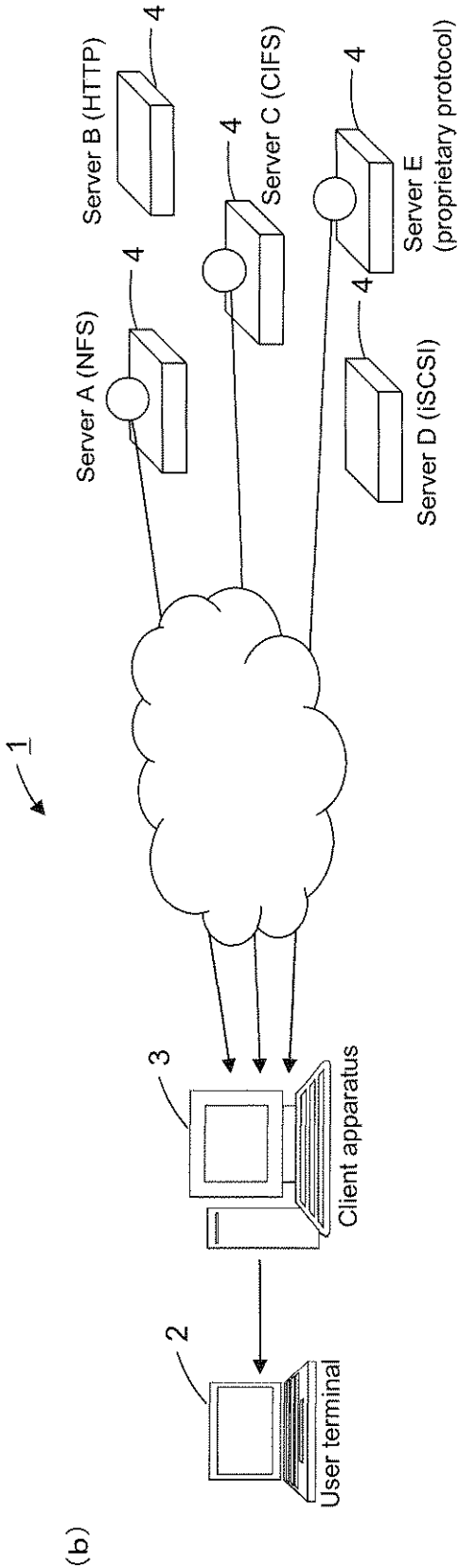
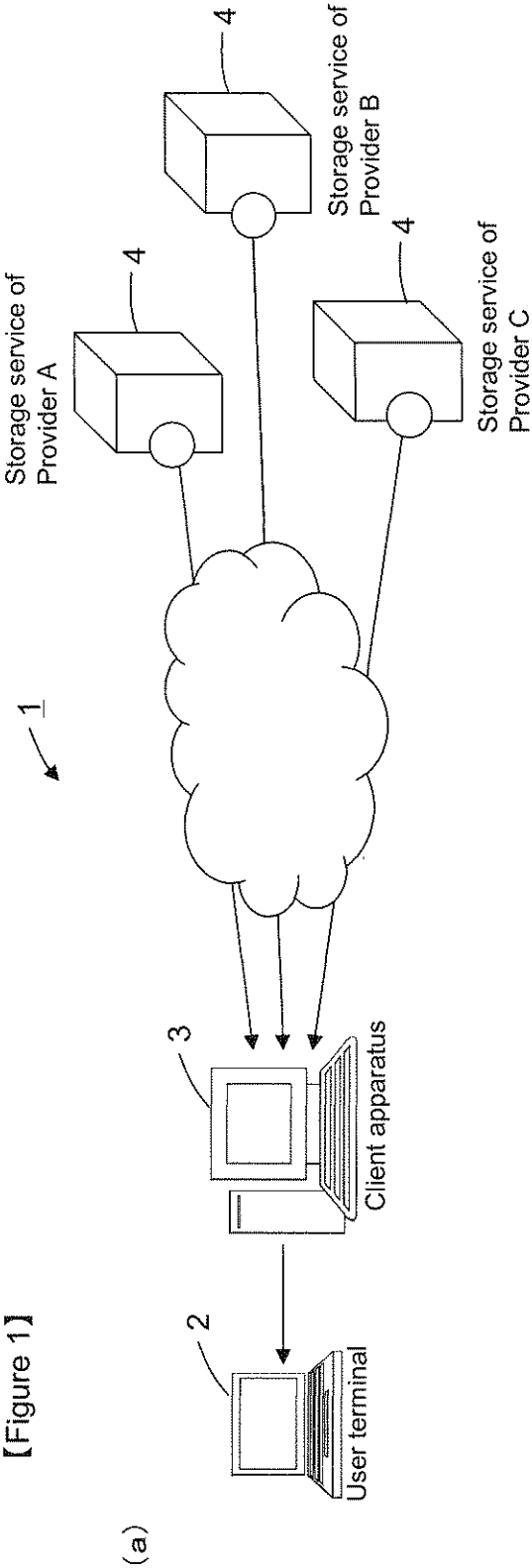
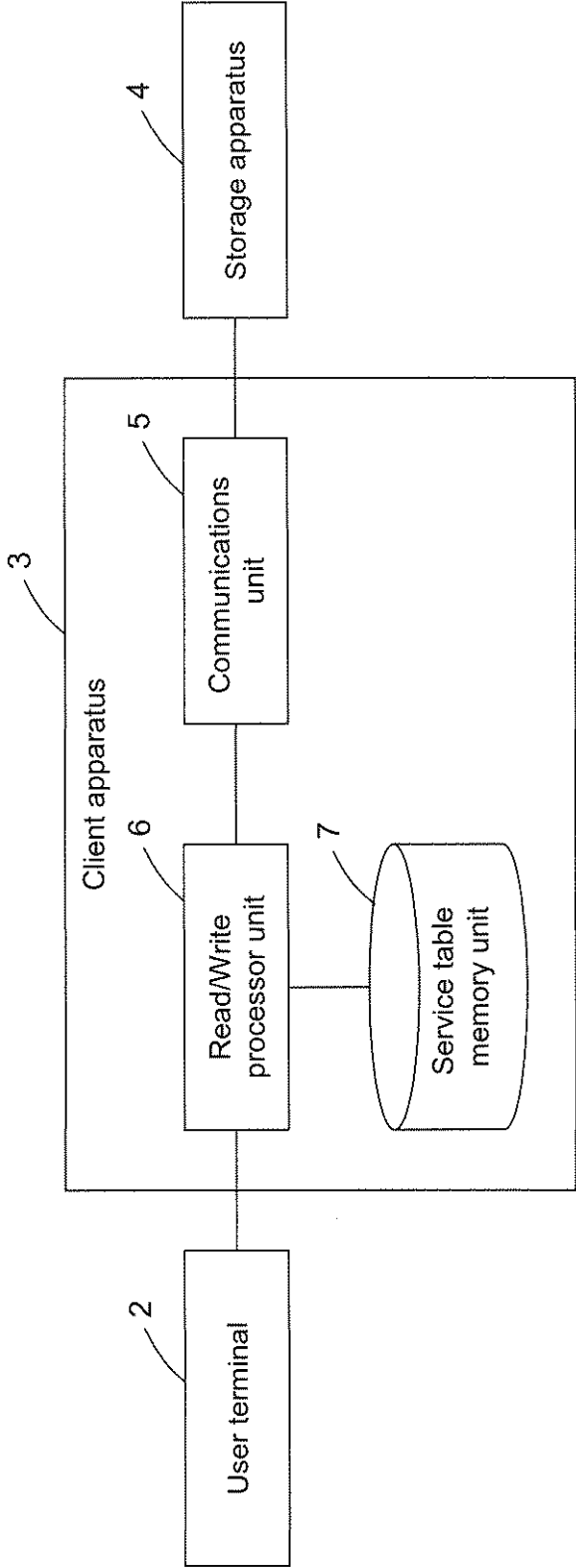
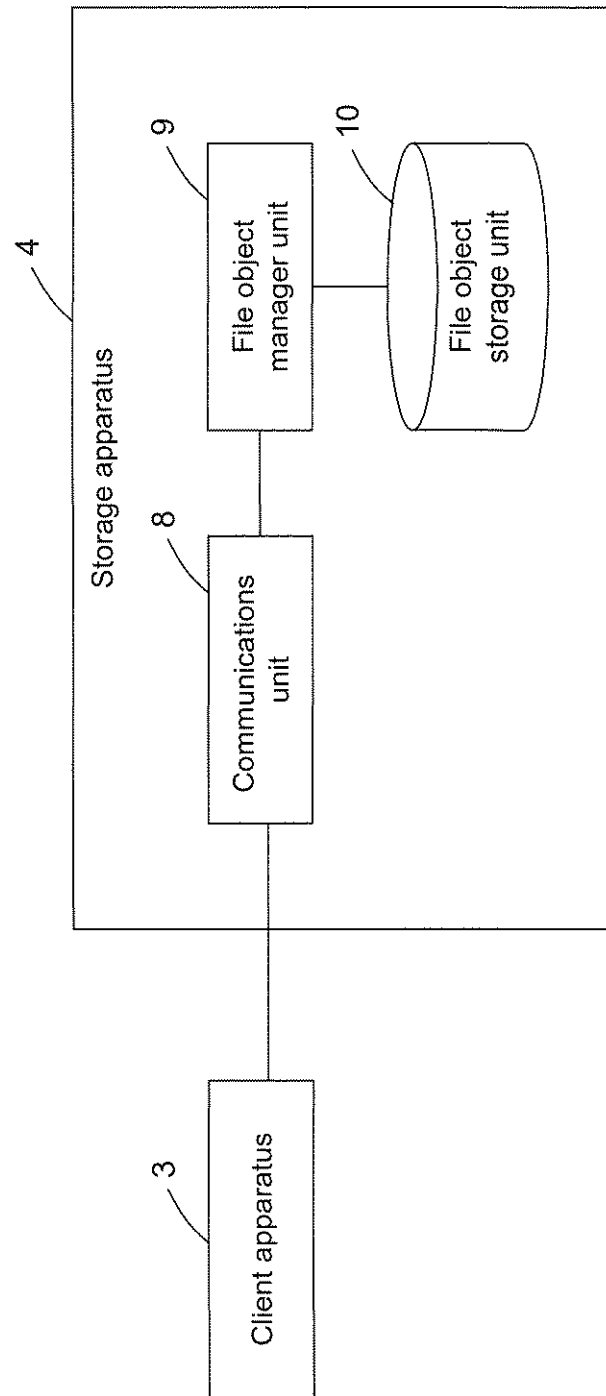


【Figure 1】



【Figure 2】





【Figure 3】

【Figure 4】

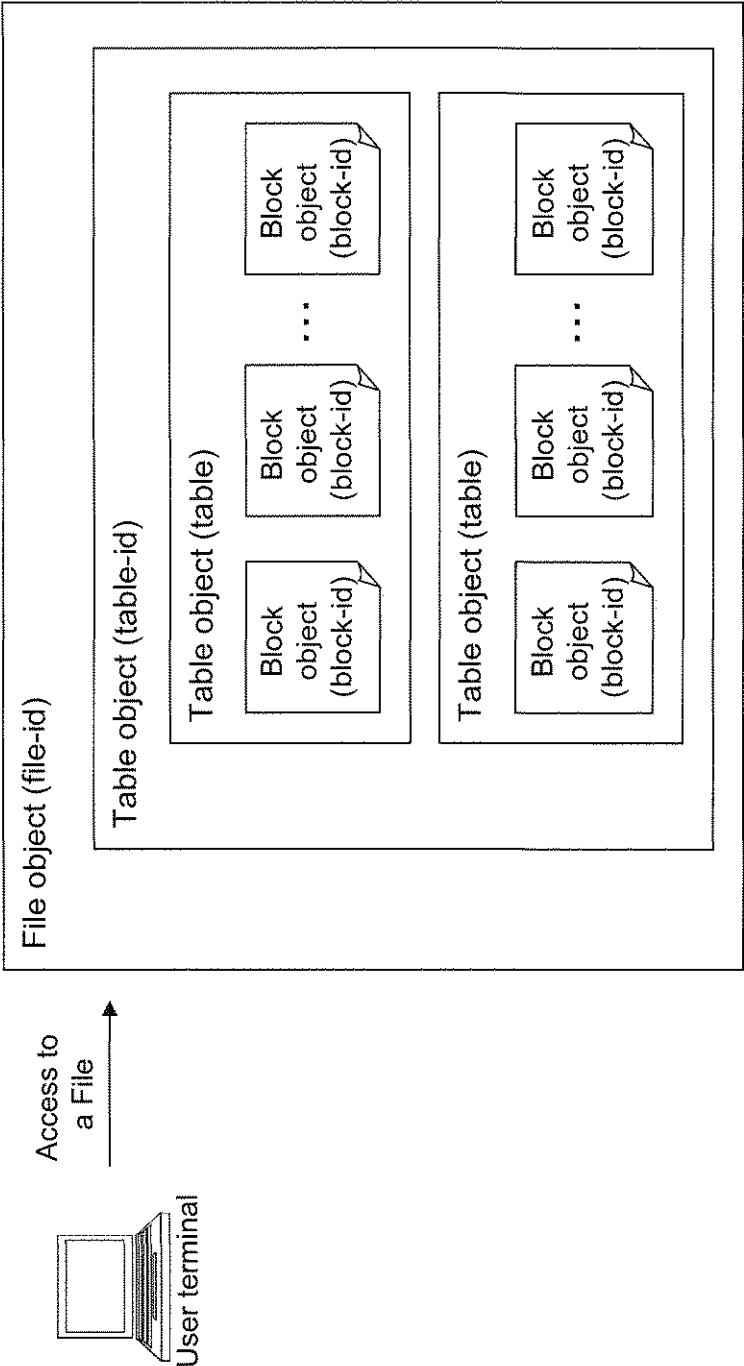
(a) Creation and Deletion of a File

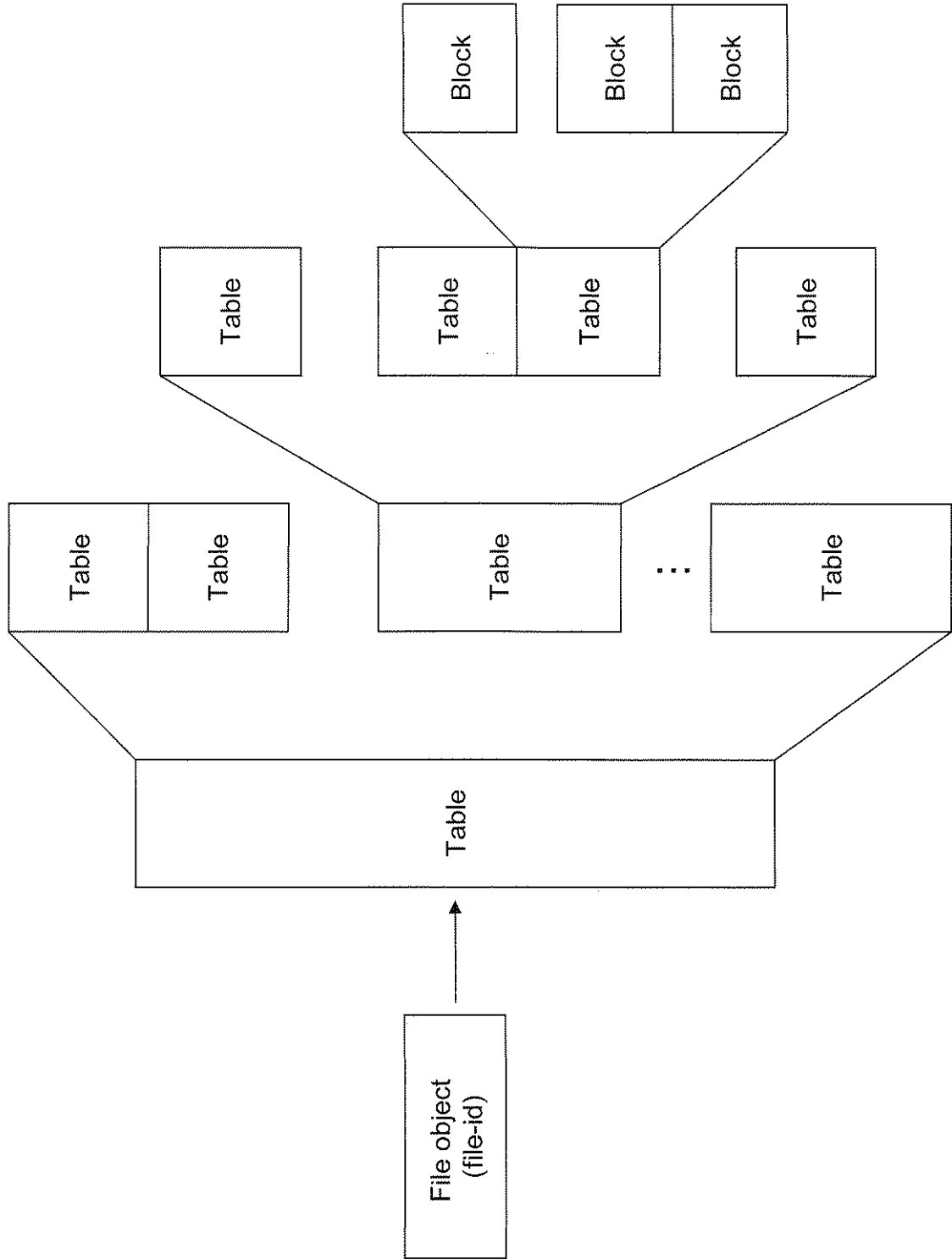
Function name	Summary	Parameter, etc.
CREATE	Creation of a file	None
DELETE	Deletion of a file	file-id

(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)

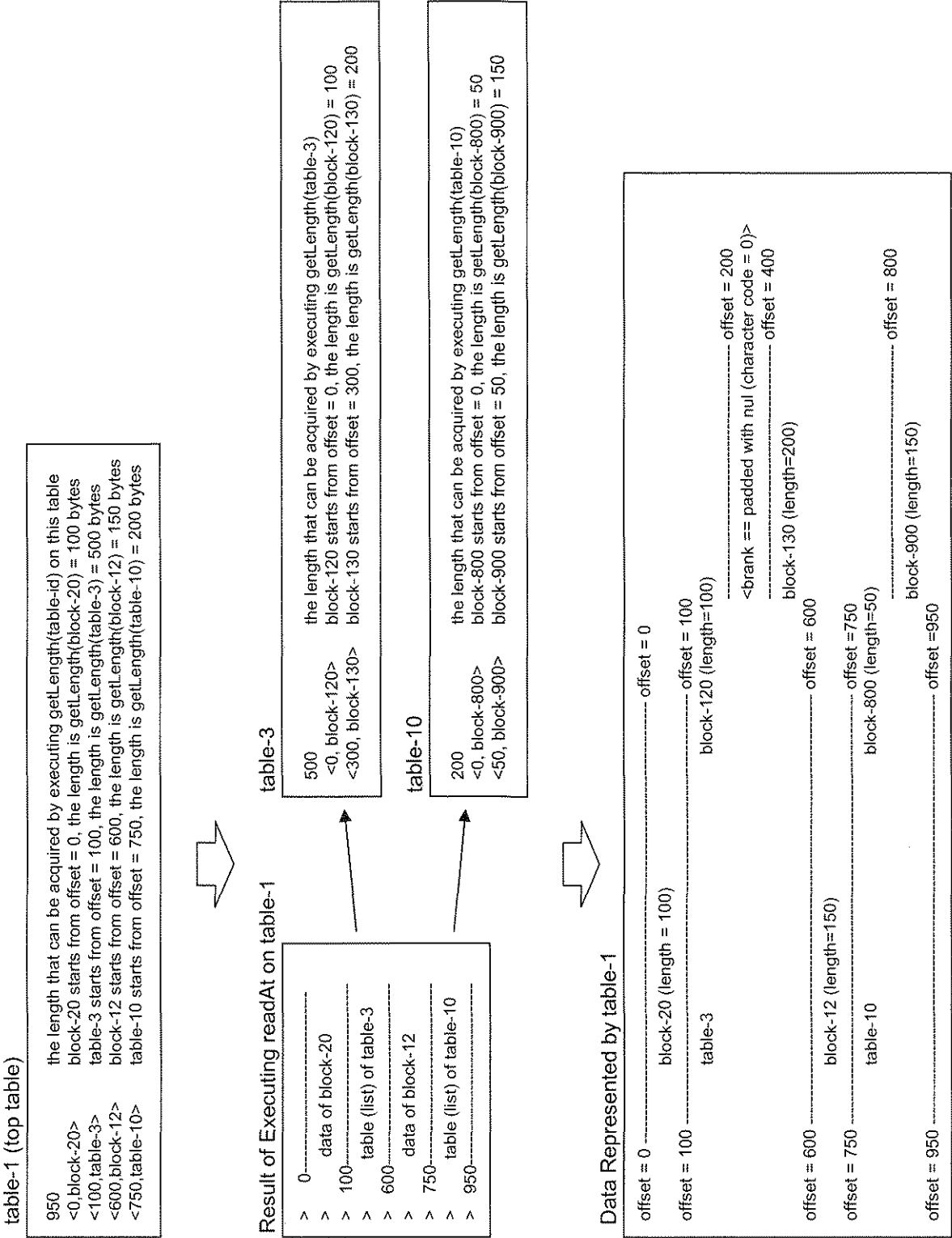
【Figure 5】



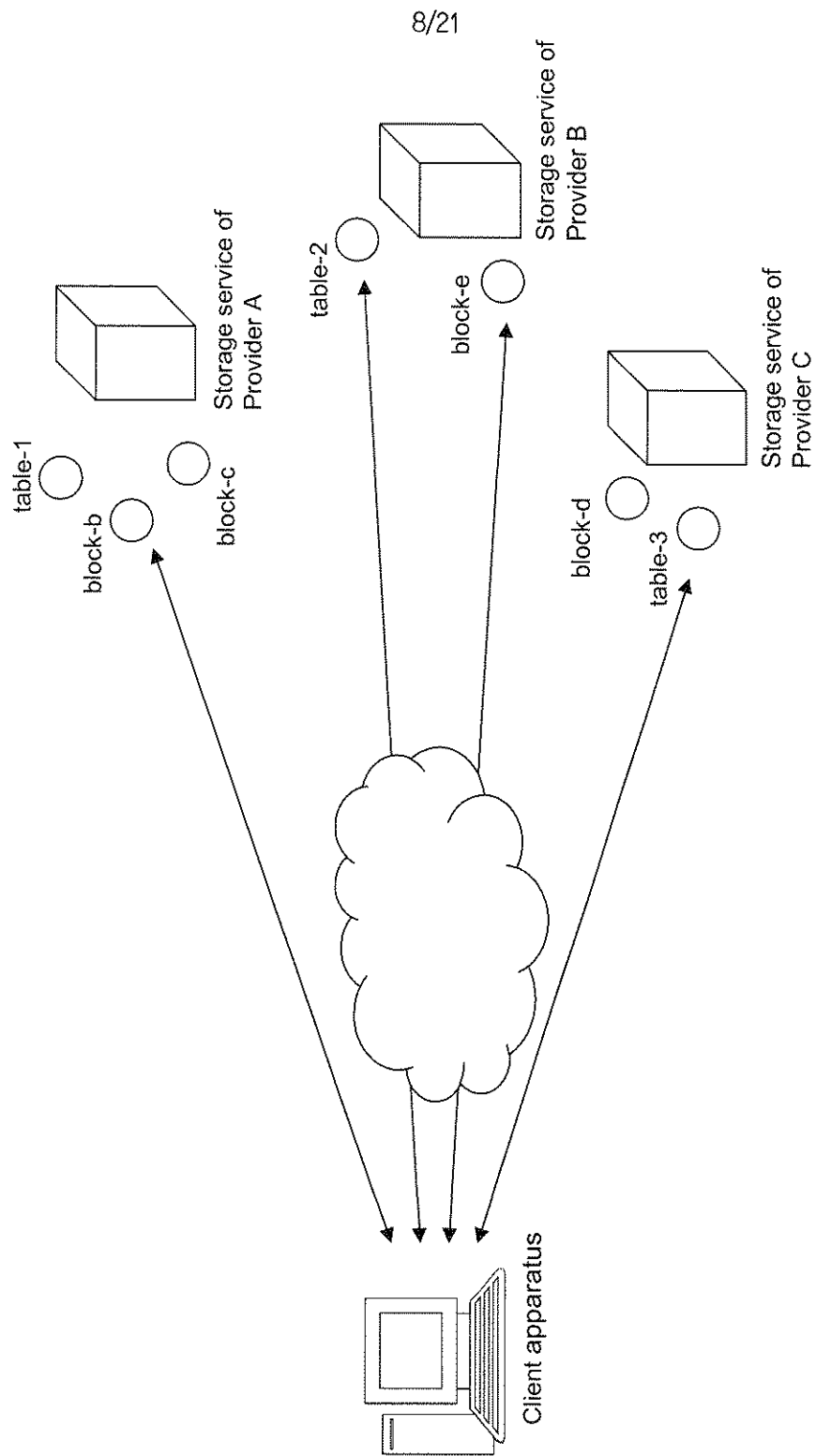


【Figure 6】

【Figure 7】



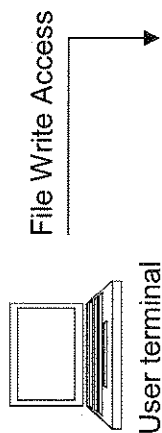
【Figure 8】



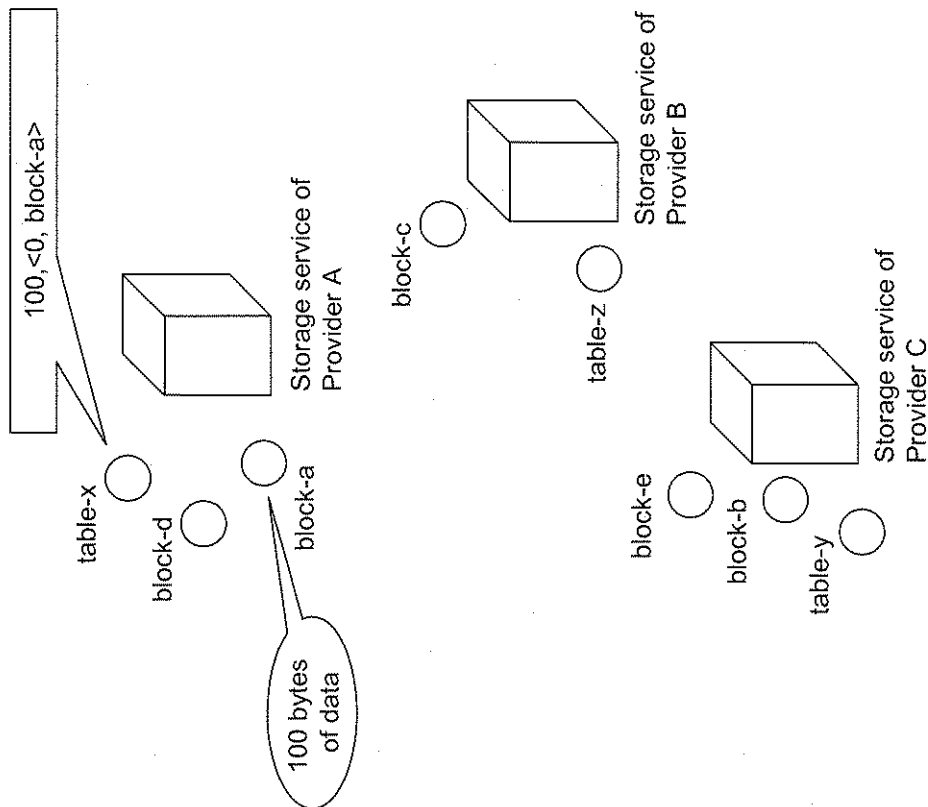
A diagram showing a 'User terminal' (represented by a computer monitor icon) connected to a 'File Read Access' block. An arrow points from the 'User terminal' to the 'File Read Access' block.

-
- 9/21
- Diagram illustrating a storage system with three providers (A, B, and C) and their associated data blocks and tables. The diagram shows how data is distributed across different storage services.
- Storage service of Provider A:**
 - table-1: 650, <0, table-2>, <300, table-3>
 - block-d: 250 bytes of data
 - block-a: 100 bytes of data
 - block-c: 100 bytes of data
 - Storage service of Provider B:**
 - table-3: 350, <0, table-d>, <250, table-e>
 - block-e: 100 bytes of data
 - Storage service of Provider C:**
 - block-b: 100 bytes of data
 - table-2: 300, <0, block-a>, <100, block-b>, <200, block-c>

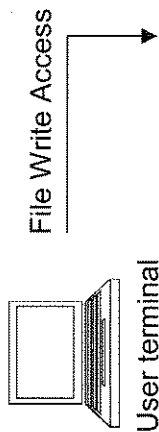
【Figure 10】



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.

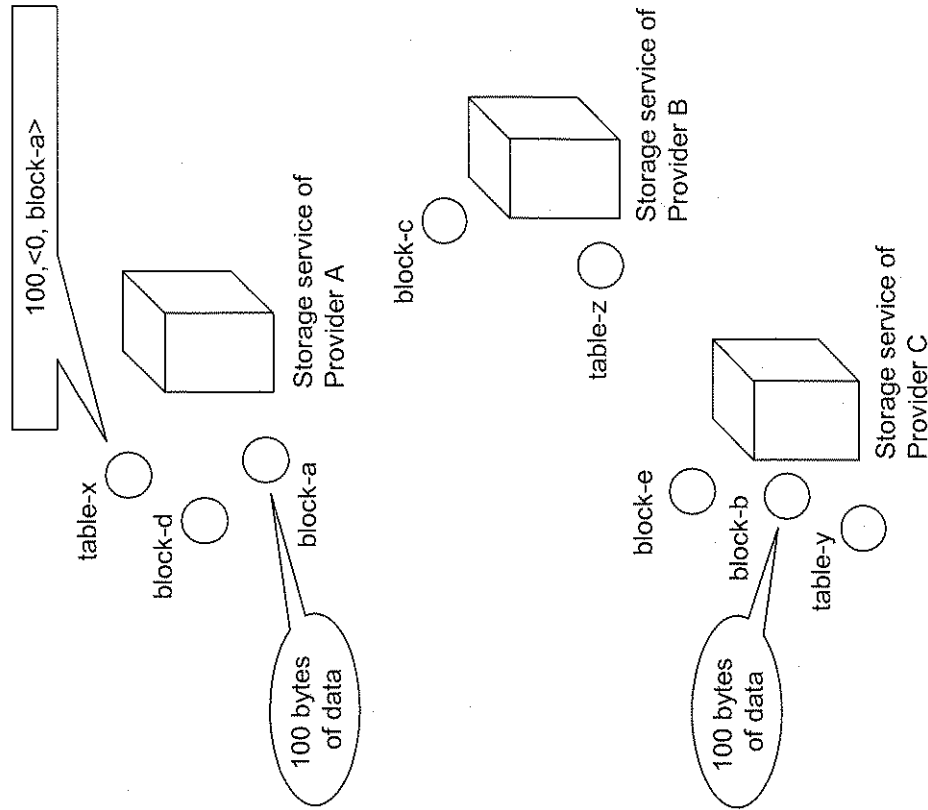
3.1. [WRITE-1] writeAt on file
`writeAt(table-x, 50, data, 0, 150)` is executed on the top table table-x.

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.

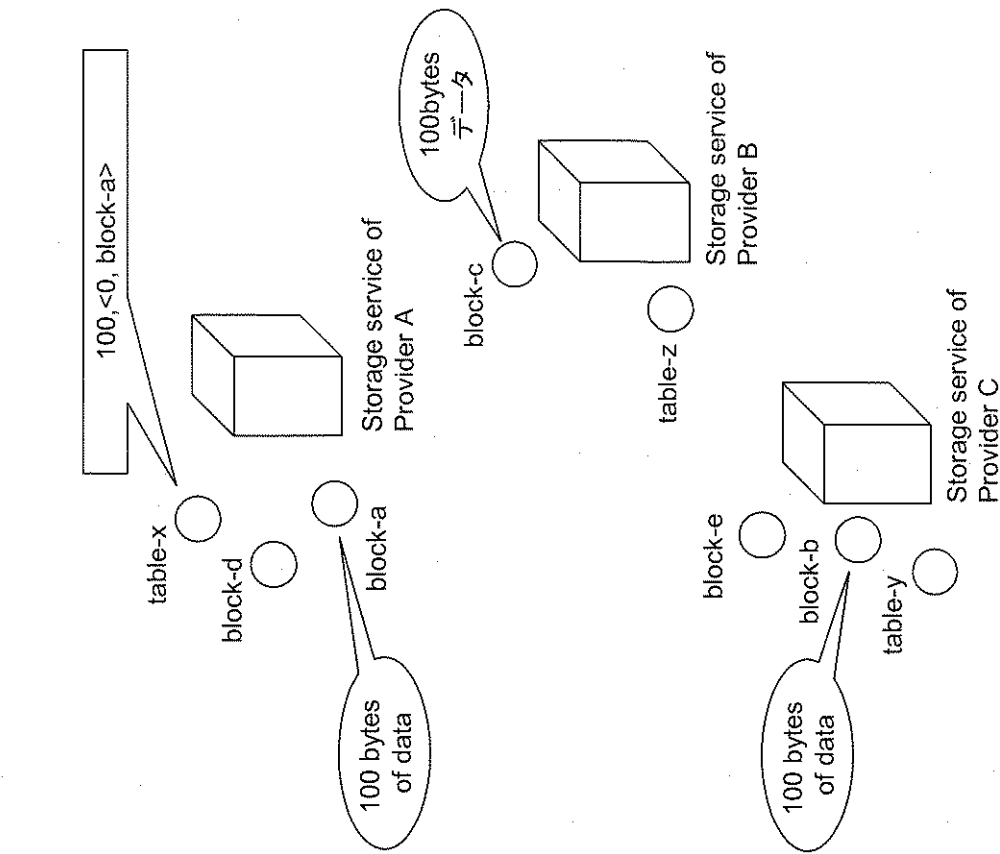
3.2.1. [WRITE-3]
 100 bytes of current data of block-a, orig, is loaded and orig is overwritten at offset = 50 with (data, 0, 50).
 A new block is created by
`block-b = createId()` and
`putContent(block-b, 100, orig)`

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>)`

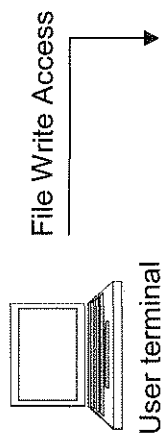
to create a new table.

`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.

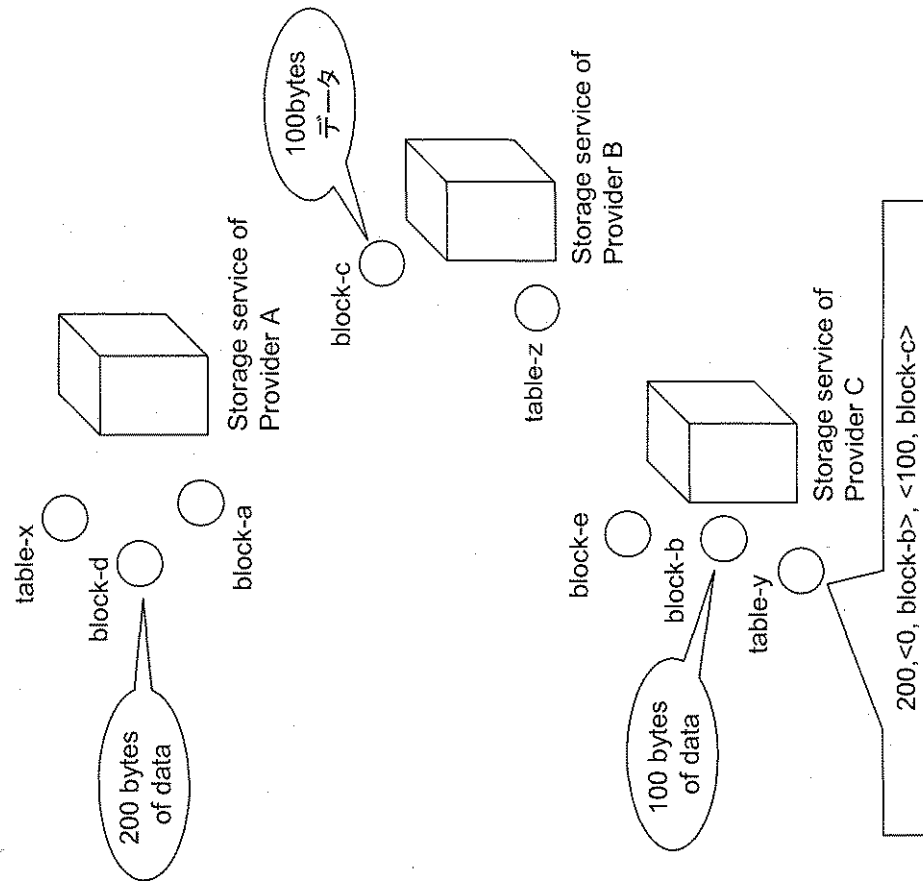
Each type of parameter is updated with

object_offset = 10000 + 200 = 10200,

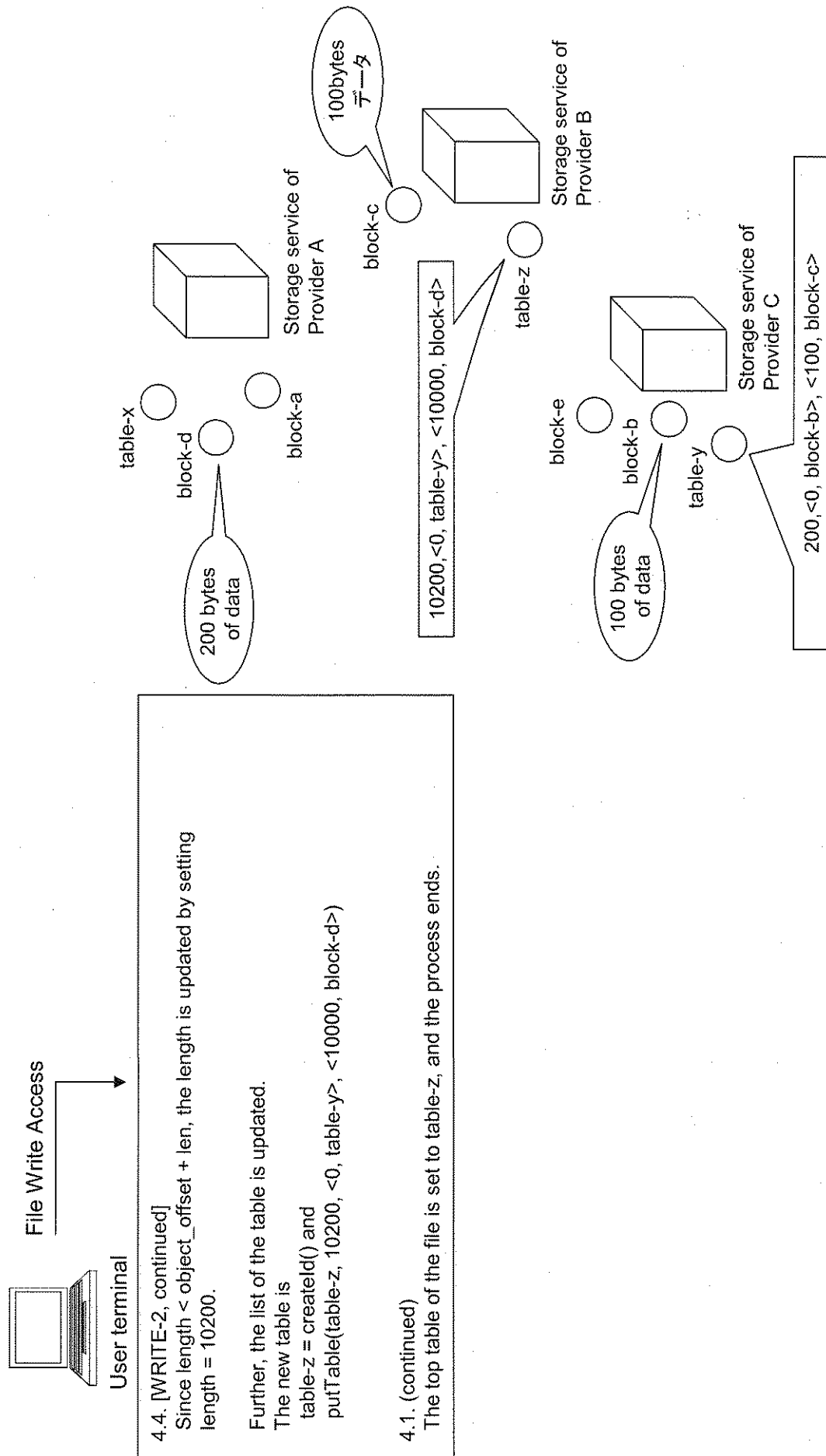
buflen = 200 - 200, and

bufoff = 0 + 200 = 200,

and the loop ends.



【Figure 14】



【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
...	...	...	...

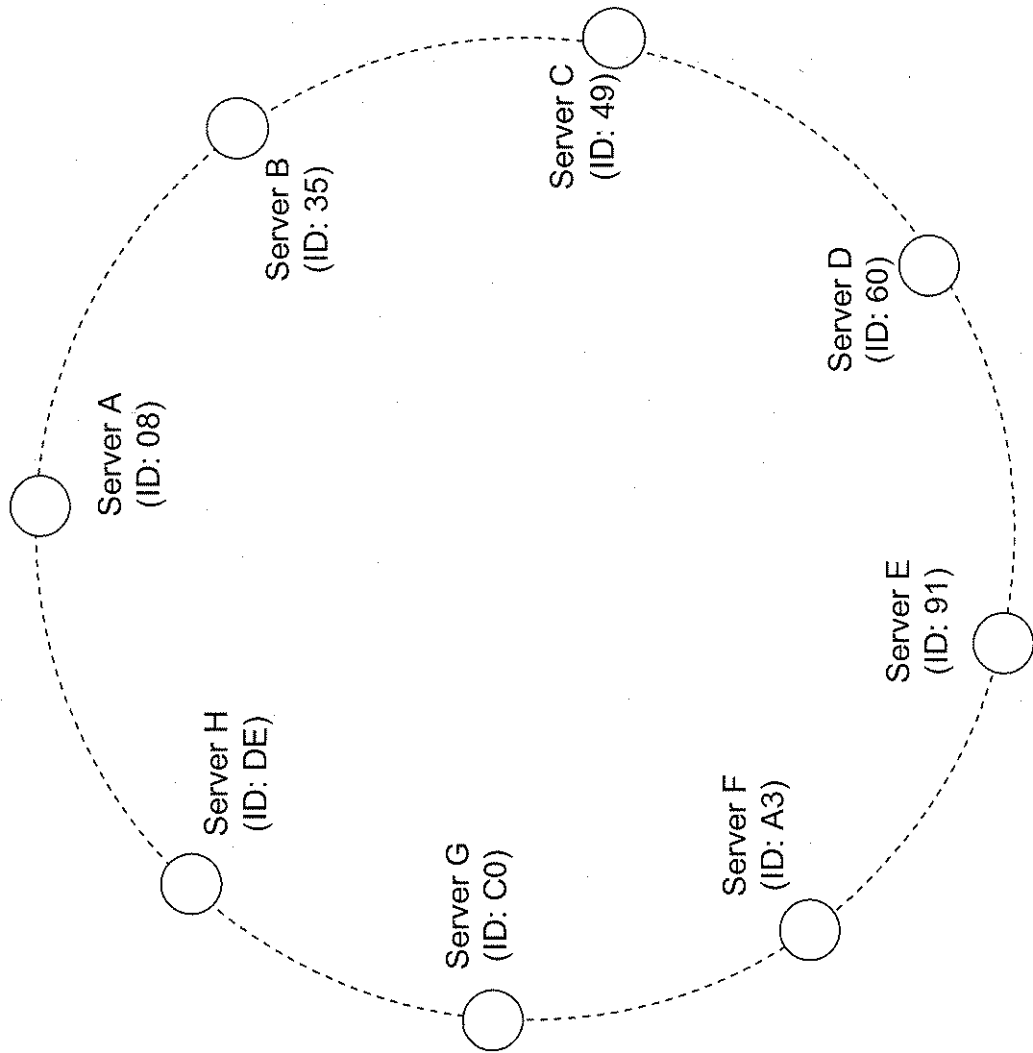
Before Parameter Conversion

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



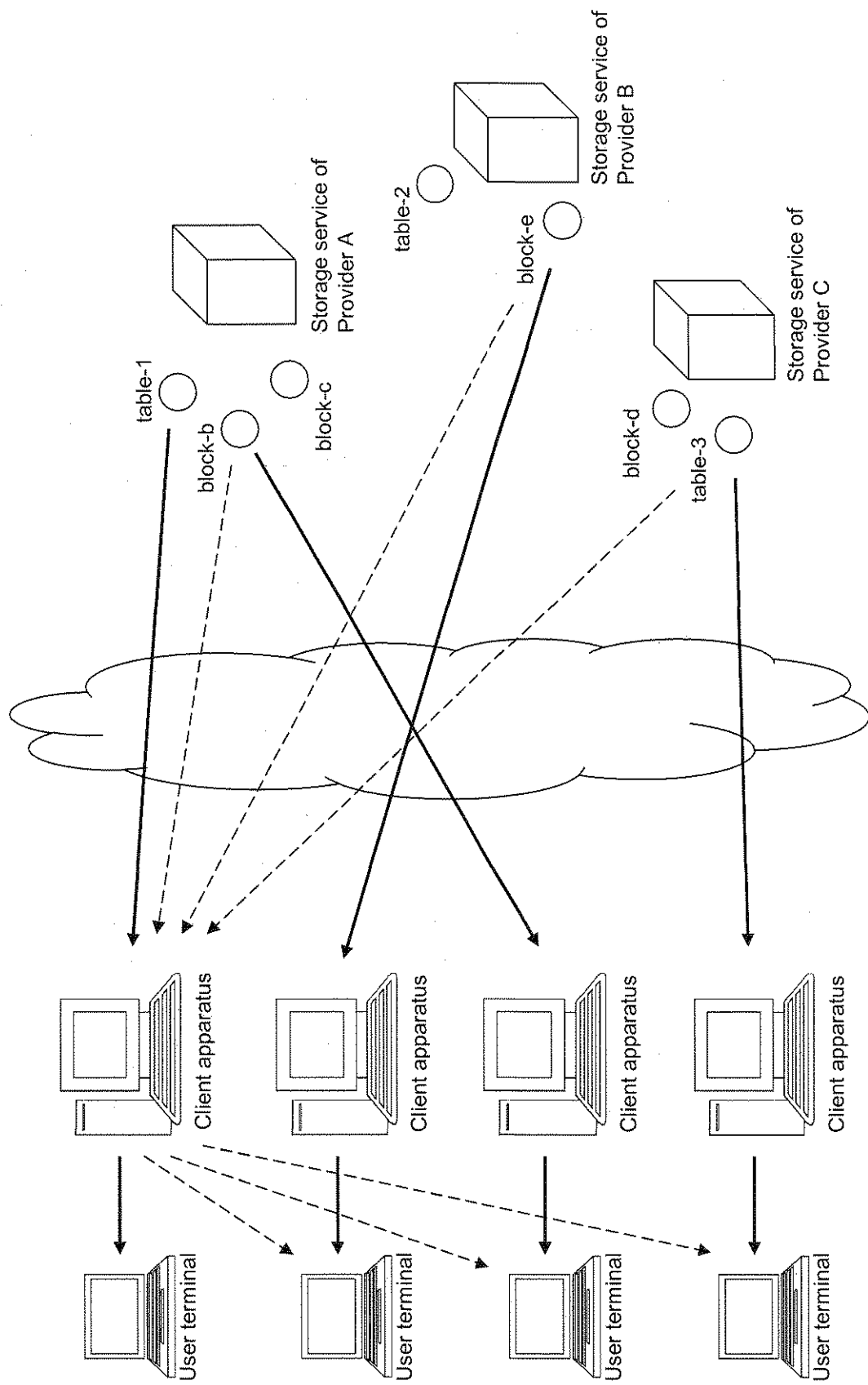
After Parameter Conversion

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

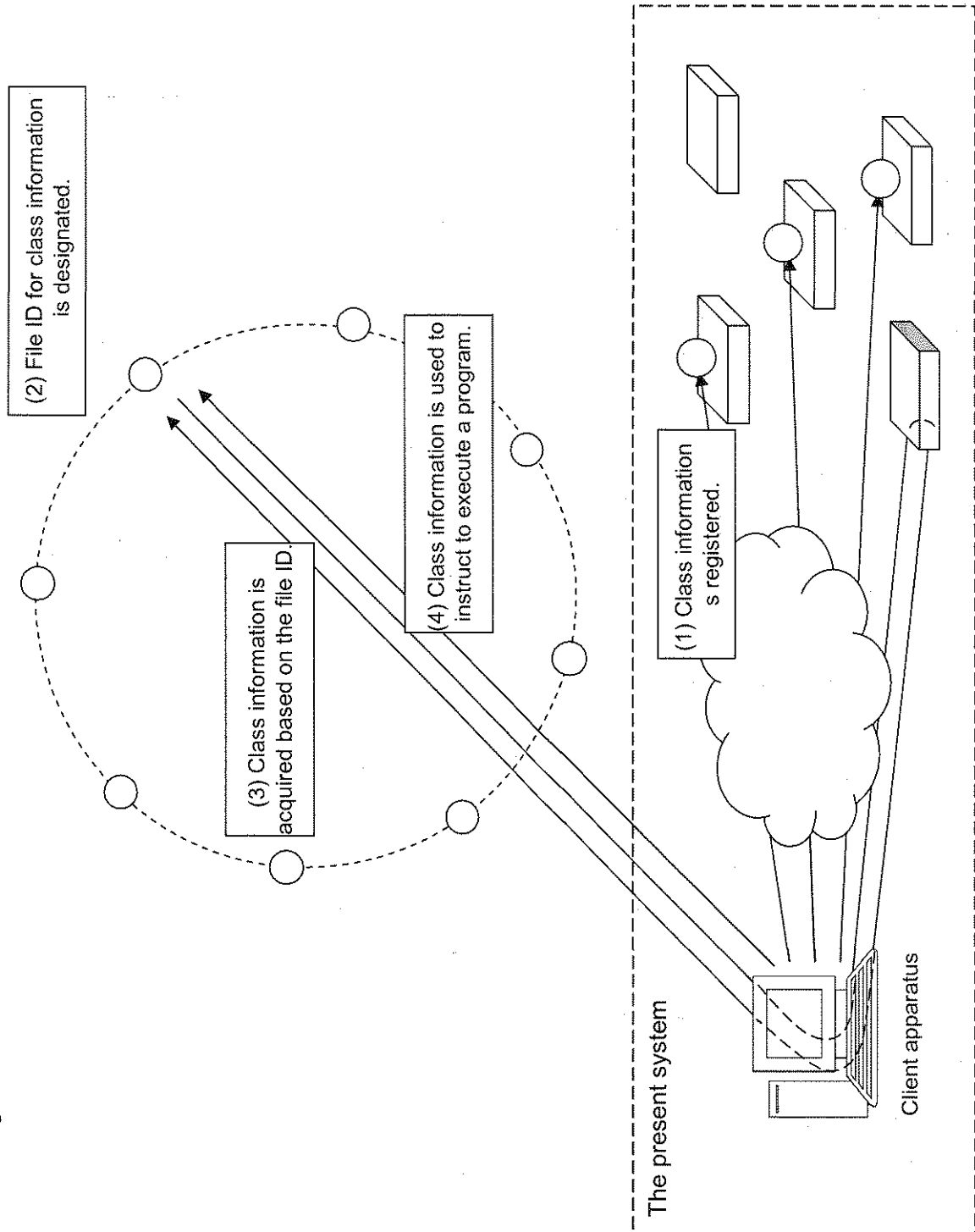


【Figure 17】

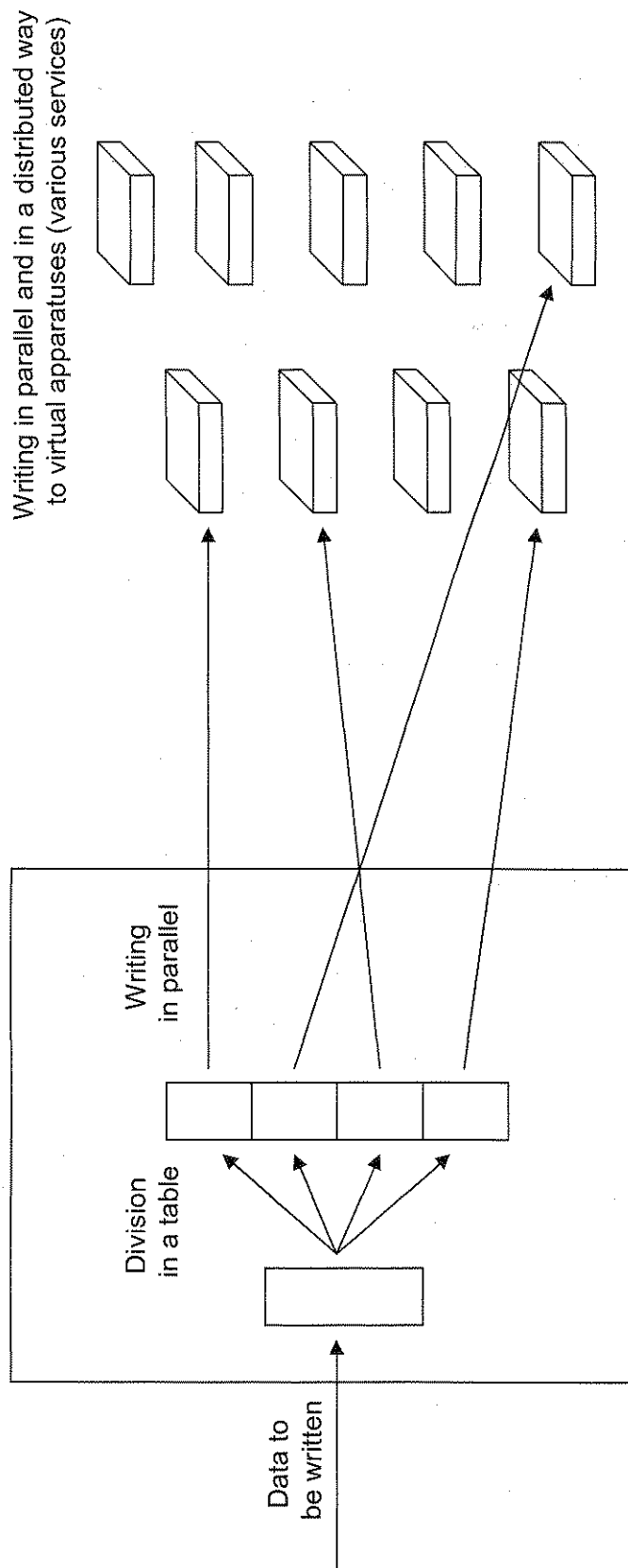
【Figure 18】



【Figure 19】



【Figure 20】



【Figure 21】

