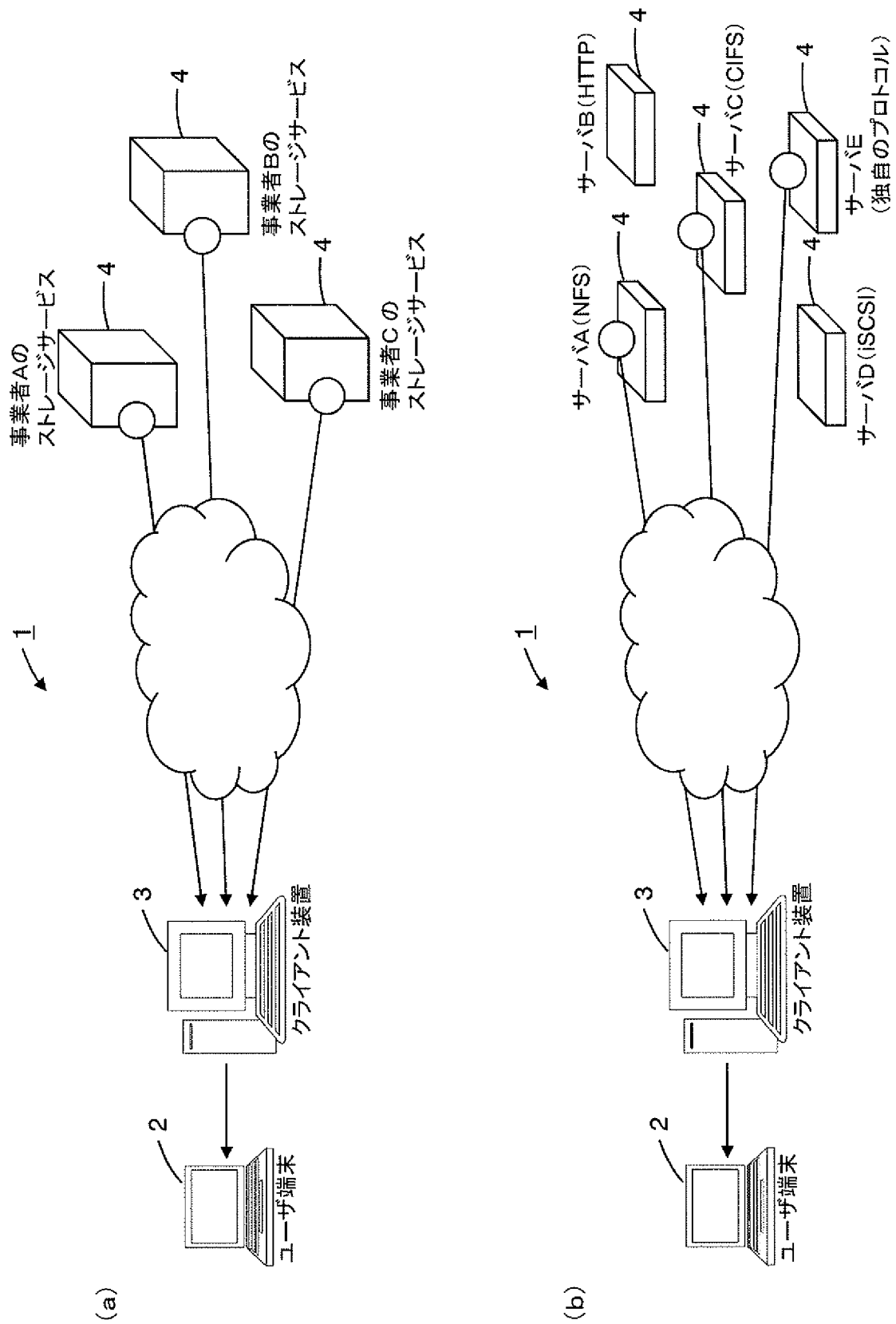
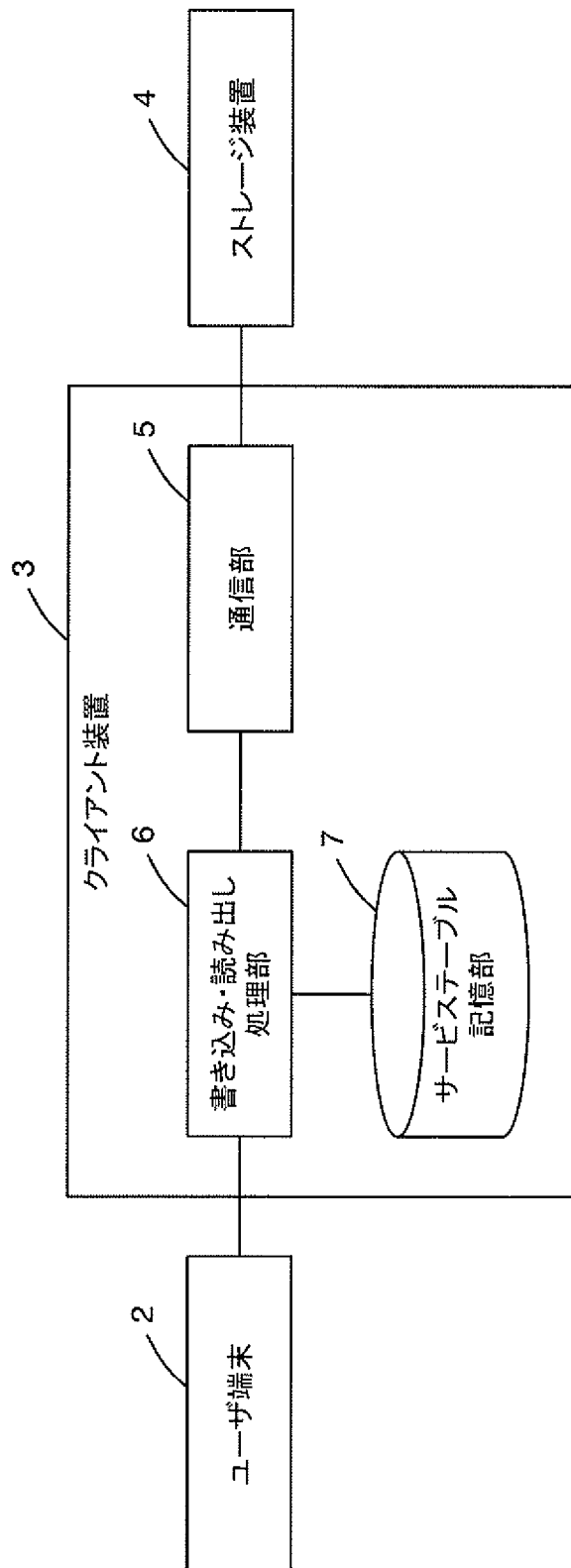


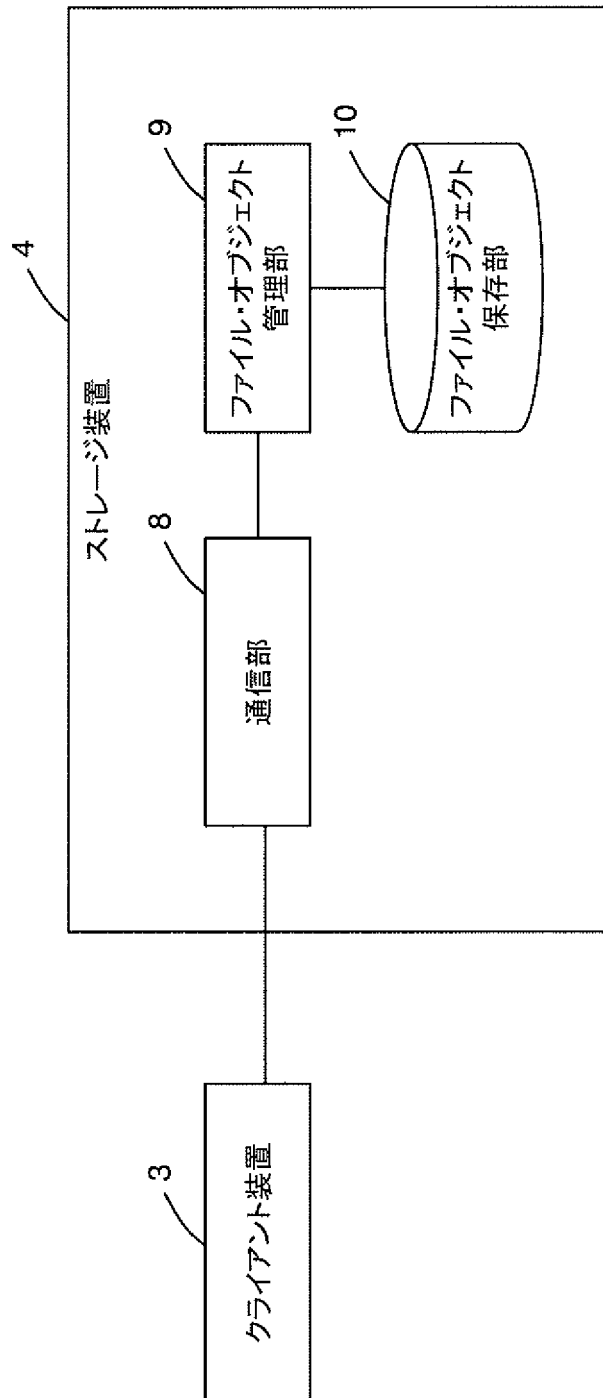
[図1]



[図2]



[図3]



[図4]

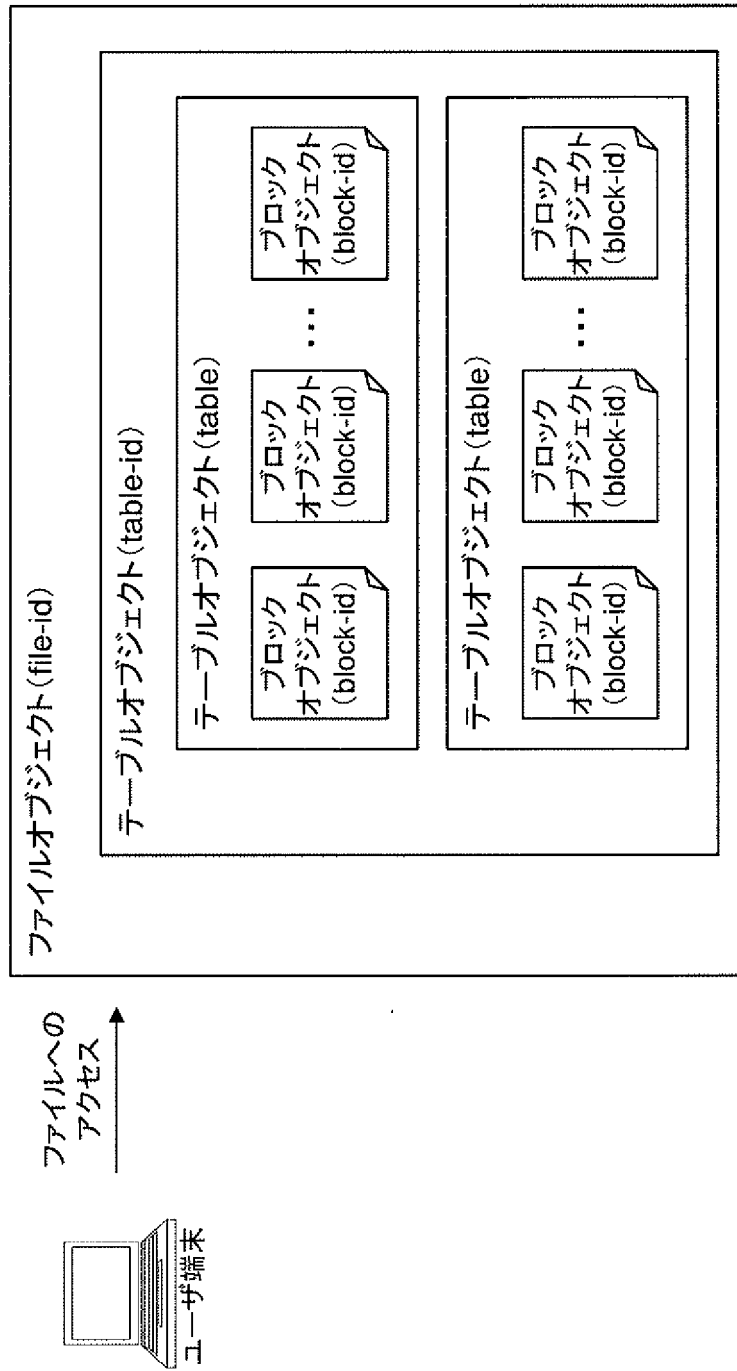
(a) ファイルの作成と削除

機能名	概要	パラメータ等
CREATE	ファイルの作成	なし
DELETE	ファイルの削除	file-id

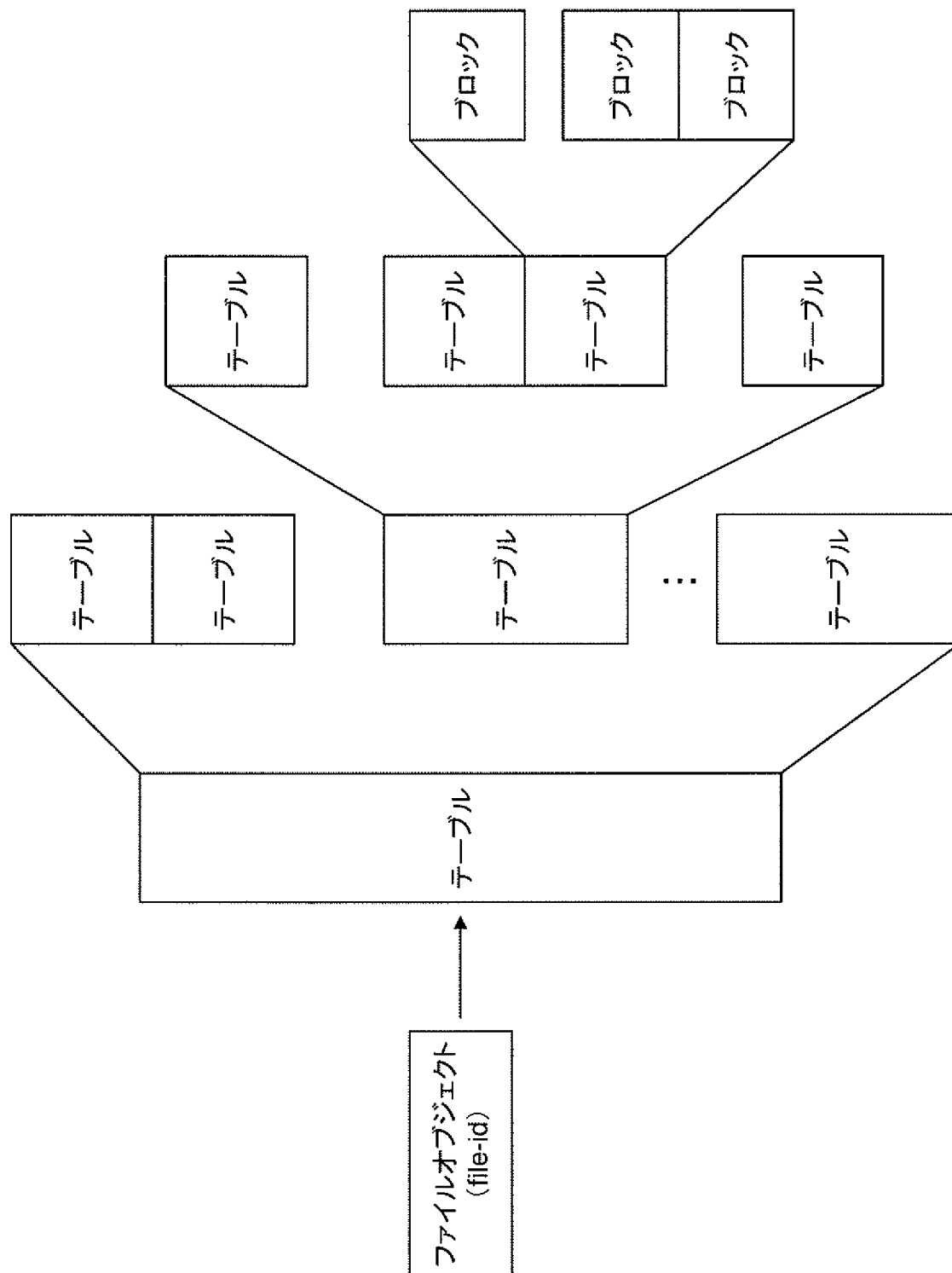
(b) データの読み書き

機能名	概要	パラメータ等
READ	ファイルからのデータの読み出し	file-id, offset, data (lengthを含む)
WRITE	ファイルへのデータの書き込み	file-id, offset, データ領域 (lengthを含む)

[図5]



[図6]



[図7]

table-1 (先頭テーブル)

950	このテーブルに対して getLength(table-id) して取得できる長さ
<0,block-20>	offset = 0 から block-20 が始まる、長さは getLength(block-20) = 100 バイト
<100,table-3>	offset = 100 から table-3 が始まる、長さは getLength(table-3) = 500 バイト
<600,block-12>	offset = 600 から block-12 が始まる、長さは getLength(block-12) = 150 バイト
<750,table-10>	offset = 750 から table-10 が始まる、長さは getLength(table-10) = 200 バイト



table-1をreadAtした結果

> 0-----	block-20のデータ
> 100-----	table-3のテーブル(リスト)
> 600-----	block-12のデータ
> 750-----	table-10のテーブル(リスト)
> 950-----	

table-3

500	getLength(table-3) で取得できる長さ
<0, block-120>	offset=0 から block-120 が始まる、長さは getLength(block-120) = 100 とする。
<300, block-130>	offset=300 から block-130 が始まる、長さは getLength(block-130) = 200 とする。

table-10

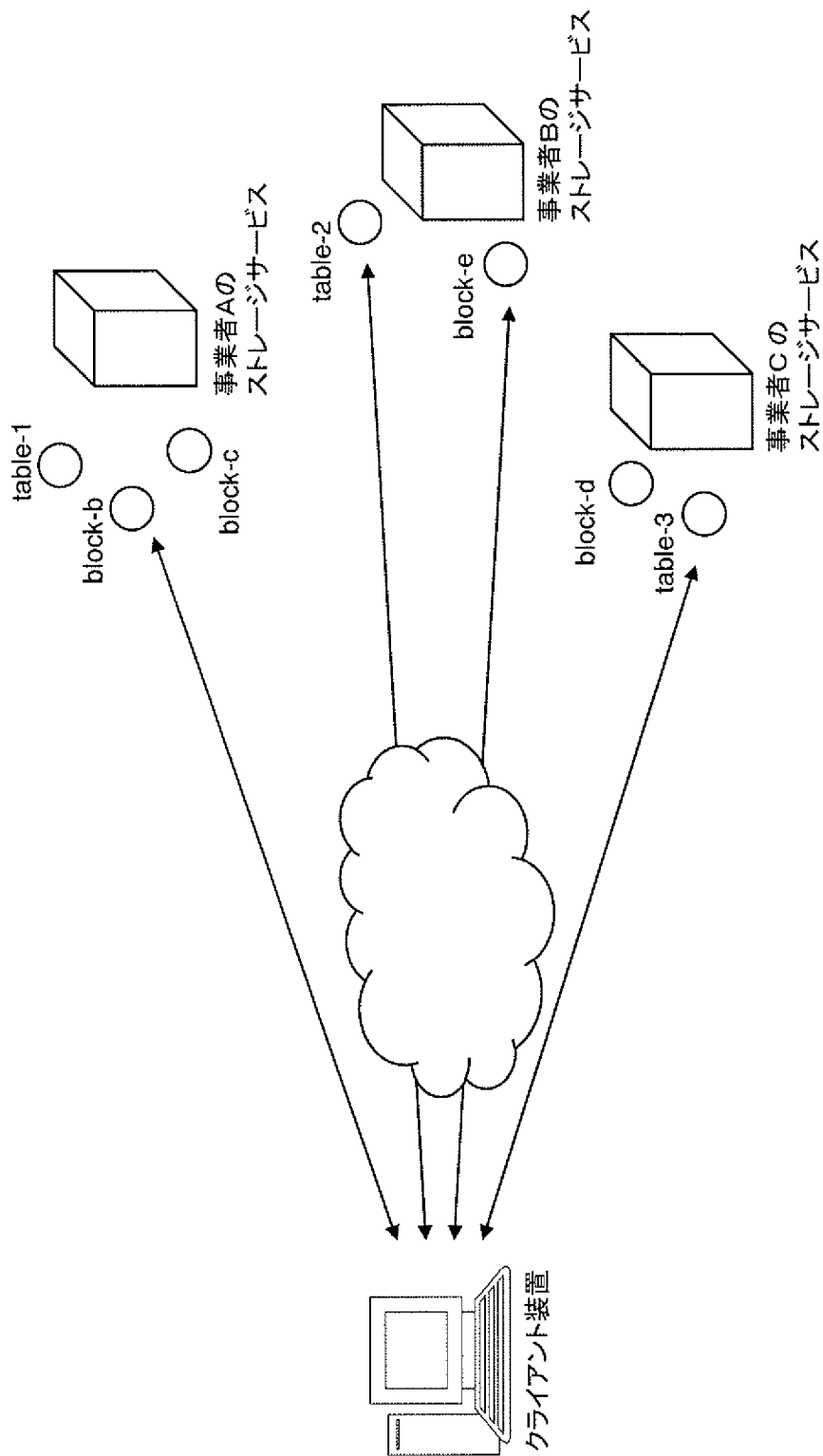
200	getLength(table-10) で取得できる長さ
<0, block-800>	offset=0 から block-800 が始まる。長さは getLength(block-800) = 50 とする。
<50, block-900>	offset=50 から block-900 が始まる。長さは getLength(block-900) = 150 とする。



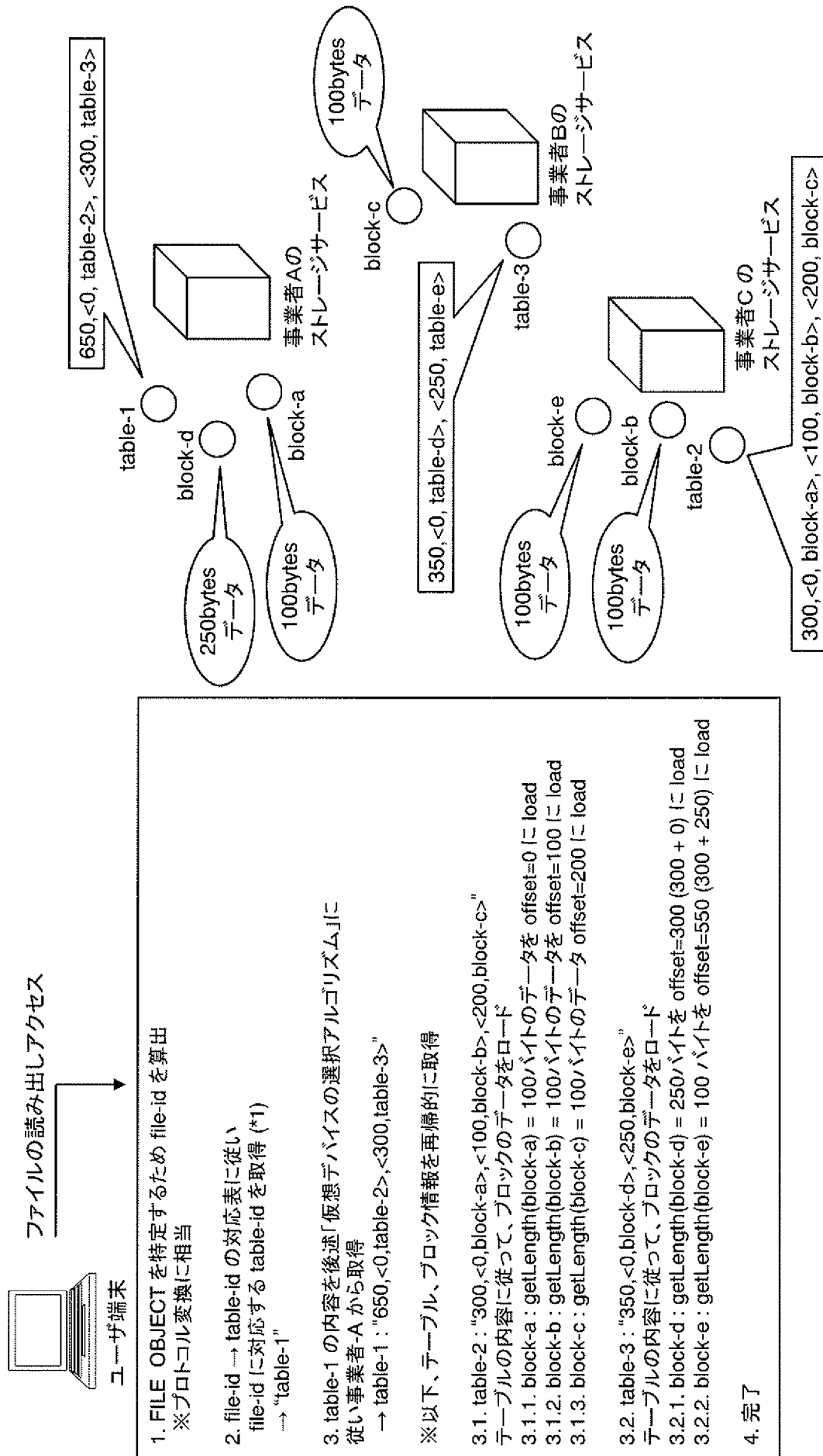
table-1で表現されるデータ

offset = 0 -----	block-20 (length = 100)	offset = 0
offset = 100 -----	table-3	offset = 100
	block-120 (length=100)	
		offset = 200
		<brank == nul (character code = 0) で埋められる>
		offset = 400
	block-130 (length=200)	
offset = 600 -----		offset = 600
	block-12 (length=150)	
offset = 750 -----	table-10	offset = 750
	block-800 (length=50)	
		offset = 800
	block-900 (length=150)	
offset = 950 -----		offset = 950

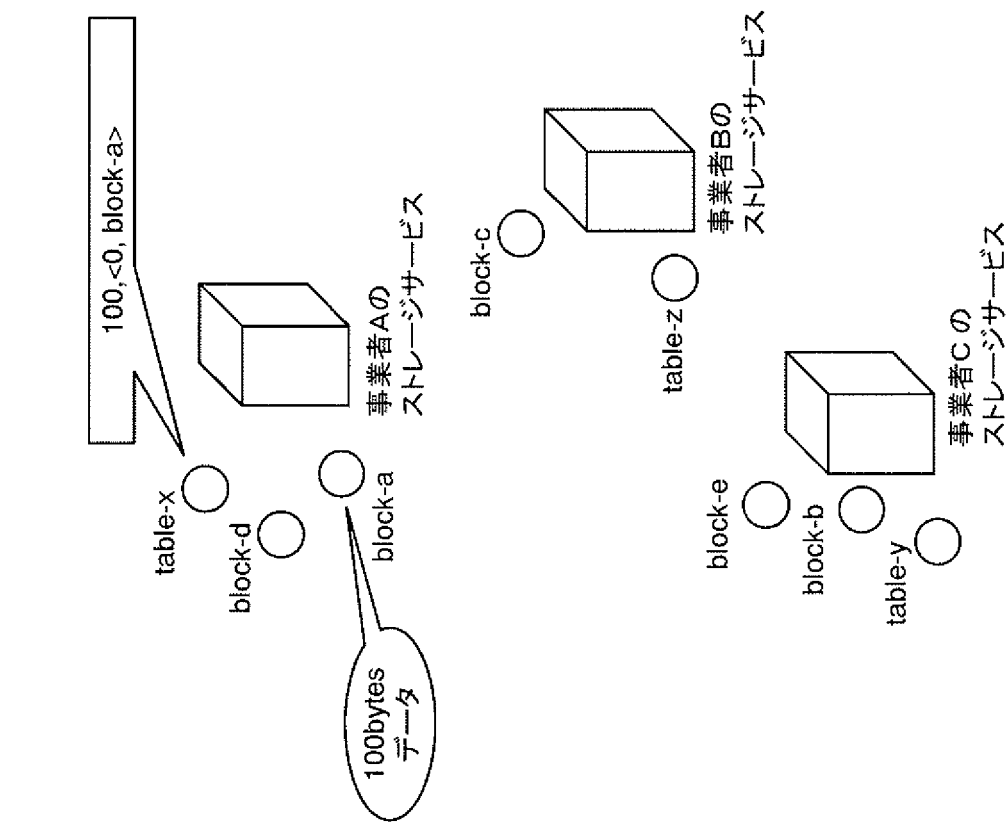
[図8]



[図9]

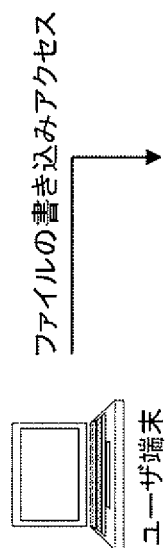
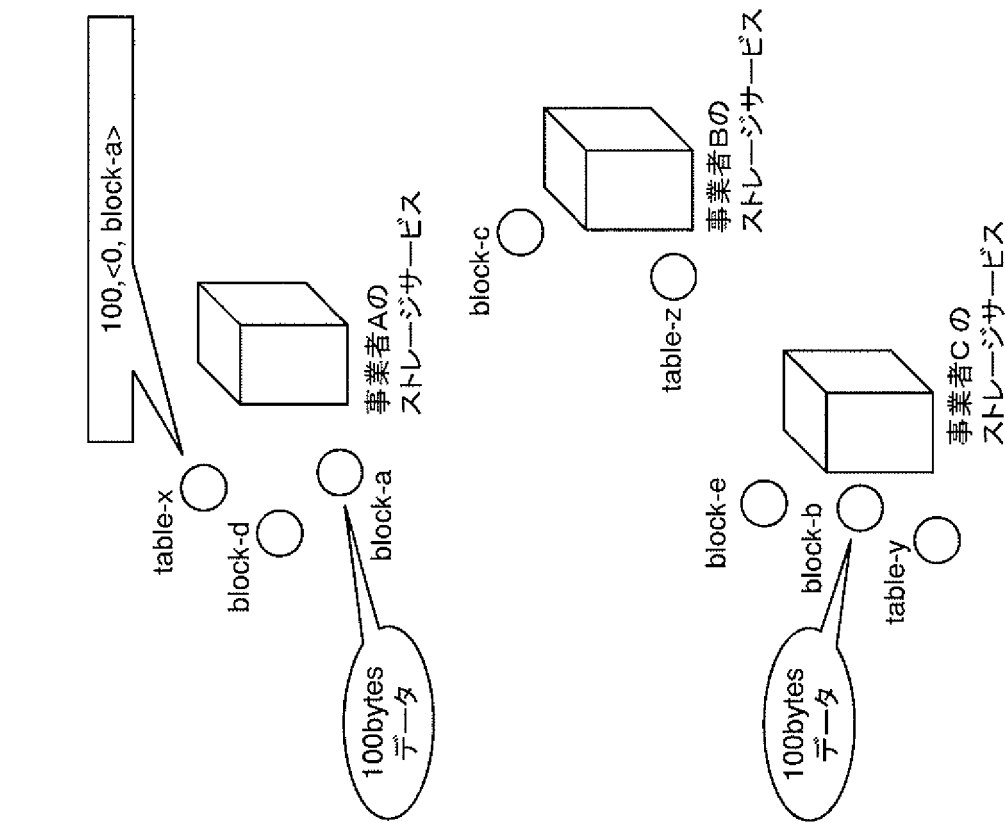


[図10]



1. [CREATE] 新規ファイルを作成
先頭テーブルに EMPTY_TABLE をセット
2. 先頭から 100バイトのデータ data を書き込み
 - 2.1. [WRITE-1] ファイルへのwriteAt
先頭テーブル EMPTY_TABLE に対し
writeAt(EMPTY_TABLE, 0, data, 0, 100) を実行
 - 2.2. [WRITE-2] テーブルへのwriteAt
offset に該当するブロックが存在しないためブロックを作成
block-a = createId()
putContent(block-a, 0 - 0, data, 0, 100)
により block-a のブロックを作成 → リストを更新
writeAt内のメインループを抜けて、
length を 100 に更新後、
table-x = createId()
putTable(table-x, 100, <0, block-a>)
を実行し、table-xを返す
- 2.1. (continued)
上記 table-xを先頭テーブルにセットし、終了

[図11]



3. offset = 50 から 150 バイトのデータ data を書き込み

3.1. [WRITE-1] ファイルへのwriteAt

先頭テーブル table-x に対し

writeAt(table-x, 50, data, 0, 150) を実行

3.2. [WRITE-2] テーブルへのwriteAt

loop-1:

offset=50 に該当するブロックは block-a

writeAt(block-a, 50 - 0, data, 0, 50) を実行

3.2.1. [WRITE-3]

block-a の現在のデータ 100 バイト分origをロードし

origの offset=50 に (data, 0, 50) を上書き

block-b = createId()

putContent(block-b, 100, orig) で新規ブロックを作成

3.2. (continued)

上記 block-bを、新たなブロックとして、リストを更新

→ 新しいリストは<0,block-b>になる (メモリ内)

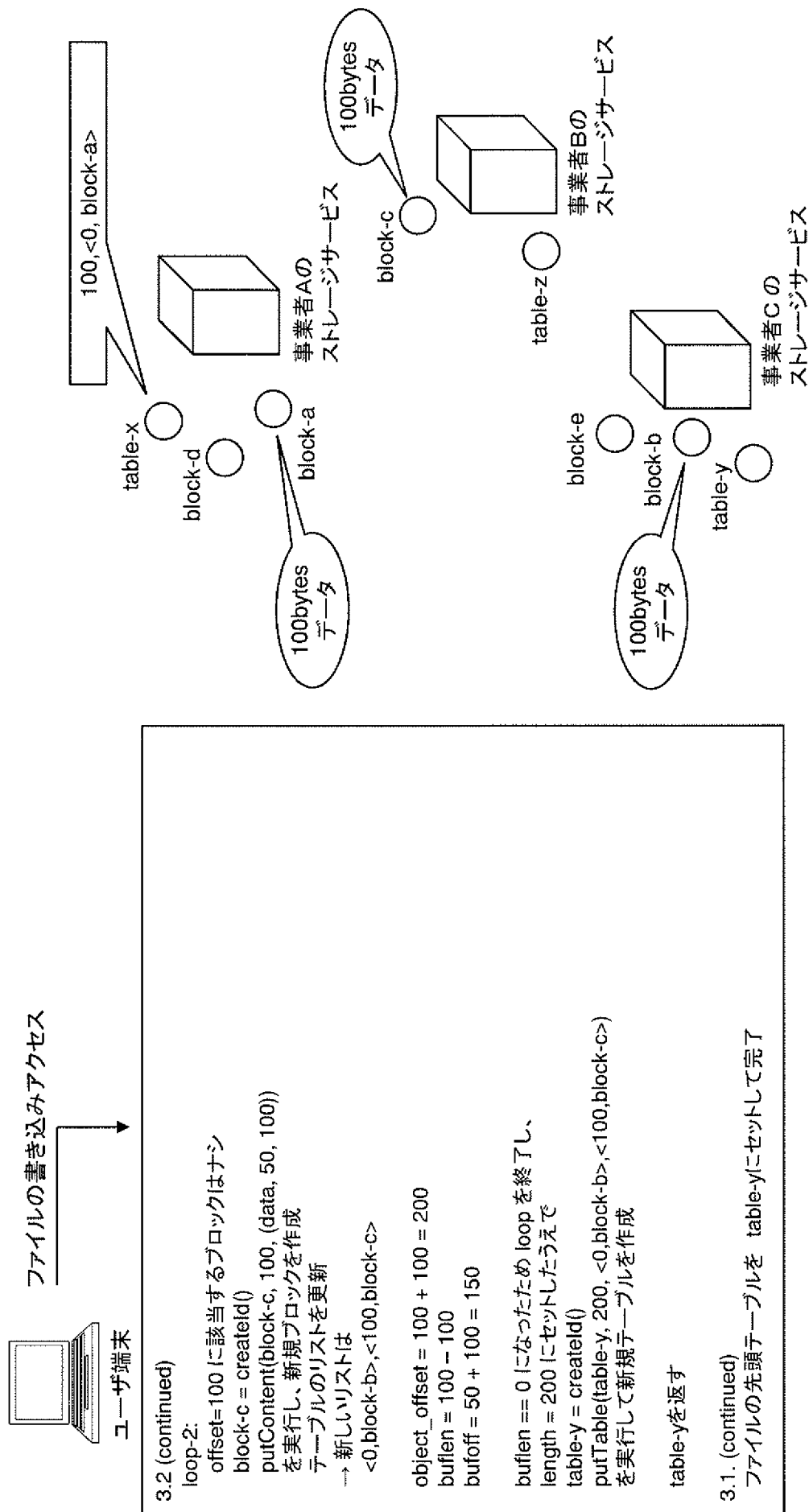
object_offset = 50 + 50 = 100

buflen = 150 - 50 = 100

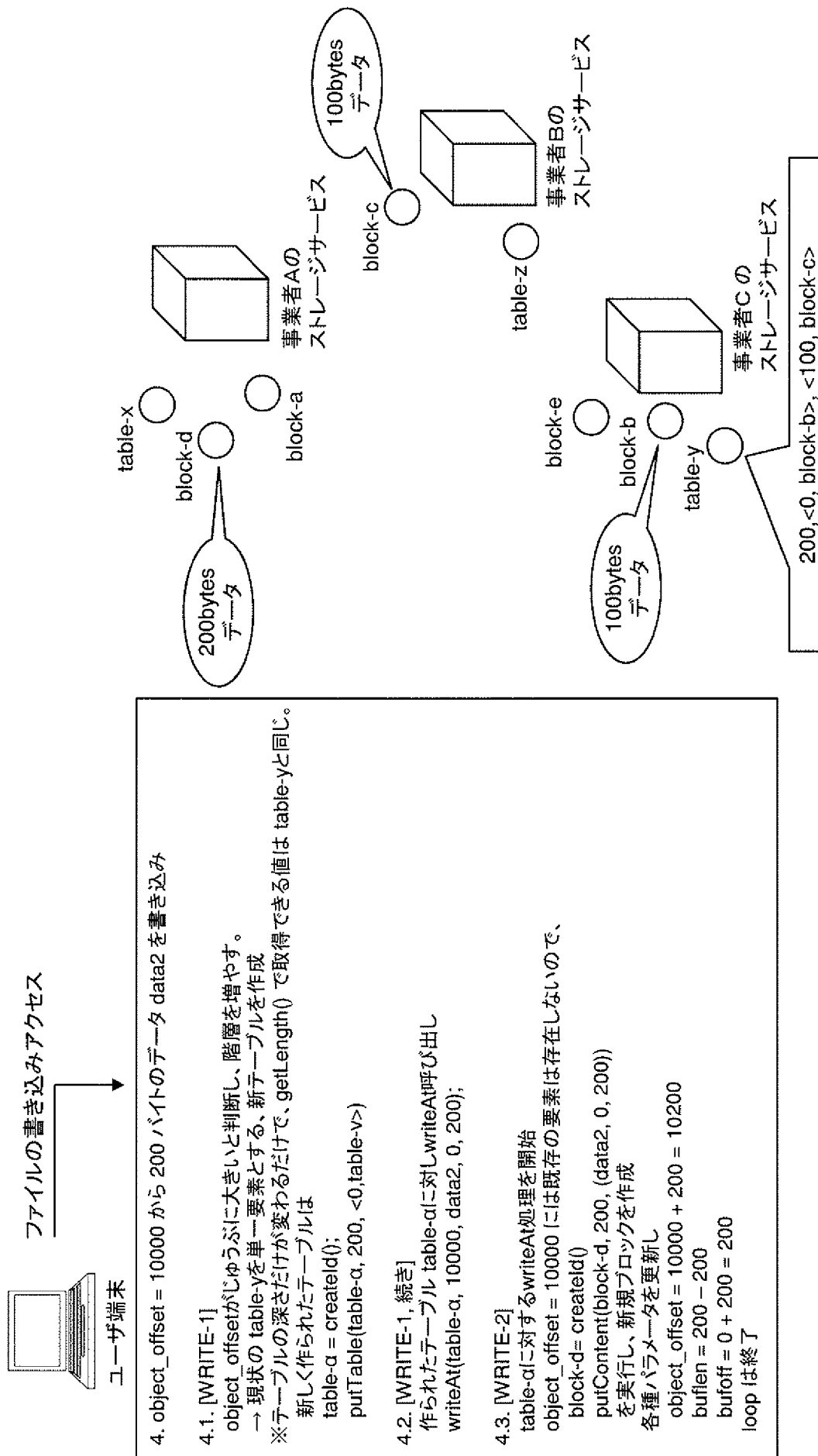
bufoff = 0 + 50 = 50

をセットして次の loop へ

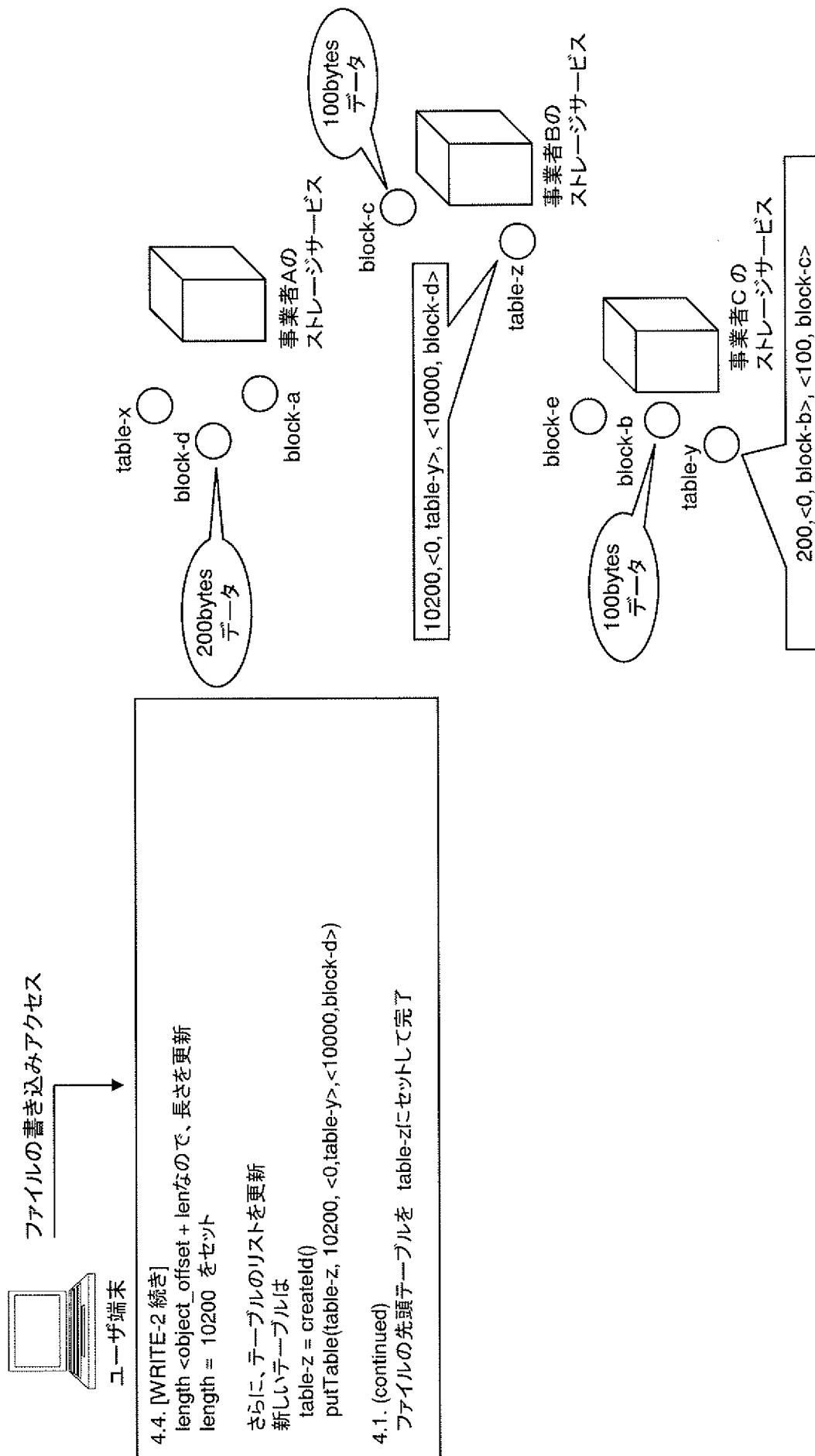
[図12]



[図13]



[図14]



[図15]

静的 (static) なサービステーブル

番号	提供者	アクセス方法	認証
0	事業者A	http://a.com/storage/%o	user-1/pass-1
1	事業者B	http://b.com/disk/%u/%o	user01/pass-2
2	事業者C	nfs://c.com/home/%u/disk/%o	user001/pass-3
...	...	...	...

パラメータ変換前

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



パラメータ変換後

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

[図16]

動的 (dynamic) なサービステーブル

番号	提供者	アクセス方法	認証
01a3	事業者A	http://a.com/storage/%o	user-1/pass-1
25a0	事業者B	http://b.com/disk/%u/%o	user01/pass-2
5c8a	事業者C	nfs://c.com/home/%u/disk/%o	user001/pass-3
...	...	...	...

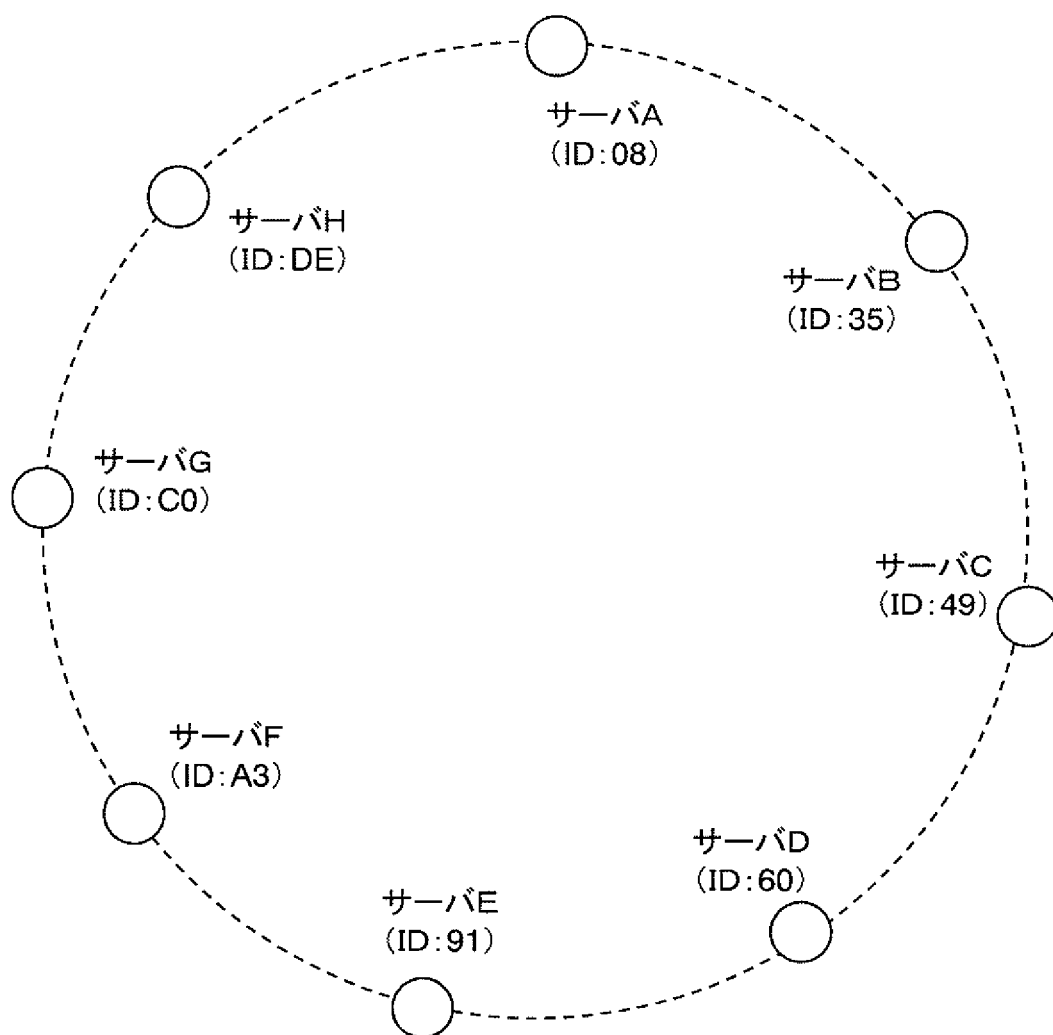
パラメータ変換前

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

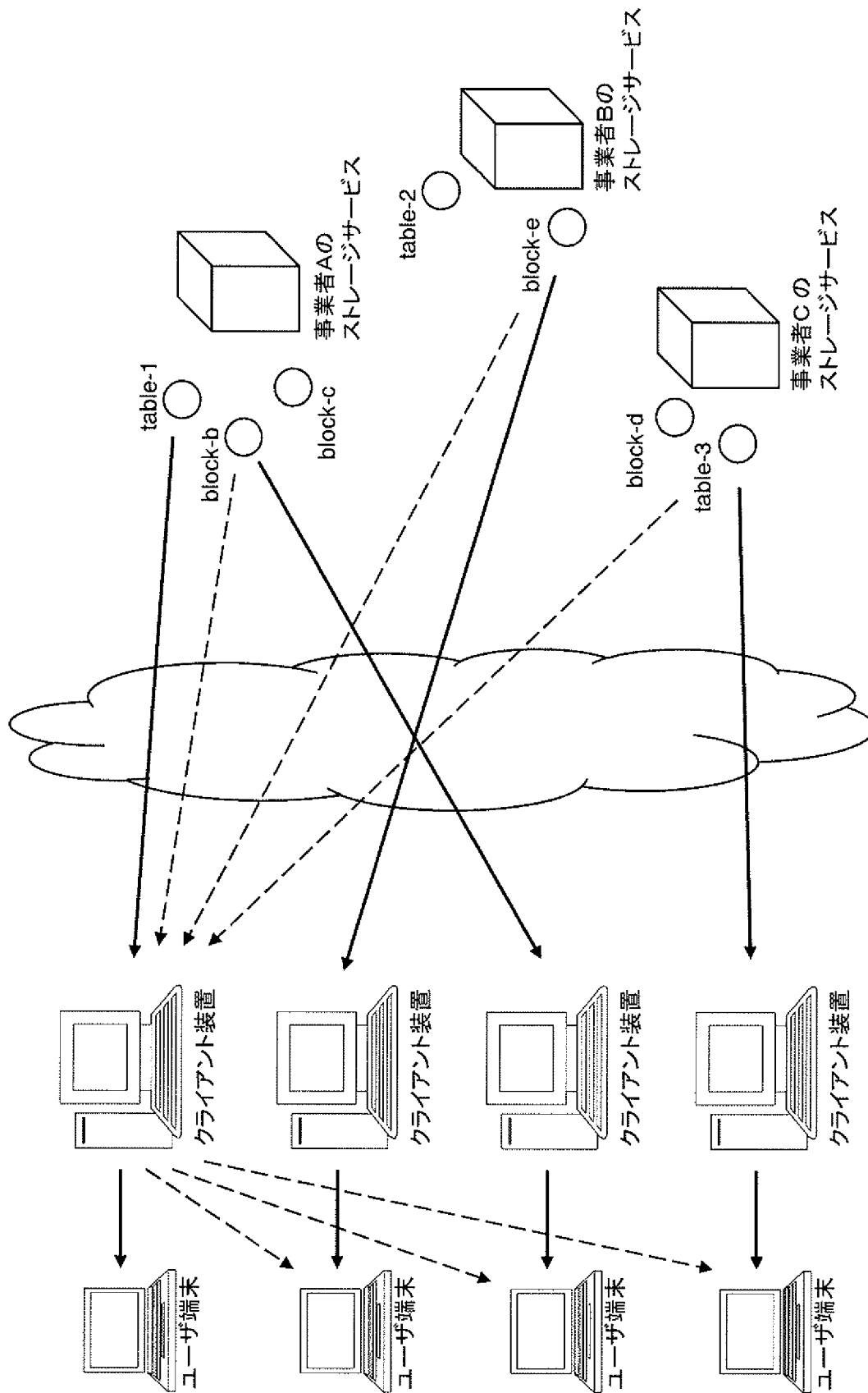
↓
パラメータ変換後

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

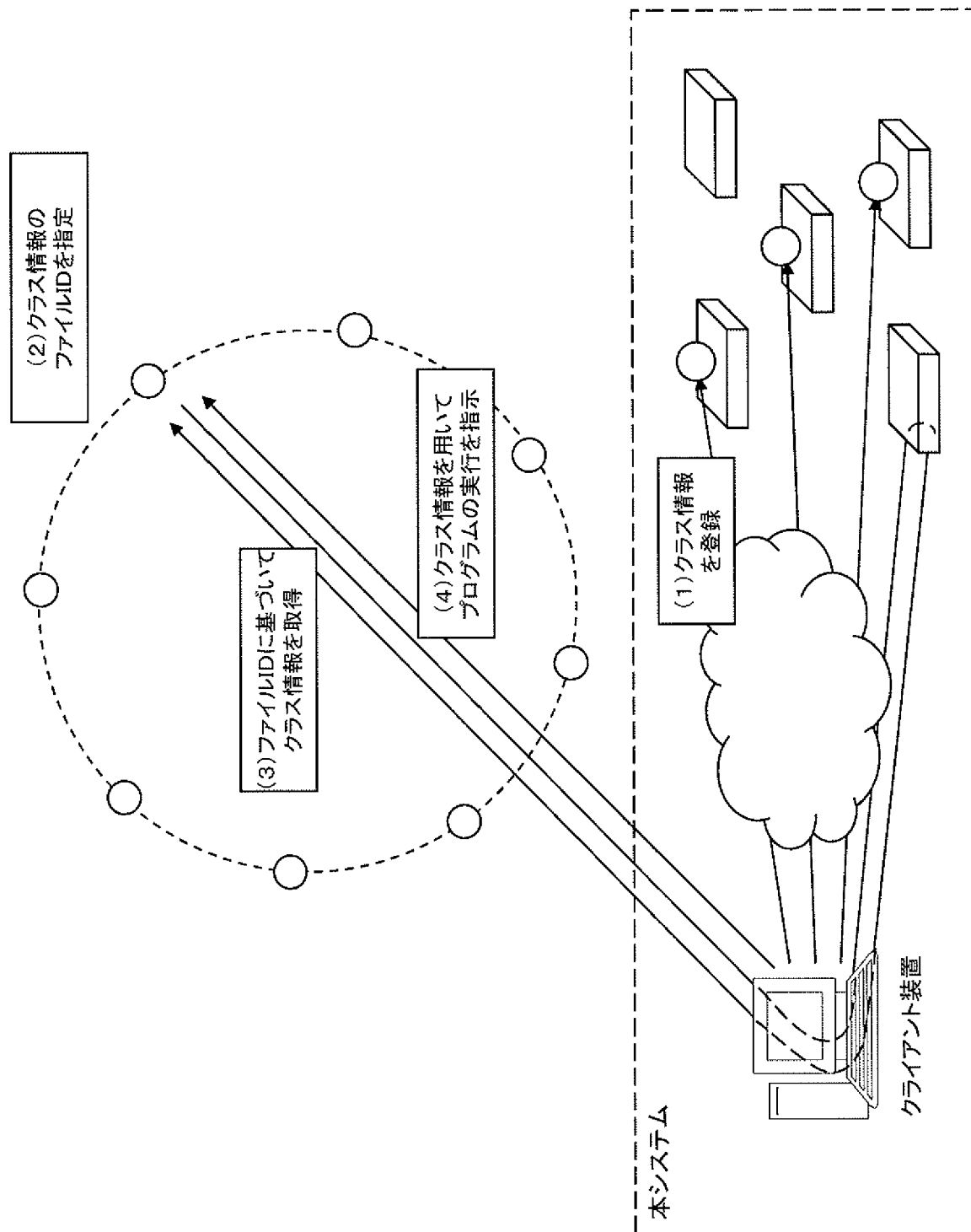
[図17]



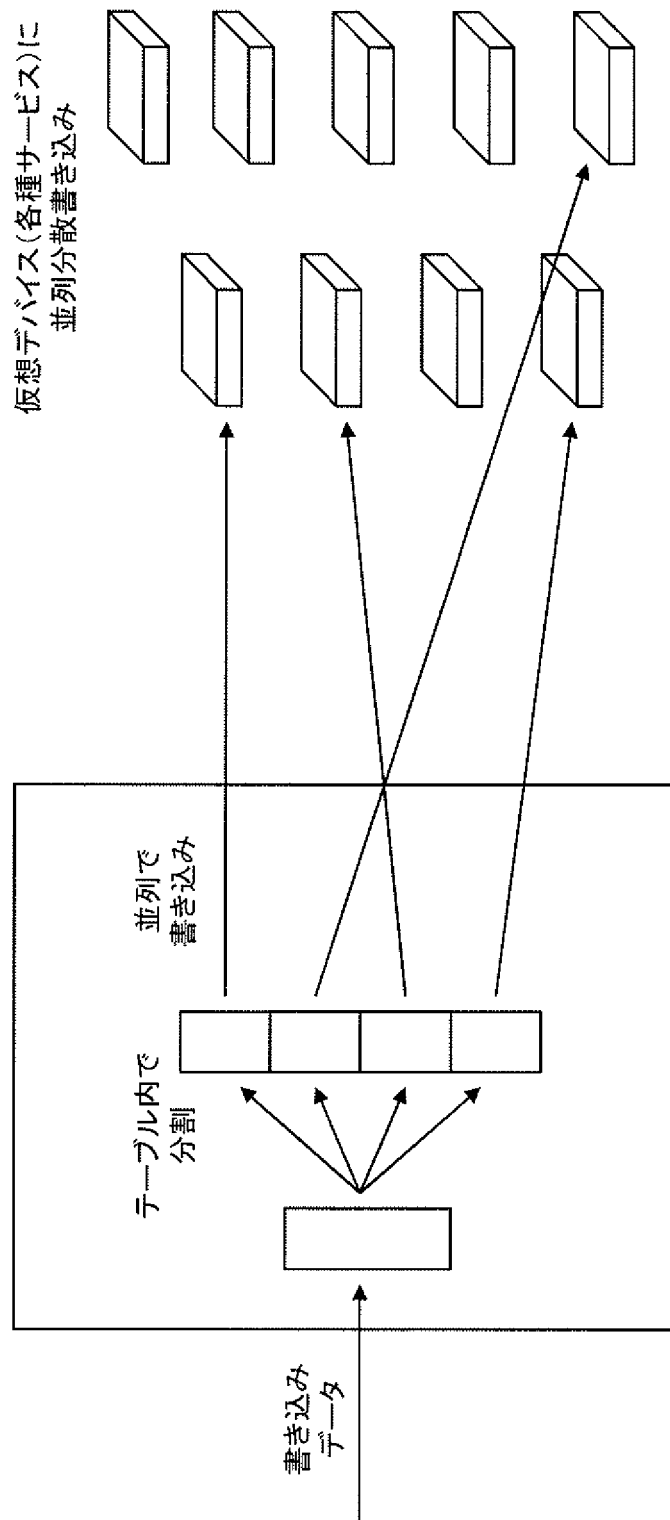
[図18]



[図19]



[図20]



[図21]

