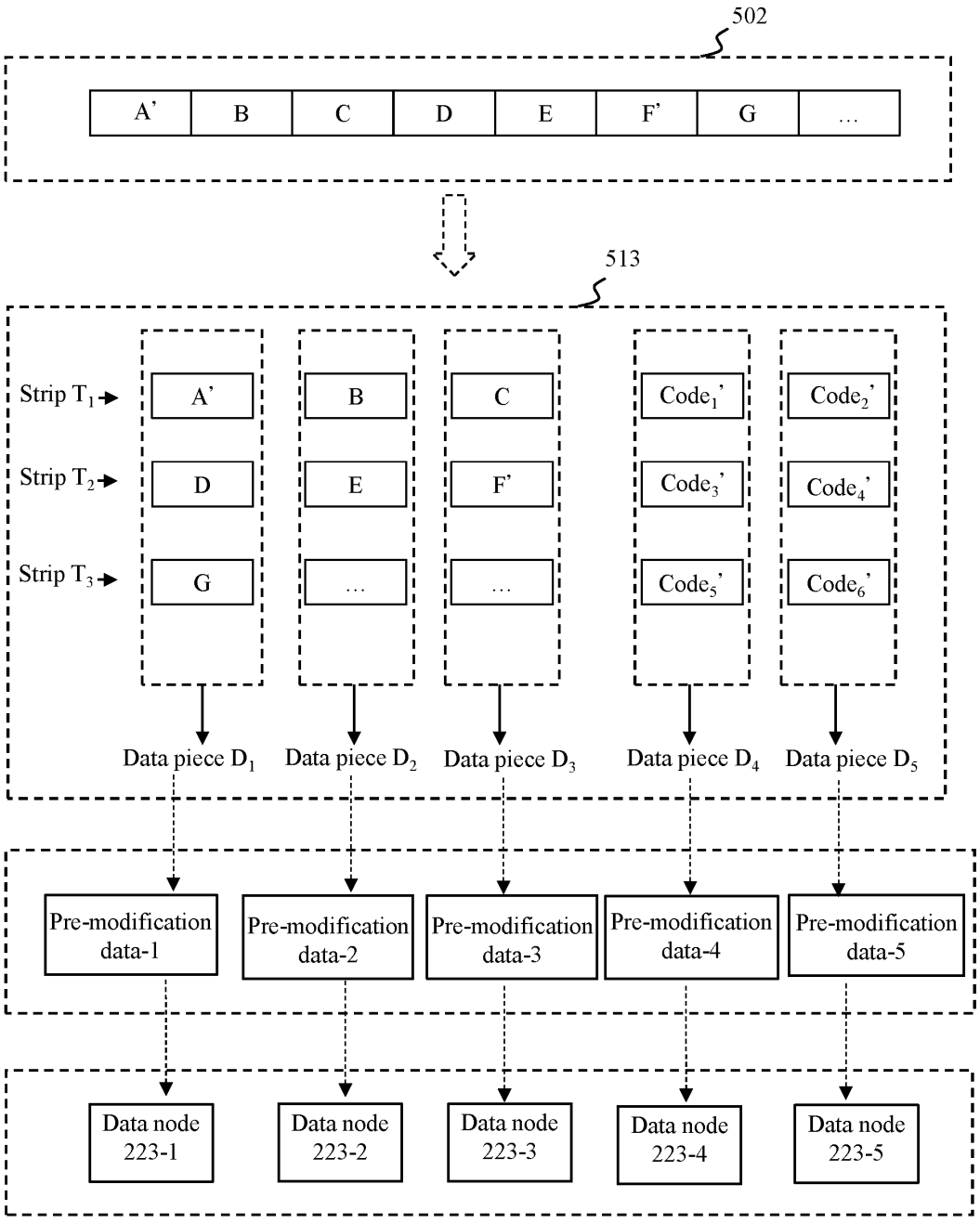

502
A'
B
C
D
E
F'
G
…
513
Strip T1
Strip T2
Strip T3
A'
D
G
B
E
…
C
F'
…
Code1'
Code3'
Code5'
Code2'
Code4'
Code6'
Data piece D1
Data piece D2
Data piece D3
Data piece D4
Data piece D5
Pre-modification data-1
Pre-modification data-2
Pre-modification data-3
Pre-modification data-4
Pre-modification data-5
Data node 223-1
Data node 223-2
Data node 223-3
Data node 223-4
Data node 223-5

FIG. 5C

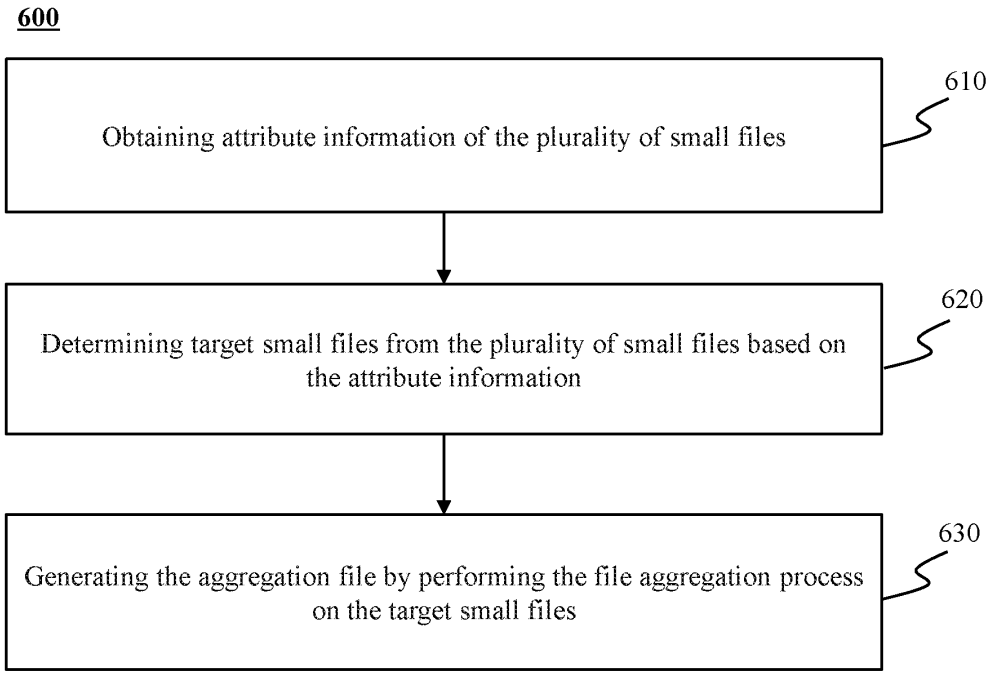

600
Obtaining attribute information of the plurality of small files
610
Determining target small files from the plurality of small files based on the attribute information
620
Generating the aggregation file by performing the file aggregation process on the target small files
630

FIG. 6

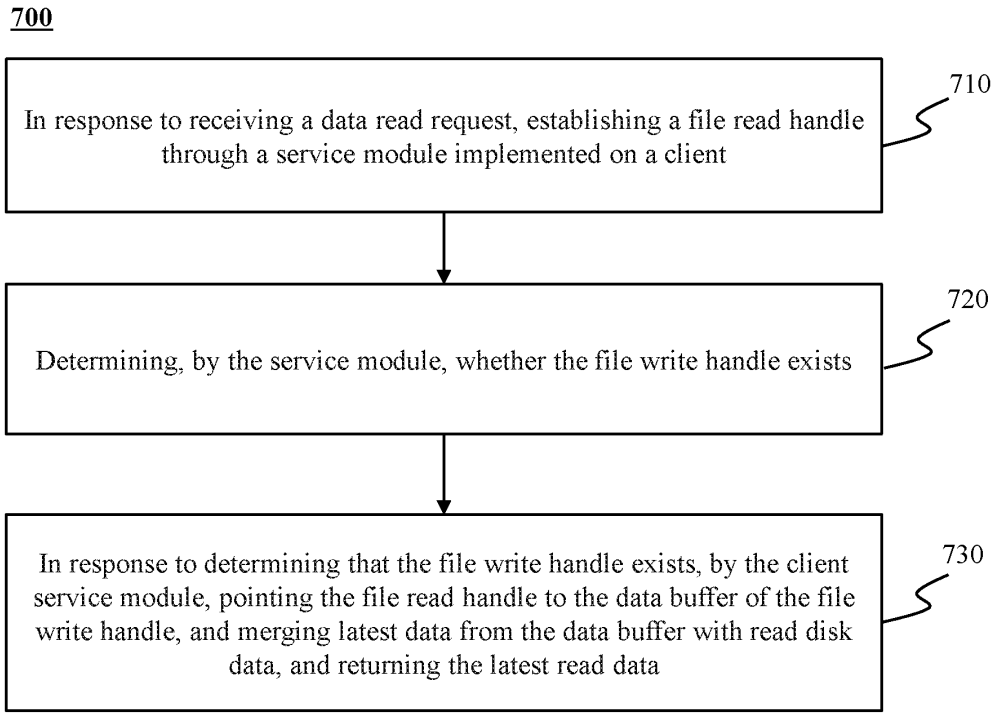

700
In response to receiving a data read request, establishing a file read handle through a service module implemented on a client
710
Determining, by the service module, whether the file write handle exists
720
In response to determining that the file write handle exists, by the client service module, pointing the file read handle to the data buffer of the file write handle, and merging latest data from the data buffer with read disk data, and returning the latest read data
730

FIG. 7A

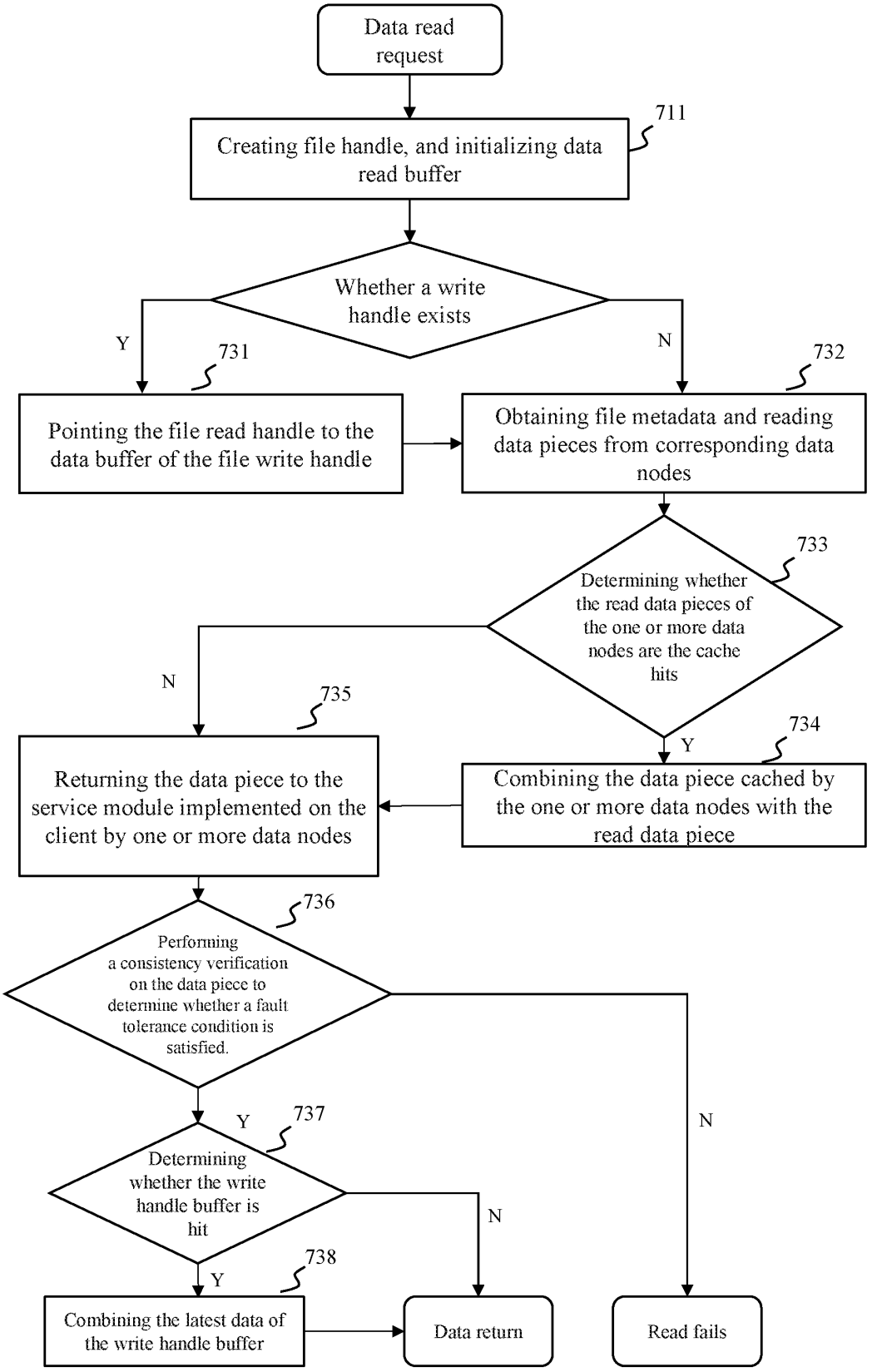

Data read request
Creating file handle, and initializing data read buffer
711
Whether a write handle exists
Y
N
731
732
Pointing the file read handle to the data buffer of the file write handle
Obtaining file metadata and reading data pieces from corresponding data nodes
733
Determining whether the read data pieces of the one or more data nodes are the cache hits
N
Y
735
734
Returning the data piece to the service module implemented on the client by one or more data nodes
Combining the data piece cached by the one or more data nodes with the read data piece
736
Performing a consistency verification on the data piece to determine whether a fault tolerance condition is satisfied.
Y
N
737
Determining whether the write handle buffer is hit
N
Y
738
Combining the latest data of the write handle buffer
Data return
Read fails

FIG. 7B

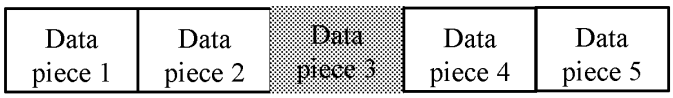

Data piece 1
Data piece 2
Data piece 3
Data piece 4
Data piece 5

FIG. 7C

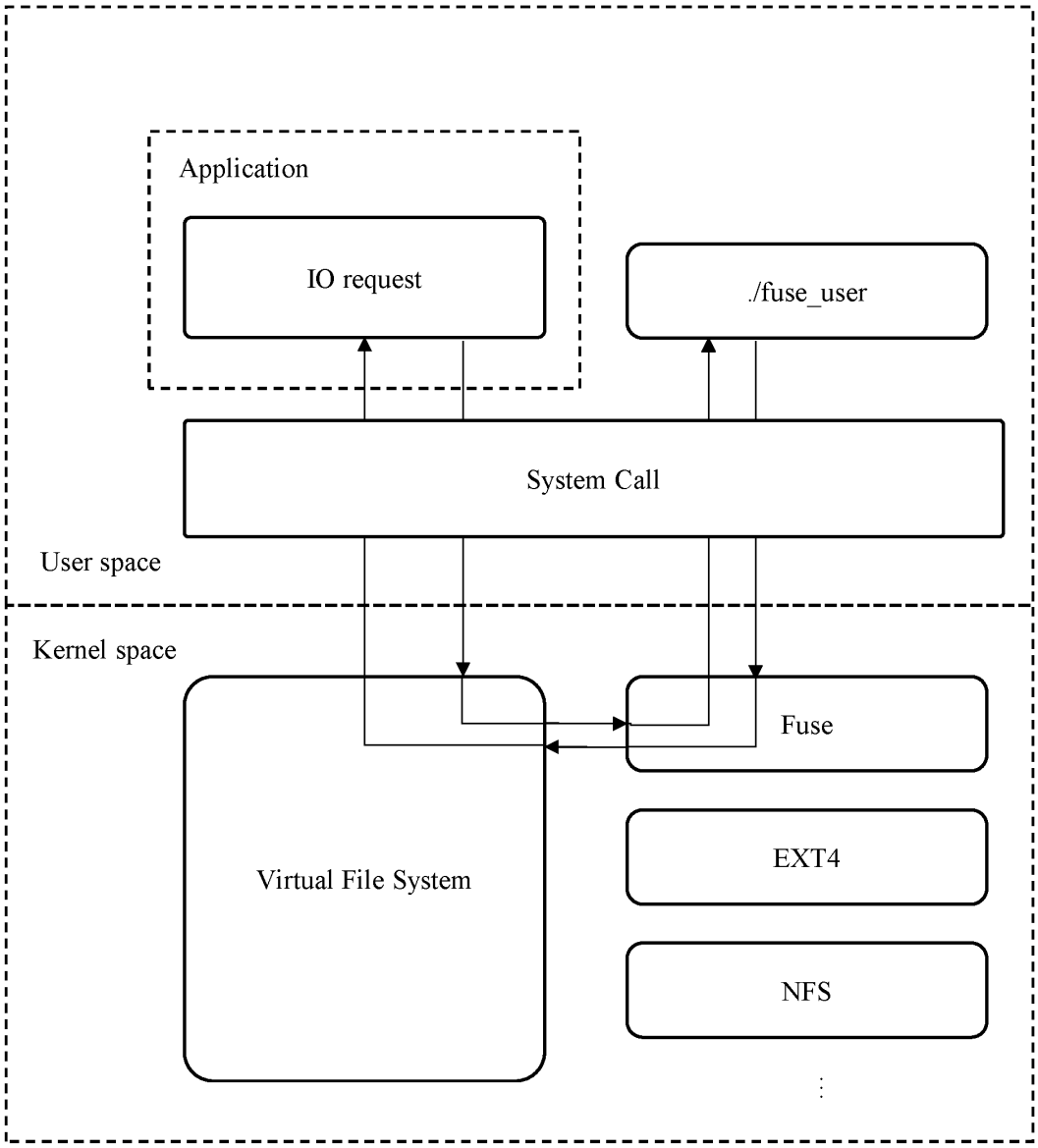

Application
IO request
./fuse_user
System Call
User space
Kernel space
Fuse
Virtual File System
EXT4
NFS

FIG. 8

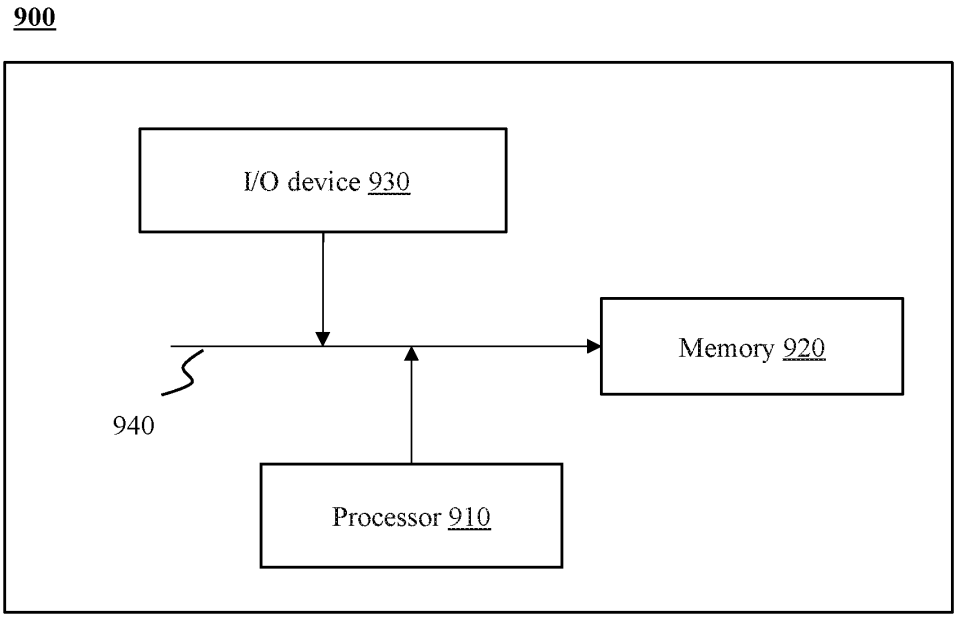


FIG. 9

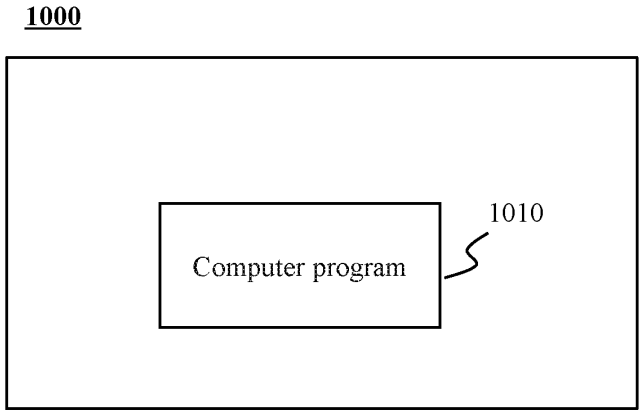


FIG. 10