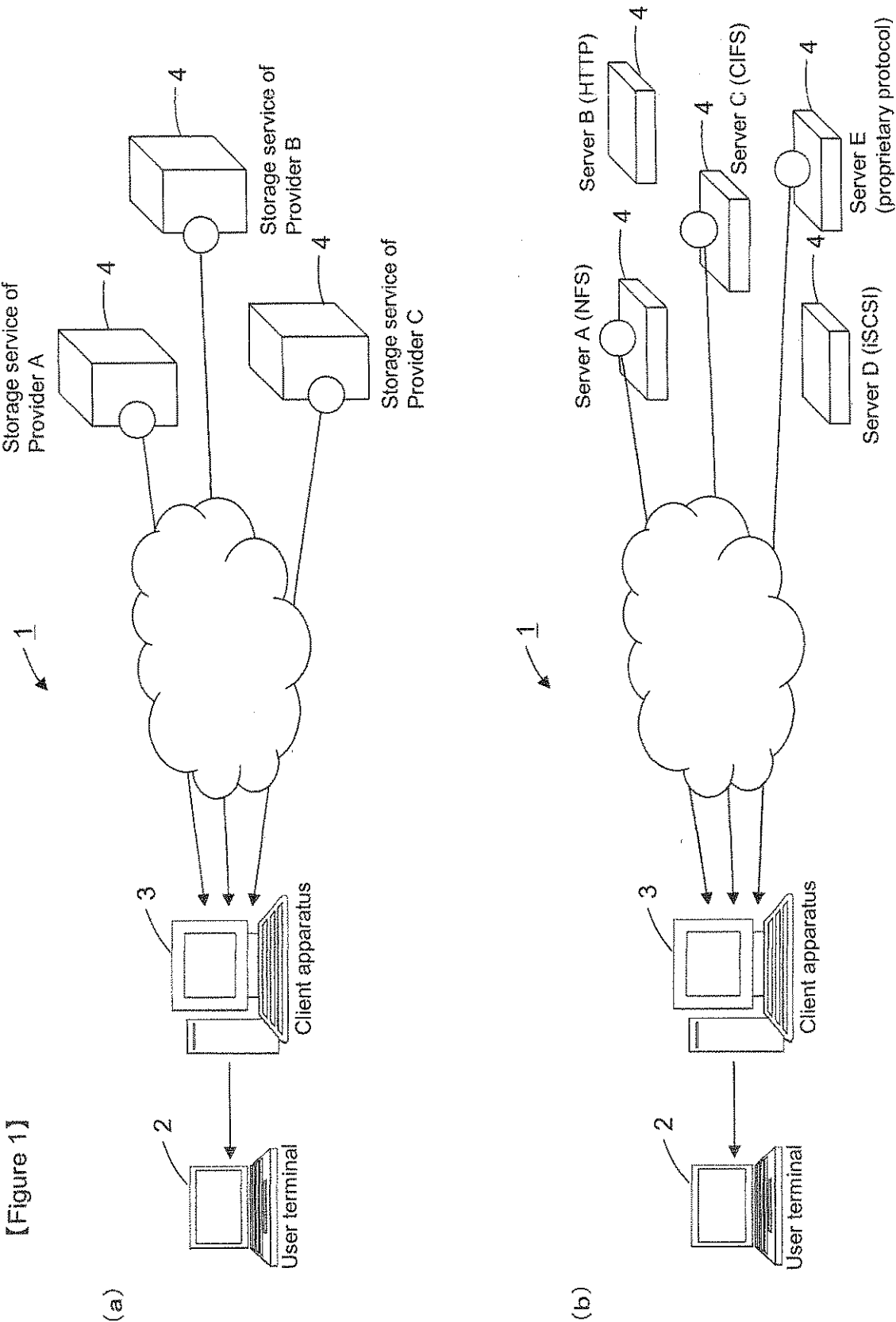
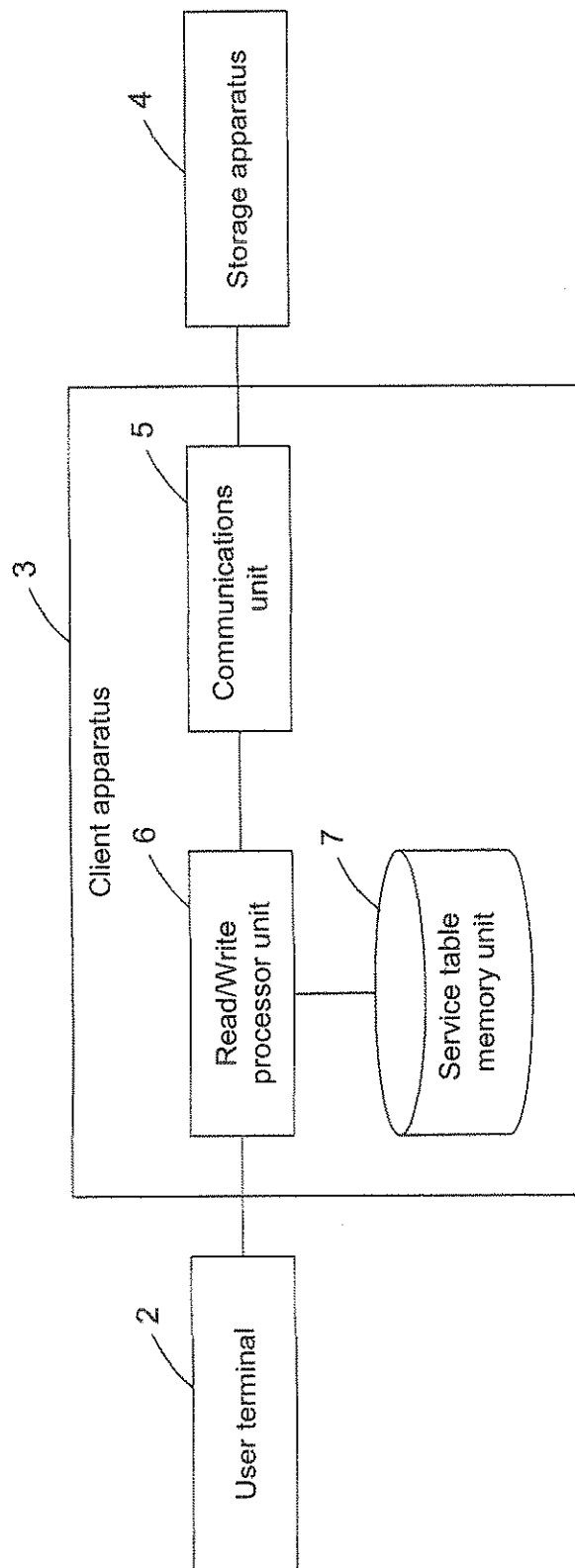


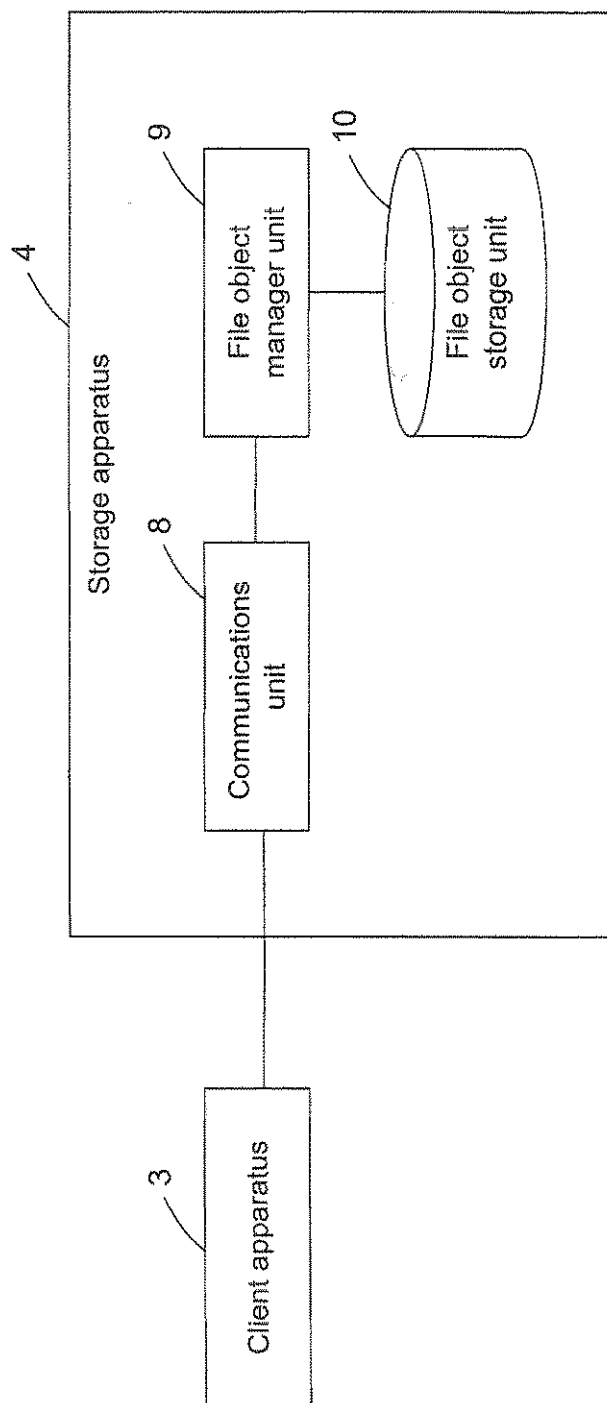
【Figure 1】



【Figure 2】



【Figure 3】



【Figure 4】

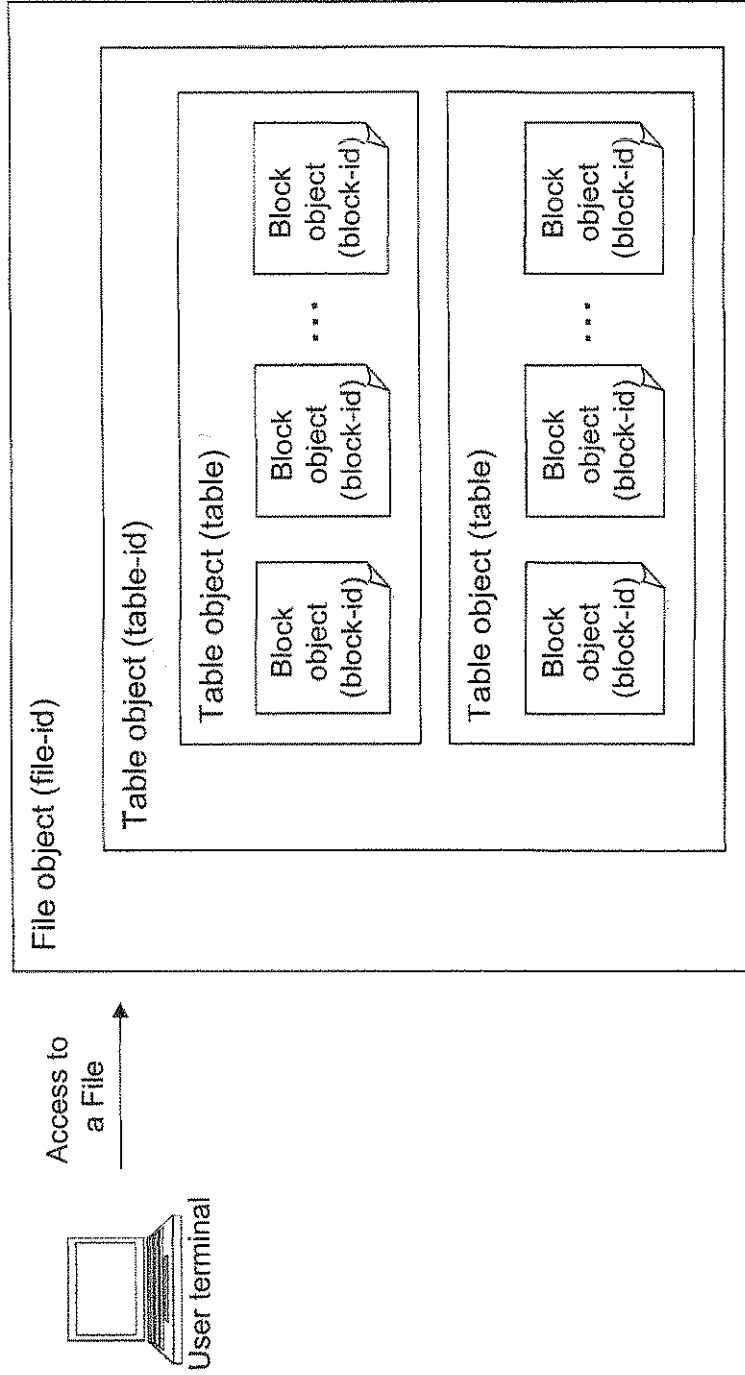
(a) Creation and Deletion of a File

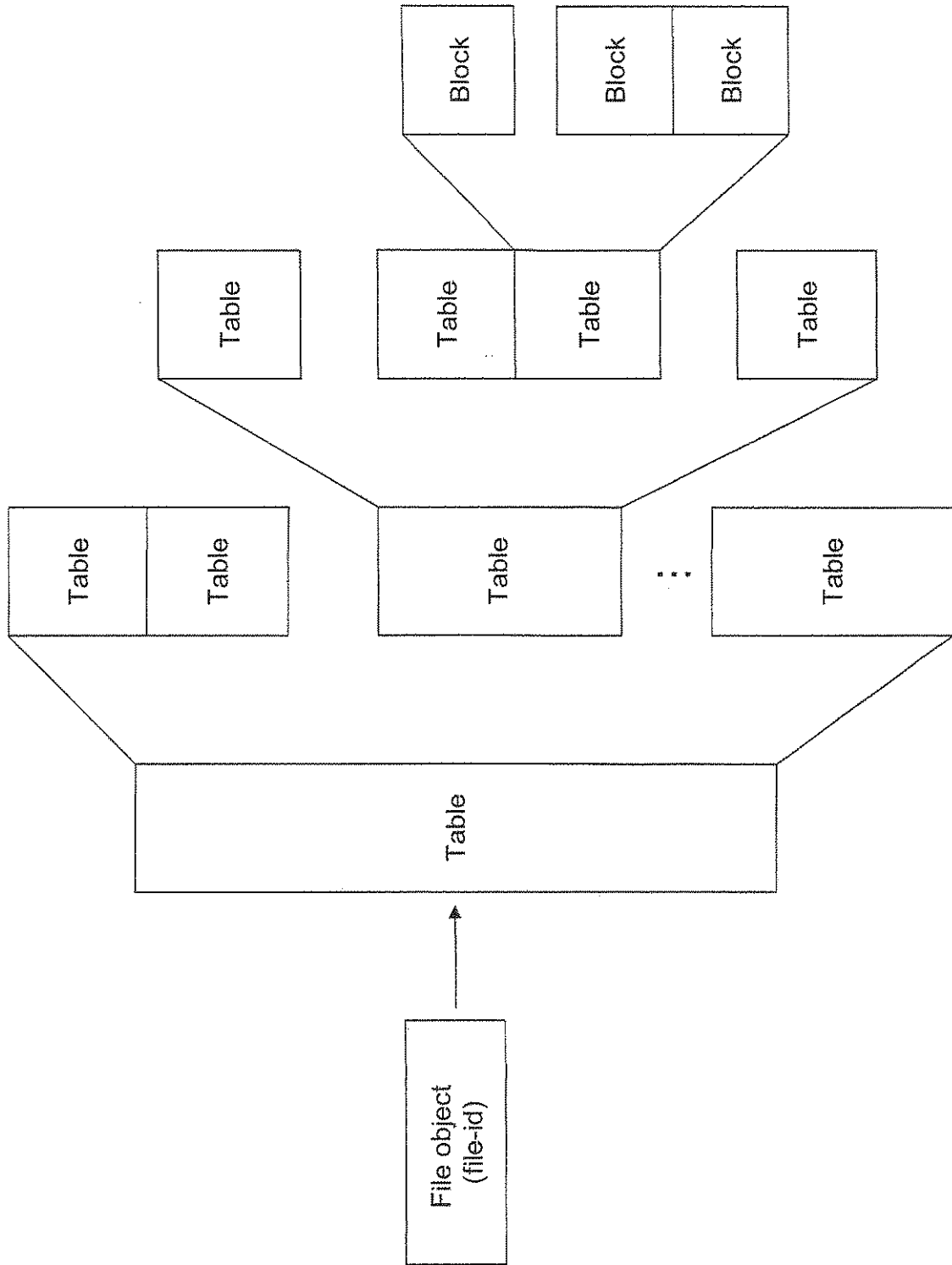
Function name	Summary	Parameter, etc.
CREATE	Creation of a file	None
DELETE	Deletion of a file	file-id

(b) Reading and Writing of Data

Function name	Summary	Parameter, etc.
READ	Reading of data from a file	file-id, offset, data (including length)
WRITE	Writing of data to a file	file-id, offset, data area (including length)

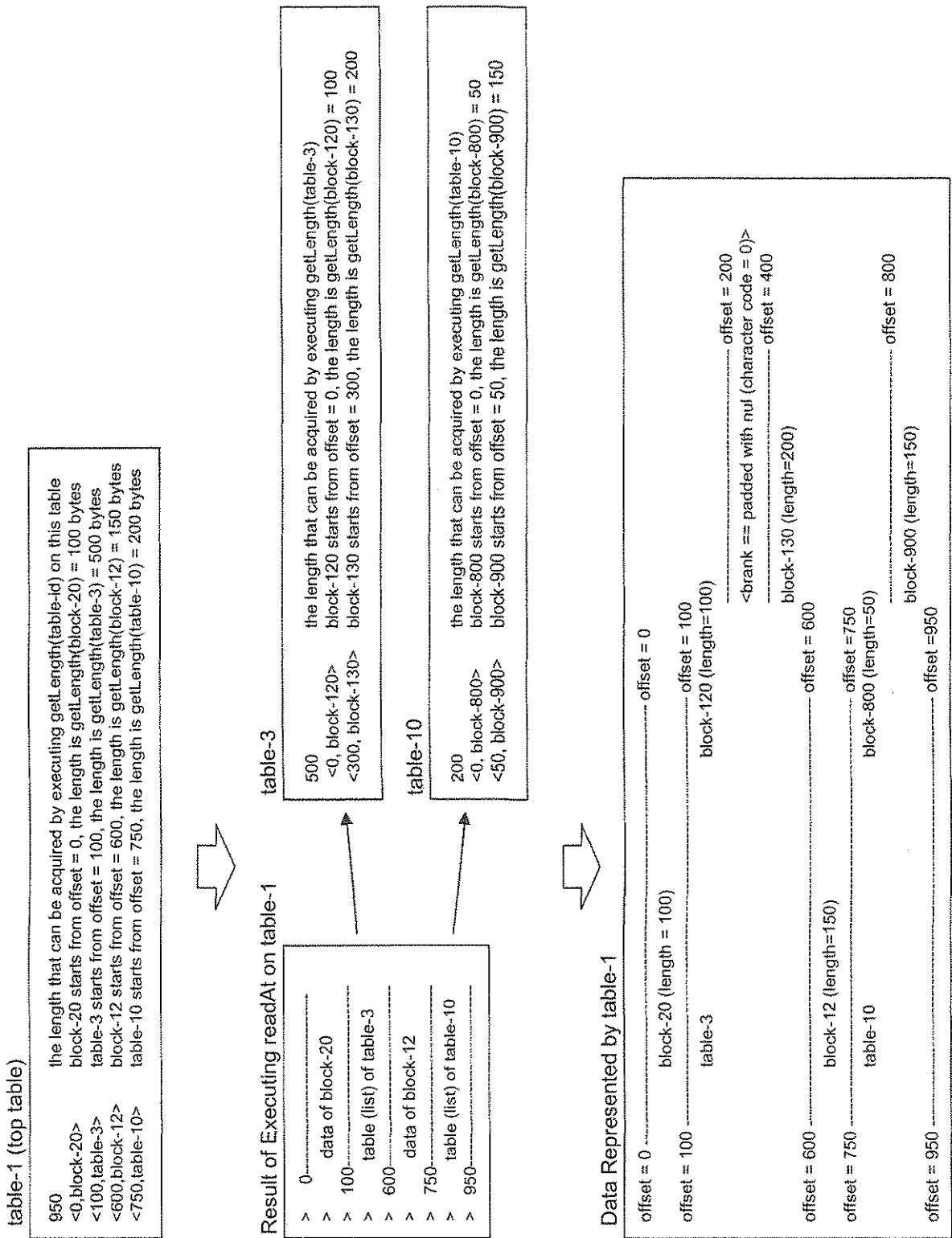
【Figure 5】



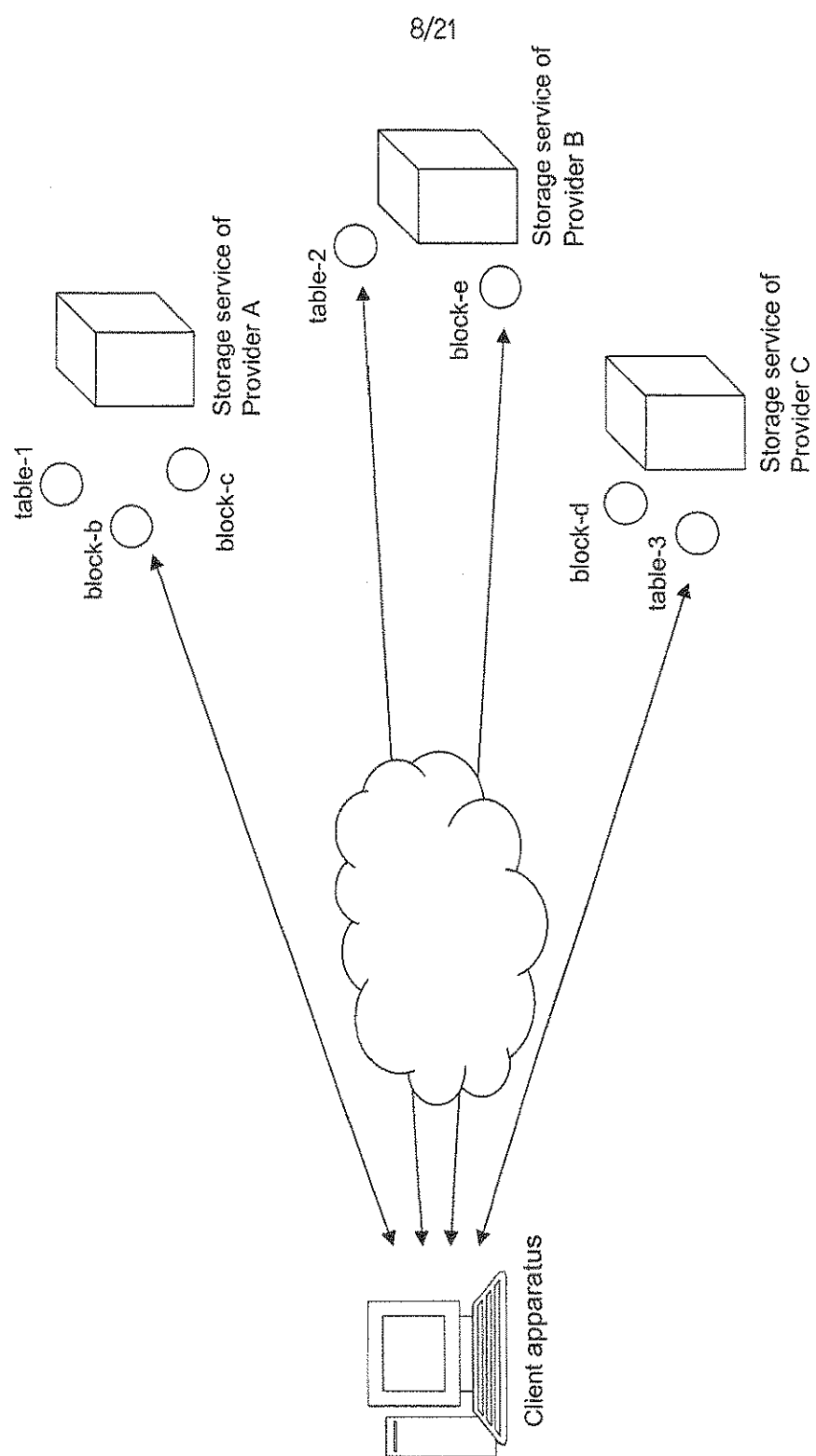


651

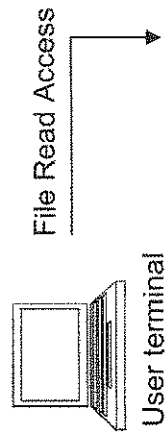
【Figure 7】



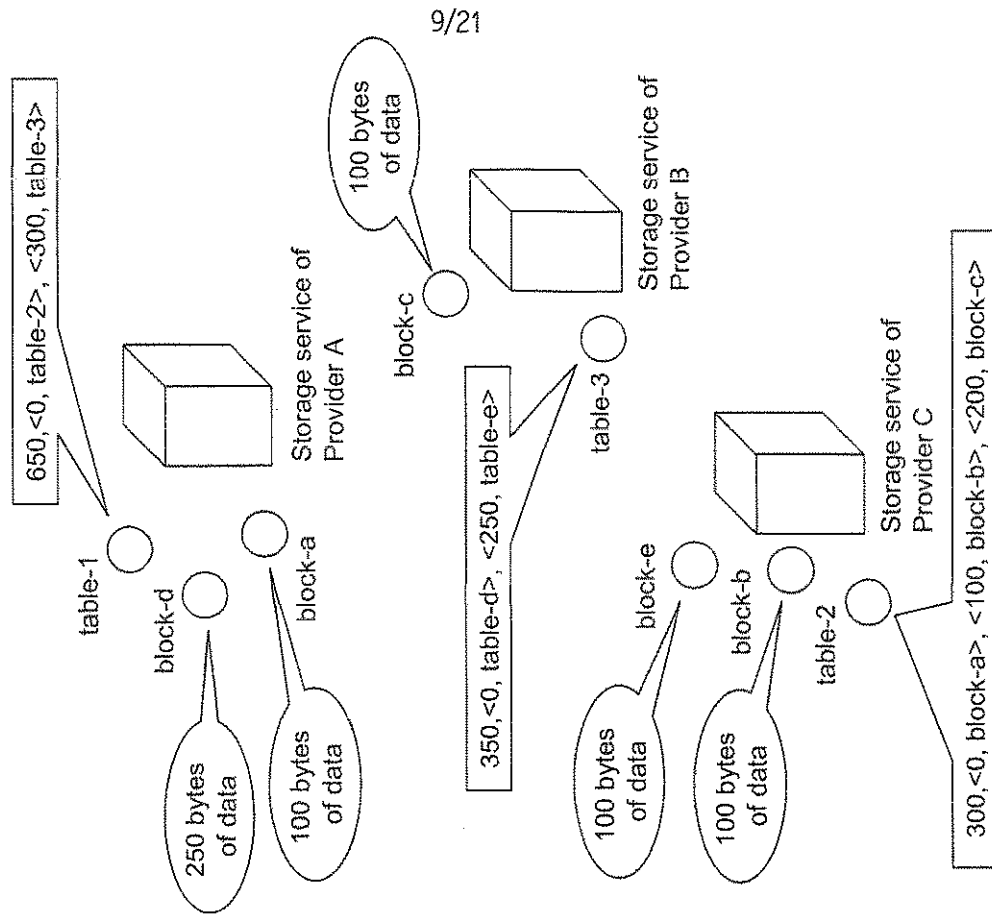
【Figure 8】



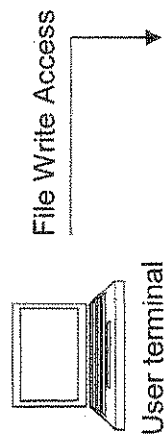
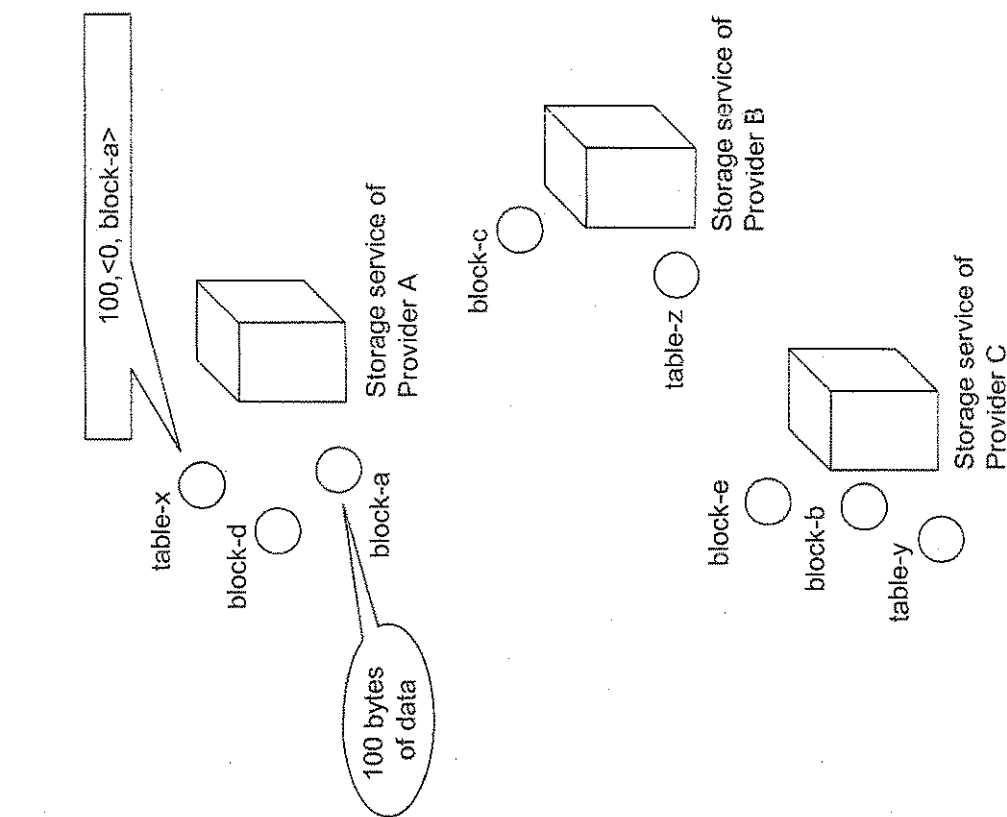
【Figure 9】



1. file-id is calculated to identify a file object.
*corresponding to protocol conversion
2. table-id corresponding to file-id is acquired in accordance with correspondence table from file-id to table-id (*1).
→ "table-1"
3. Contents of table-1 are acquired from Provider A in accordance with "virtual apparatus selection algorithm" described later.
→ table-1: "650, <0, table-2>, <300, table-3>"
*In the following, table and block information is recursively acquired.
 - 3.1. table-2: "300, <0, block-a>, <100, block-b>, <200, block-c>"
Data of blocks are loaded in accordance with the contents of the table.
 - 3.1.1. block-a: $\text{getLength}(\text{block-a}) = 100$ bytes of data are loaded to offset = 0.
 - 3.1.2. block-b: $\text{getLength}(\text{block-b}) = 100$ bytes of data are loaded to offset = 100
 - 3.1.3. block-c: $\text{getLength}(\text{block-c}) = 100$ bytes of data are loaded to offset = 200.
 - 3.2. table-3: "350, <0, block-d>, <250, block-e>"
Data of blocks are loaded in accordance with the contents of the table.
 - 3.2.1. block-d: $\text{getLength}(\text{block-d}) = 250$ bytes are loaded to offset = 300 (300 + 0).
 - 3.2.2. block-e: $\text{getLength}(\text{block-e}) = 100$ bytes are loaded to offset = 550 (300 + 250)
4. End.

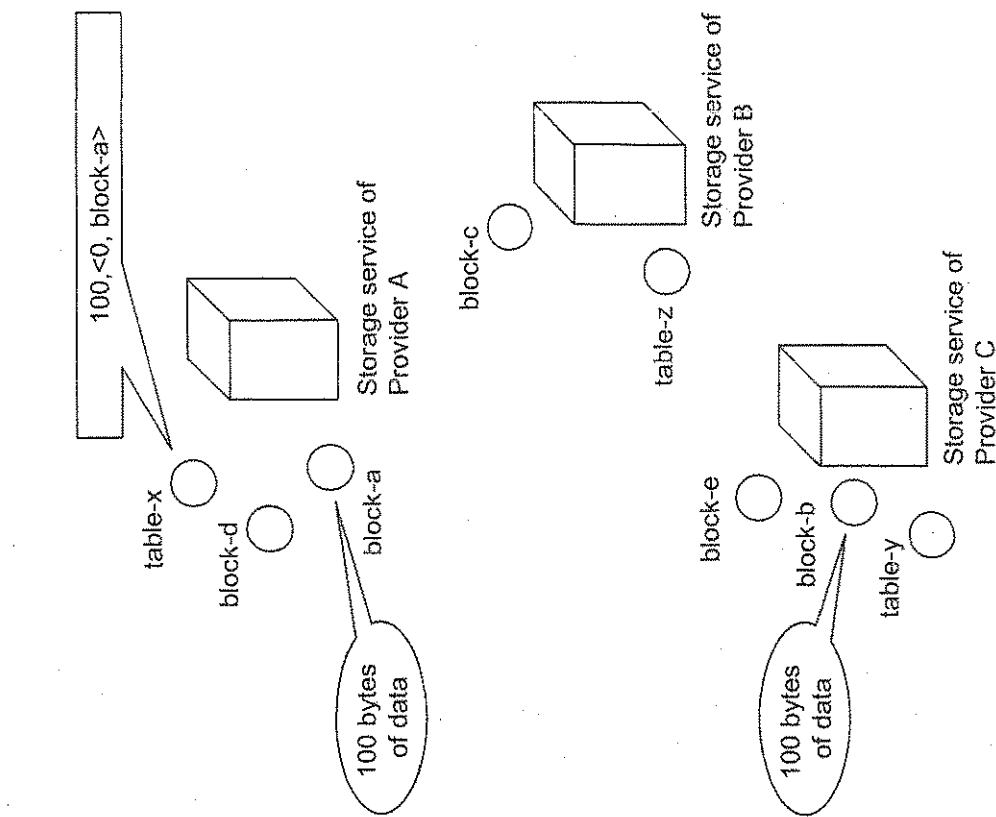


【Figure 10】



1. [CREATE] A new file is created.
The top table is set to EMPTY_TABLE.
2. 100 bytes of data, data, is written from the top.
- 2.1. [WRITE-1] writeAt on file
writeAt(EMPTY_TABLE, 0, data, 0, 100) is executed on the top table EMPTY_TABLE.
- 2.2. [WRITE-2] writeAt on table
A block is created since there is no block corresponding to the offset.
A block block-a is created by
block-a = createId() and
putContent(block-a, 0 - 0, data, 0, 100),
and then the list is updated.
The process exits from the main loop of "writeAt,"
updates length to 100, then executes
table-x = createId() and
putTable(table-x, 100, <0, block-a>),
and returns table-x.
- 2.1. (continued)
The top table is set to table-x above, and the process ends.

【Figure 11】



3. 150 bytes of data, data, is written from offset = 50.

3.1. [WRITE-1] writeAt on file
`writeAt(table-x, 50, data, 0, 150)` is executed on the top table table-x.

3.2. [WRITE-2] writeAt on table
 loop-1:

The block corresponding to offset = 50 is block-a.
`writeAt(block-a, 50 - 0, data, 0, 50)` is executed.

3.2.1. [WRITE-3]

100 bytes of current data of block-a, orig, is loaded and orig is overwritten at offset = 50 with (data, 0, 50).

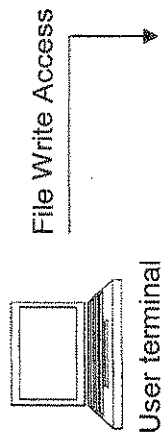
A new block is created by
`block-b = createId()` and
`putContent(block-b, 100, orig)`

3.2. (continued)

The list is updated with the above block-b assigned as a new block.
 → The new list is <0, block-b> (in memory).

`object_offset = 50 + 50 = 100,`
`bufflen = 150 - 50 = 100, and`
`bufoff = 0 + 50 = 50`
 are set and the process goes to the next loop.

【Figure 12】



3.2. (continued)

loop-2:

There is no block corresponding to offset = 100.

A new block is created by executing

`block-c = createId()` and

`putContent(block-c, 100, (data, 50, 100)).`

The list of the table is updated.

→ The new list is

`<0, block-b>, <100, block-c>.`

`object_offset = 100 + 100 = 200`

`buflen = 100 - 100`

`bufoff = 50 + 100 = 150`

Since `buflen == 0`, the process ends the loop,

sets `length = 200`, and then executes

`table-y = createId()` and

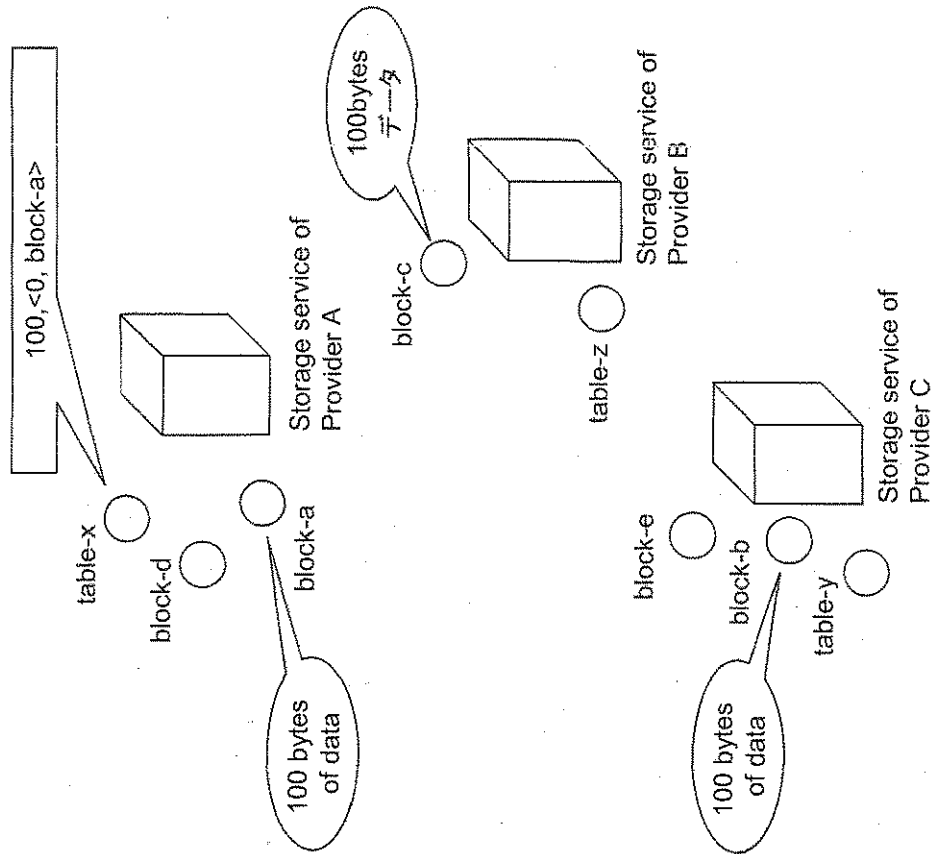
`putTable(table-y, 200, <0, block-b>, <100, block-c>)`

to create a new table.

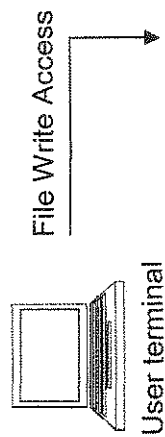
`table-y` is returned.

3.1. (continued)

The top table of the file is set to `table-y`, and the process ends.



【Figure 13】



4. 200 bytes of data, data2, is written from object_offset = 10000.

4.1. [WRITE-1]

object_offset is judged to be large enough, and the depth of the hierarchy is increased.

→A new table is created with the current table-y being a single element.
 *This only causes change in the depth of the table, and the value acquired by getLength() is the same as table-y.

The newly created table is

table-α = createId() and
 putTable(table-α, 200, <0, table-v>).

4.2. [WRITE-1, continued]

writeAt is called for the created table, table-α.

writeAt(table-α, 10000, data2, 0, 200)

4.3. [WRITE-2]

writeAt process on table-α is started.

Since there is no existing element at object_offset = 10000,

block-d = createId() and

putContent(block-d, 200, (data2, 0, 200))

are executed to create a new block.

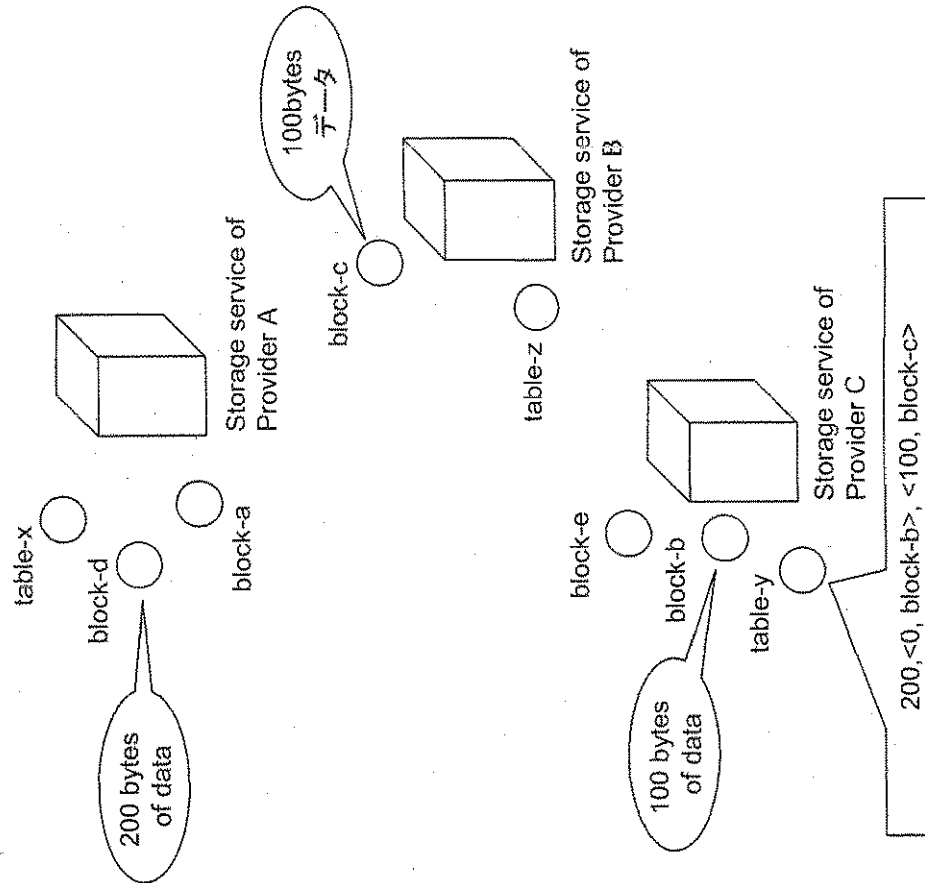
Each type of parameter is updated with

object_offset = 10000 + 200 = 10200,

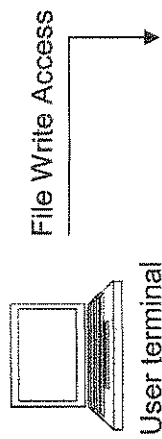
bufLen = 200 - 200, and

bufOff = 0 + 200 = 200,

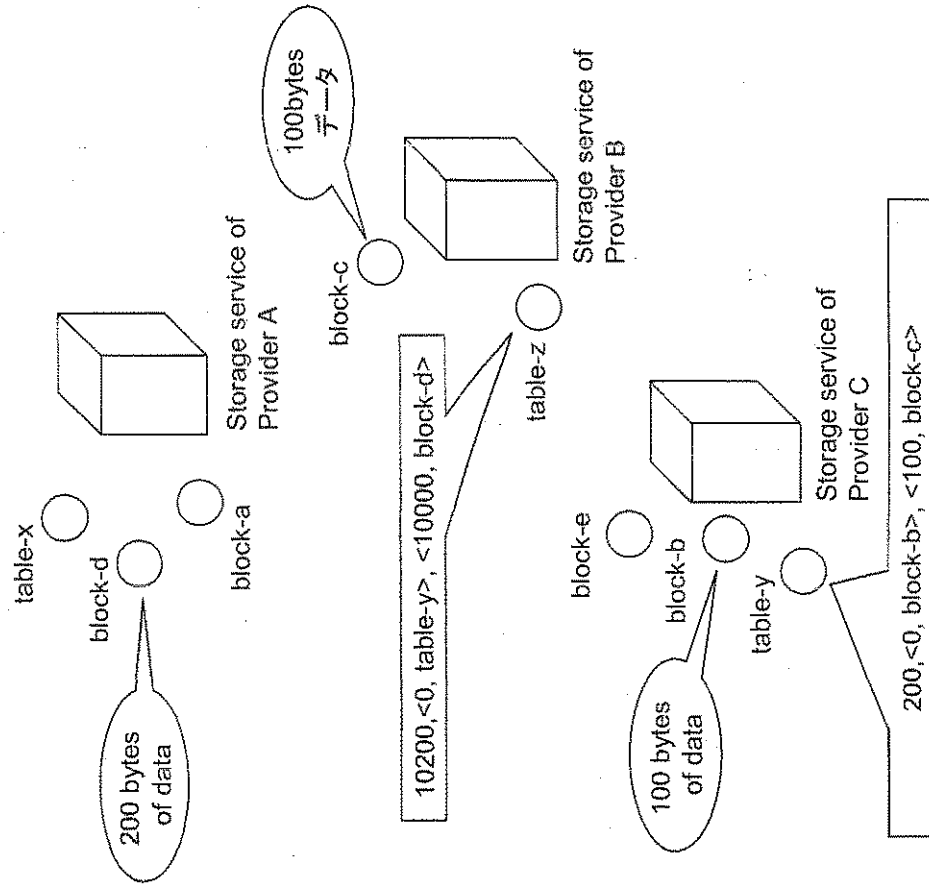
and the loop ends.



【Figure 14】



4.4. [WRITE-2, continued]
 Since length < object_offset + len, the length is updated by setting length = 10200.
 Further, the list of the table is updated.
 The new table is
 table-z = createId() and
 putTable(table-z, 10200, <0, table-y>, <10000, block-d>)
 4.1. (continued)
 The top table of the file is set to table-z, and the process ends.



【Figure 15】

Static Service Table

Number	Provider	Access method	Authorization
0	Provider A	http://a.com/storage/%o	user-1/pass-1
1	Provider B	http://b.com/disk/%u/%o	user01/pass-2
2	Provider C	nfs://c.com/home/%u/disk/%o	user001/pass-3
...	...	...	...

Before Parameter Conversion

nfs://c.com/home/%u/disk/%o



After Parameter Conversion

nfs://c.com/home/user-id/disk/0a12cd-05201a-...-ab00fa

【Figure 16】

Dynamic Service Table

Number	Provider	Access method	Authorization
01a3	Provider A	http://a.com/storage/%o	user-1/pass-1
25a0	Provider B	http://b.com/disk/%u/%o	user01/pass-2
5c8a	Provider C	nfs://c.com/home/%u/disk/%o	user001/pass-3
...	...	...	...

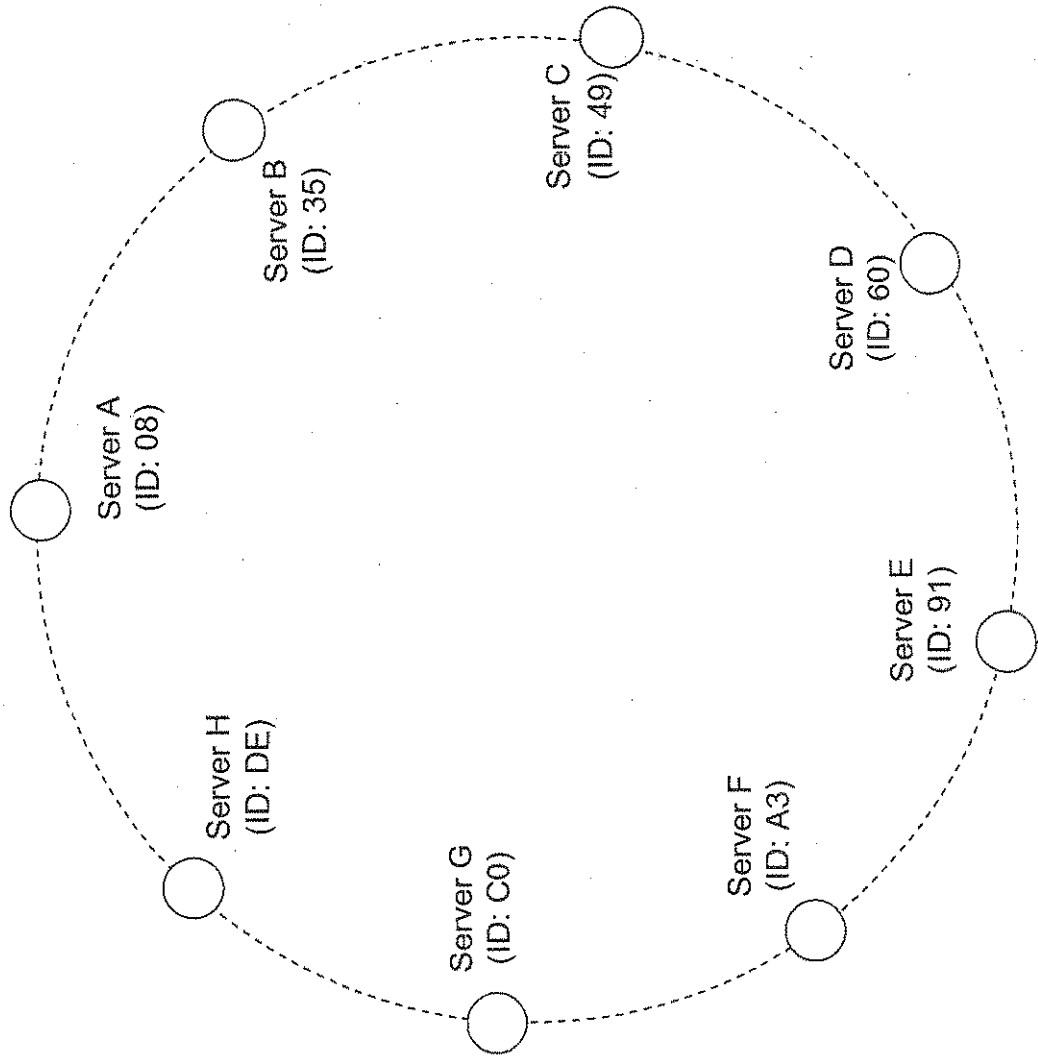
Before Parameter Conversion

nfs://c.com/home/%u/disk/%o



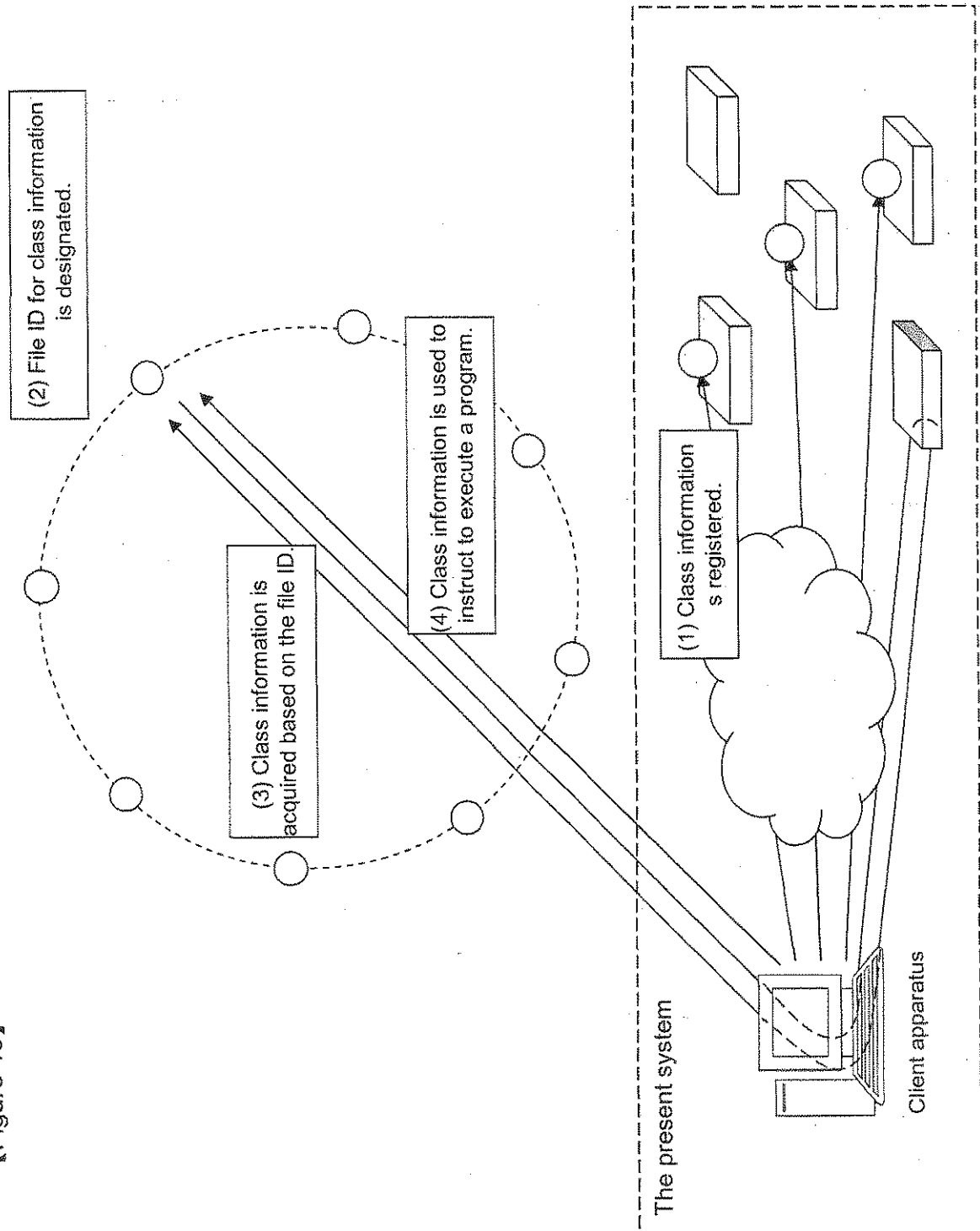
After Parameter Conversion

nfs://c.com/home/user-id/disk/0a12cd-05201a-...-ab00fa

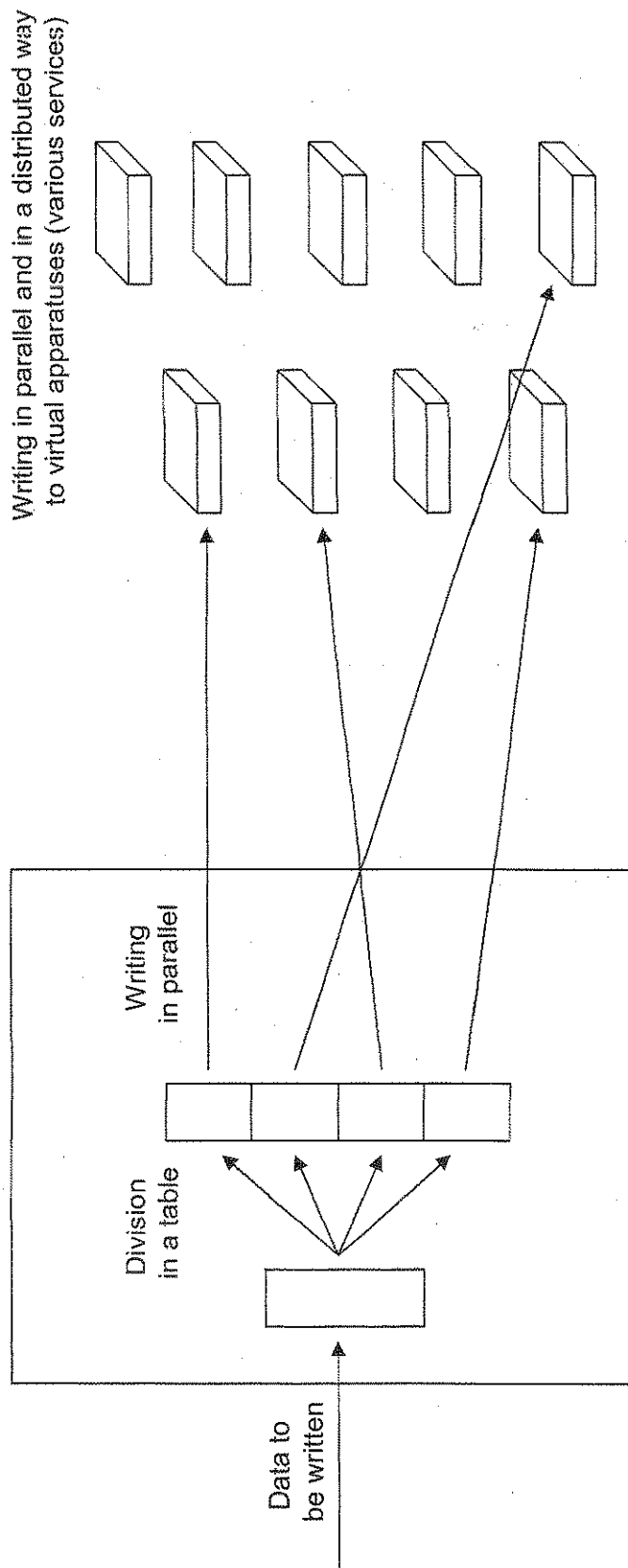


【Figure 17】

【Figure 19】



【Figure 20】



【Figure 21】

