

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4064033号  
(P4064033)

(45) 発行日 平成20年3月19日(2008.3.19)

(24) 登録日 平成20年1月11日(2008.1.11)

(51) Int.Cl.

F 1

GO6F 12/00	(2006.01)	GO6F 12/00	531M
GO6F 3/06	(2006.01)	GO6F 3/06	304F
GO6F 12/16	(2006.01)	GO6F 12/16	310M

請求項の数 6 (全 18 頁)

(21) 出願番号	特願2000-46675 (P2000-46675)
(22) 出願日	平成12年2月24日 (2000.2.24)
(65) 公開番号	特開2001-236253 (P2001-236253A)
(43) 公開日	平成13年8月31日 (2001.8.31)
審査請求日	平成15年12月19日 (2003.12.19)

(73) 特許権者	000005223 富士通株式会社 神奈川県川崎市中原区上小田中4丁目1番 1号
(74) 代理人	100074099 弁理士 大菅 義之
(74) 代理人	100067987 弁理士 久木元 彰
(72) 発明者	長尾 卓巳 神奈川県川崎市中原区上小田中4丁目1番 1号 富士通株式会社内

審査官 高瀬 勤

最終頁に続く

(54) 【発明の名称】複数の記録媒体を利用したデータバックアップ装置およびプログラム記憶媒体

## (57) 【特許請求の範囲】

## 【請求項 1】

第1の記録媒体に格納されている複数のデータセットを複数の第2の記録媒体を利用してバックアップする装置であって、

上記第1の記録媒体から上記複数のデータセットを読み出す読み出し手段と、

その読み出し手段により読み出された複数のデータセットを上記複数の第2の記録媒体に書き込むための複数の書き込み手段を並列的に実行する書き込み手段と、

複数のデータセットを上記複数の第2の記録媒体を利用してバックアップするためには要した時間を各データセットについてそれぞれ検出する検出手段と、

過去に上記検出手段により検出されたデータセット毎のバックアップ時間に基づいて、それらのデータセットを上記書き込み手段により並列的に実行される複数の書き込み処理に割り当てる割当て手段と、を有し、

上記割当て手段は、

上記複数のデータセットを、上記検出手段により検出されたバックアップ時間の長い順に、上記複数の第2の記録媒体のそれぞれに順番に割り当てる第1の手順、

上記第1の手順に続いて、上記複数のデータセットを、上記検出手段により検出されたバックアップ時間の長い順に、上記第1の手順とは逆の順番で上記複数の第2の記録媒体のそれぞれに割り当てる第2の手順、

全データセットが割り当られるまで上記第1および第2の手順を交互に繰り返す第3の手順、

10

20

上記第3の手順の後に、上記第1の手順の1つ飛ばしの順番に、上記第2の記録媒体の書込み実行順序を反転させる第4の手順、を実行する  
ことを特徴とするバックアップ装置。

【請求項2】

請求項1に記載のバックアップ装置であって、  
上記割当て手段は、上記並列的に実行される複数の書込み処理の処理時間が均一または略均一になるように、上記複数のデータセットをそれらの書込み処理に割り当てる  
ことを特徴とするバックアップ装置。

【請求項3】

請求項1に記載のバックアップ装置であって、  
上記割当て手段は、  
過去にバックアップ処理が実行されたデータセットについて、過去に上記検出手段により検出されたバックアップ時間に基づいて、次回のバックアップ処理に要するであろう予測時間を算出する第1の算出手段と、  
過去にバックアップ処理が実行されていないデータセットについて、そのデータセットのデータ量に基づいて、次回のバックアップ処理に要するであろう予測時間を算出する第2の算出手段を含み、

上記第1および第2の算出手段により算出される各データセットについての予測時間に基づいて、それらのデータセットを上記並列的に実行される複数の書込み処理に割り当てる  
ことを特徴とするバックアップ装置。

【請求項4】

第1の記録媒体に格納されている複数のデータセットを複数の第2の記録媒体を利用してバックアップする装置であって、  
上記第1の記録媒体から上記複数のデータセットを読み出す読み出し手段と、  
その読み出し手段により読み出された複数のデータセットを上記複数の第2の記録媒体に書き込むための複数の書込み処理を並列的に実行する書込み手段と、  
上記複数のデータセットの各データ量を検出する検出手段と、  
その検出手段により検出されたデータセット毎のデータ量に基づいて、それらのデータセットを上記書込み手段により並列的に実行される複数の書込み処理に割り当てる割当て手段と、を有し、

上記割当て手段は、  
上記複数のデータセットを、上記検出手段により検出されたデータ量の大きい順に、上記複数の第2の記録媒体のそれぞれに順番に割り当てる第1の手順、  
上記第1の手順に続いて、上記複数のデータセットを、上記検出手段により検出されたデータ量の大きい順に、上記第1の手順とは逆の順番で上記複数の第2の記録媒体のそれぞれに割り当てる第2の手順、  
全データセットが割り当てられるまで上記第1および第2の手順を交互に繰り返す第3の手順、  
上記第3の手順の後に、上記第1の手順の1つ飛ばしの順番に、上記第2の記録媒体の書込み実行順序を反転させる第4の手順、を実行する  
ことを特徴とするバックアップ装置。

【請求項5】

第1の記録媒体に格納されている複数のデータセットを複数の第2の記録媒体を利用してバックアップする方法であって、  
上記第1の記録媒体から上記複数のデータセットを読み出す読み出し手順と、  
読み出された複数のデータセットを上記複数の第2の記録媒体に書き込むための複数の書込み処理を並列的に実行する書込み手順と、  
複数のデータセットを上記複数の第2の記録媒体を利用してバックアップするためには  
した時間を各データセットについてそれぞれ検出手順と、

10

20

30

40

50

過去に検出されたデータセット毎のバックアップ時間に基づいて、それらのデータセットを上記書込み手順により並列的に実行される複数の書き込み処理に割り当てる割当て手順、を有し、

上記割当て手順は、

上記複数のデータセットを、上記検出手順により検出されたバックアップ時間の長い順に、上記複数の第2の記録媒体のそれぞれに順番に割り当てる第1の手順、

上記第1の手順に続いて、上記複数のデータセットを、上記検出手順により検出されたバックアップ時間の長い順に、上記第1の手順とは逆の順番で上記複数の第2の記録媒体のそれぞれに割り当てる第2の手順、

全データセットが割り当たるまで上記第1および第2の手順を交互に繰り返す第3の手順、

上記第3の手順の後に、上記第1の手順の1つ飛ばしの順番に、上記第2の記録媒体の書き込み実行順序を反転させる第4の手順、を含む

ことを特徴とするバックアップ方法。

#### 【請求項6】

第1の記録媒体に格納された複数のデータセットを複数の第2の記録媒体を利用してバックアップするために使用するコンピュータを、

上記第1の記録媒体から上記複数のデータセットを読み出す読み出し手段、

その読み出し手段により読み出された複数のデータセットを上記複数の第2の記録媒体に書き込むための複数の書き込み処理を並列的に実行する書き込み手段、

複数のデータセットを上記複数の第2の記録媒体を利用してバックアップするためを要した時間を各データセットについてそれぞれ検出する検出手段、

過去に上記検出手段により検出されたデータセット毎のバックアップ時間に基づいて、それらのデータセットを上記書込み手段により並列的に実行される複数の書き込み処理に割り当てる割当て手段、として機能させるプログラムであって、

上記割当て手段は、

上記複数のデータセットを、上記検出手段により検出されたバックアップ時間の長い順に、上記複数の第2の記録媒体のそれぞれに順番に割り当てる第1の手順、

上記第1の手順に続いて、上記複数のデータセットを、上記検出手段により検出されたバックアップ時間の長い順に、上記第1の手順とは逆の順番で上記複数の第2の記録媒体のそれぞれに割り当てる第2の手順、

全データセットが割り当たるまで上記第1および第2の手順を交互に繰り返す第3の手順、

上記第3の手順の後に、上記第1の手順の1つ飛ばしの順番に、上記第2の記録媒体の書き込み実行順序を反転させる第4の手順、を実行する

ことを特徴とするプログラムを記録したコンピュータ読み取り可能な記録媒体。

#### 【発明の詳細な説明】

##### 【0001】

##### 【発明の属する技術分野】

この発明は、データをバックアップする装置および方法に係わる。特に、複数のデータセットを複数の記録媒体を用いてバックアップする際の効率を向上させる技術に係わる。

##### 【0002】

##### 【従来の技術】

様々な分野において情報化が進み、それに伴って大量のデータを蓄積するための装置が必要不可欠になってきている。また、それらのデータの中には、しばしば重要な情報が含まれているので、大量のデータをバックアップするための技術も実用化されている。

##### 【0003】

大量のデータは、一般に、複数の記憶装置に分散して格納される。この場合、記憶装置としては、たとえば、磁気ディスク装置等のD A S D (Direct Access Storage Device) が用いられる。一方、バックアップ用記憶装置としては、しばしば磁気テープ記録媒体が用

10

20

30

40

50

いられる。そして、格納すべきデータ量が大きい場合には、このバックアップ用の磁気テープ記録媒体も複数必要となる。

#### 【0004】

複数の記録媒体を用いてデータをバックアップする場合は、そのバックアップ時間を短縮するために、通常、それらの記録媒体へデータを書き込むための複数の書き込み処理が並列的に実行される。すなわち、上述の例においては、複数の磁気ディスク装置から読み出されたデータは、複数の磁気テープ記録媒体に並列に書き込まれる。この並列処理は、例えば、磁気ディスク装置からデータ読み出して所定の磁気テープ記録媒体に書き込むジョブを磁気ディスク装置毎に（あるいは、論理ボリューム毎に）作成し、それらのジョブを並列に実行することにより実現される。

10

#### 【0005】

##### 【発明が解決しようとする課題】

ところで、各磁気ディスク装置に格納されているデータの量は、一般に、均一ではない。このため、上述の並列処理により実行されるジョブの組合せが適切でないと、バックアップ時間が必要以上に長くなってしまう。例えば、第1～第3の磁気ディスク装置にそれぞれ格納されているデータを第1の磁気テープ記録媒体に書き込む処理と、第4～第6の磁気ディスク装置にそれぞれ格納されているデータを第2の磁気テープ記録媒体に書き込む処理とを並列して実行する場合を考える。この場合、もし第1～第3の磁気ディスク装置に格納されているデータ量が第4～第6の磁気ディスク装置に格納されているデータの量よりも大きかったとすると、第2の磁気テープ記録媒体にデータを書き込む処理が終了した後、しばらくの間、第1の磁気テープ記録媒体にデータを書き込む処理が続くことになる。これにより、全体のバックアップ時間が必要以上に長くなってしまう。

20

#### 【0006】

全体のバックアップ時間を短縮するためには、上述の並列処理により実行されるジョブの組合せが適切でなければならない。しかし、従来は、上述のジョブを自動的に適切に組み合わせる手法は考えられておらず、ユーザがマニュアルでジョブの組合せを設定（チューニング処理）していた。したがって、ジョブ数が多くなると、それらのジョブを適切に組み合わせることが困難であった。また、任意の磁気ディスク装置にデータが追加されたり、あるいは任意の磁気ディスク装置からデータが削除された場合には、その都度、ユーザがマニュアルでジョブの組合せを変更する必要があった。

30

#### 【0007】

なお、この問題は、磁気ディスク装置に格納されているデータを磁気テープ記録媒体を用いてバックアップする場合に限らず、任意の記録媒体に格納されているデータを他の記録媒体を利用してバックアップする際に発生し得る。

#### 【0008】

本発明の課題は、複数の記録媒体を利用してデータをバックアップする際の処理時間を自動的に短くする装置および方法を提供することである。

#### 【0009】

##### 【課題を解決するための手段】

本発明のバックアップ装置は、第1の記録媒体に格納されている複数のデータセットを複数の第2の記録媒体を利用してバックアップする構成を前提とし、第1の記録媒体から上記複数のデータセットを読み出す読み出し手段と、その読み出し手段により読み出された複数のデータセットを複数の第2の記録媒体に書き込むための複数の書き込み処理を並列的に実行する書き込み手段と、複数のデータセットを複数の第2の記録媒体を利用してバックアップするために要した時間を各データセットについてそれぞれ検出する検出手段と、過去に検出手段により検出されたデータセット毎のバックアップ時間に基づいて、それらのデータセットを上記並列的に実行される複数の書き込み処理に割り当てる割当て手段とを有する。

40

#### 【0010】

上記構成において、各書き込み処理の処理時間は、基本的に、それぞれの書き込み処理に割

50

り当てられているデータセットのバックアップ時間の合計値として表される。ここで、各データセットをバックアップするための時間は、検出手段により過去に検出された時間から推測することができる。したがって、過去に検出手段により検出されたデータセット毎のバックアップ時間に基づいてそれらのデータセットを複数の書き込み処理に割り当てるようすれば、各書き込み処理の処理時間を調整することができる。本発明のバックアップ装置は、これを利用し、全体のバックアップ時間が短くなるように、複数のデータセットを上記複数の書き込み処理に割り当てる。

#### 【0011】

割当て手段は、例えば、上記複数の書き込み処理の処理時間が均一または略均一になるように、上記複数のデータセットをそれらの書き込み処理に割り当てるようにしてよい。これにより、処理時間が必要以上に長くなる書き込み処理がなくなるので、全体のバックアップ時間が短くなる。

10

#### 【0012】

##### 【発明の実施の形態】

以下、本発明の実施形態について図面を参照しながら説明する。

図1は、本発明の実施形態のシステム構成図である。以下の実施例では、DASD (Direct Access Storage Device) に格納されているデータを磁気テープ装置を用いてバックアップする場合を想定する。なお、DASDは、ここでは、磁気ディスク装置とする。

#### 【0013】

ホストコンピュータ10は、複数の磁気ディスク装置 (DASD) 20を1つの記憶装置として利用することができる。即ち、ホストコンピュータ10は、所望のデータを1または複数の磁気ディスク装置20から取り出すことができ、また、データを所望の1または複数の磁気ディスク装置20に格納することができる。なお、磁気ディスク装置20からデータを読み出す処理および磁気ディスク装置20へデータを書き込む処理は、ホストコンピュータ10からの指示に従って、コントローラ30により実行される。

20

#### 【0014】

また、ホストコンピュータ10は、複数の磁気テープ装置 (CMT) 40を1つの記憶装置として利用することができる。すなわち、ホストコンピュータ10は、データを所望の1または複数の磁気テープ装置40に格納できる。この実施例のバックアップ処理では、磁気ディスク装置20から読み出した1または複数のデータセットが1または複数の磁気テープ装置40に格納される。そして、当然のことではあるが、ホストコンピュータ10は、磁気テープ装置40に格納されているデータを読み出すことができる。なお、磁気テープ装置40へデータを書き込む処理および磁気テープ装置40からデータを読み出す処理は、ホストコンピュータ10からの指示に従って、コントローラ50により実行される。

30

#### 【0015】

バックアップ処理においては、基本的に、すべての磁気ディスク装置20にそれぞれ格納されているデータが読み出され、それらのデータが複数の磁気テープ装置40に書き込まれる。ただし、一部の磁気ディスク装置20に格納されているデータについてバックアップ処理を行うこともできる。

40

#### 【0016】

バックアップ処理は、例えば、ボリューム単位で実行される。なお「ボリューム」は、一般には、磁気ディスクや磁気テープ等の1つの着脱可能な記録装置を指すことが多いが、ここでは、仮想的なボリュームを想定する。すなわち、1台の磁気ディスク装置を1つのボリュームを考えてもよいし、或いは、1台の磁気ディスク装置の中に複数のボリュームが存在するものとしてバックアップ処理を実行してもよい。

#### 【0017】

本実施形態のバックアップ処理においては、各ボリュームに格納されているデータを読み出してそれを所定の磁気テープ装置40に書き込む処理を「ジョブ」と呼ぶことにする。また、各ジョブを実行することによりバックアップされる1かたまりのデータを「データ

50

セット」と呼ぶことがある。

【0018】

ホストコンピュータ10は、複数のジョブを並列的に実行する。この場合、ホストコンピュータ10は、バックアップ処理に係わるジョブを複数のグループに振り分け、それら複数のグループに属するジョブを並列的に実行していく。ここで、ホストコンピュータ10は、バックアップ処理に要する総時間（すなわち、すべてのジョブが実行されるまでの時間）が出来るだけ短くなるようにジョブの組合せを最適化する。

【0019】

図2は、時間管理テーブルの例である。この時間管理テーブルは、図1に示すホストコンピュータ10が備えるメモリ上に設けられ、バックアップ処理に要する総時間が短くなるようにジョブの組合せを決定するために利用される。なお、ここでは、9つのボリューム（A01～A03、B01～B03、C01～C03）にそれぞれ格納されているデータのバックアップを実行するものとする。このとき、各ボリューム毎に対応してジョブが作成されるものとする。

10

【0020】

時間管理テーブルにおいて、「データ量」は、各ボリュームに格納されているデータの量を表す。図2に示す例では、例えば、ボリュームA01に1.5Gバイトのデータが格納されており、ボリュームA02に1.1Gバイトのデータが格納されている。この値は、各ボリュームのデータが更新されたとき（データの追加、削除、変更を含む）に更新される。なお、各ボリュームに格納されているデータの量を検出する方法は、既知の技術なので、具体的な方法についての説明は省略する。

20

【0021】

「バックアップ回数」は、各ボリュームに格納されているデータについてバックアップを実行した回数を表す。なお、この値は、対応するボリュームのデータが更新されたときに「0」に戻る。

【0022】

「当日の実績時間」は、バックアップ処理が実行されたときに、そのバックアップ処理に要した時間を表す。この値は、例えば、該当するボリュームからデータを読み出す処理が開始されたときから、そのデータが所定の磁気テープ装置に書き込まれるまでの時間を計測することにより得られる。

30

【0023】

「前日までの平均時間」は、過去に実行したバックアップ処理に要した時間の平均値である。図2に示す例では、たとえば、ボリュームA01について「15分」が得られており、ボリュームA02について「12分」が得られている。なお、この値を算出する方法は、後述する。

【0024】

「当日の予測時間」は、各ボリュームに格納されているデータをバックアップするためには、その値がそのまま使用される。一方、「前日までの平均時間」が算出されていない場合、即ち、あるボリュームのデータが更新された後の第1回目のバックアップ処理においては、そのボリュームについては、「データ量」に基づいて算出される。なお、新たに設けられたボリュームについても、「当日の予測時間」は「データ量」に基づいて算出される。図2に示す例では、ボリュームA01～A03、B01～B03、C01～C02については、「当日の予測時間」として「前日までの平均時間」がそのまま設定されており、ボリュームC03については、「当日の予測時間」として「データ量」に基づいて算出された値が設定されている。

40

【0025】

図3は、新規ボリュームが追加された際、あるいは任意のボリュームのデータが更新された際の動作を説明するフローチャートである。このフローチャートの処理は、新たなボリュームが追加されたとき、あるいは既存のボリュームの中の任意のボリュームのデータが

50

更新されたときに実行される。ここで、データ更新は、データの追加、データの削除、データを変更を含む。

【0026】

ステップS1では、新たに追加されたボリューム、或いはデータが更新されたボリュームに格納されているデータの量を検出する。ステップS2では、ステップS1で検出したデータ量を、時間管理テーブル内の対応するフィールドの「データ量」に設定する。なお、新たにボリュームが追加された場合には、時間管理テーブル内にそのボリュームに対応するフィールドを生成し、そのフィールドの「データ量」に検出した値が設定される。この後、ステップS3において、対応するフィールドの「バックアップ回数」に「0」を設定する。上記処理により、時間管理テーブルにおいて、対応するボリュームについての「データ量」が更新されると共に、「バックアップ回数」がリセットされる。

10

【0027】

図4は、バックアップ処理のフローチャートである。このフローチャートの処理は、バックアップ処理が起動されたときに、ホストコンピュータ10により実行される。

【0028】

ステップS11では、時間管理テーブルから任意のボリュームに対応するフィールドのデータを抽出する。ステップS12では、ステップS11で抽出したデータの中の「バックアップ回数」が「0」であるか否かを調べる。「バックアップ回数」が「0」であった場合は、ステップS13へ進み、そうでない場合にはステップS14へ進む。

20

【0029】

ステップS13では、ステップS11で抽出したデータの中の「データ量」に基づいて「当日の予測時間」を算出する。この算出においては、「バックアップ能力」も参照される。ここで、「バックアップ能力」は、ホストコンピュータ10、磁気ディスク装置20、磁気テープ装置40の性能などにより決まる。このため、もし、磁気ディスク装置20の性能が互いに同じでなかったとすると、あるボリュームについての「バックアップ能力」と他のボリュームについての「バックアップ能力」とが異なることになる。したがって、磁気ディスク装置20の性能が均一でない場合には、時間管理テーブルに「バックアップ能力」を設定するためのレコードを設け、各ボリューム毎にその値を予め設定しておく必要がある。

【0030】

30

ステップS13において、「当日の予測時間」は、例えば下記の式により求められる。ここで、「バックアップ能力」の単位は、例えば、「メガバイト/秒」である。

$$\text{「当日の予測時間」} = \text{「データ量」} / \text{「バックアップ能力」}$$

ステップS14では、ステップS11で抽出したデータの中の「前日までの平均時間」をそのまま「当日の予測時間」に設定する。

【0031】

ステップS15は、時間管理テーブルに登録されているすべてのボリュームについて上記ステップS12～S14の処理を実行するために設けられている。すなわち、未処理のボリュームが残っている場合には、ステップS11に戻って次のボリュームに対応するフィールドのデータを抽出する。

40

【0032】

ステップS16では、各ボリュームに対応するジョブを並列的に処理する際のジョブの組合せを決定する。図2に示す例では、時間管理テーブルに9つのボリュームが登録されており、9つのジョブが作成される。ここで、図1に示すシステムにおいて、コントローラ50a～50cが互いに並列的にデータ書き込み処理を実行できるものとする。この場合、各ジョブは、コントローラ50a～50cの中のいずれか1つに割り当てられることになる。なお、ジョブの組合せを決定する方法については、後述詳しく説明する。

【0033】

ステップS17では、ステップS16において決定された組合せに従って、並列的にバックアップ処理を実行する。ステップS18では、各ジョブの処理時間を検出することによ

50

り、各ボリュームごとにバックアップ時間を検出する。このステップS18の処理は、ステップS17の処理と並列に実行される。

【0034】

ステップS19では、バックアップ処理の結果に従って、時間管理テーブルを更新する。具体的には、まず、ステップS18において検出した各ボリューム毎のバックアップ時間を、それぞれ対応する「当日の実績時間」に設定する。続いて、各ボリューム毎に、過去の実績を参照してバックアップ処理に要した時間の平均値を算出する。この平均時間は、例えば、下記の式により求められる。

$$\text{「平均時間」} = (\text{「前日までの平均時間」} \times \text{「バックアップ回数」} + \text{「当日の実績時間」}) / (\text{「バックアップ回数」} + 1)$$

10

ここで、「前日までの平均時間」「バックアップ回数」「当日の実績時間」および「バックアップ回数」は、それぞれ時間管理テーブルに設定されている値を用いる。例えば、図2に示す例において、ボリュームA03のバックアップ処理のために21分を要したとすると、平均時間は、下記の値となる。

$$\text{「平均時間」} = (20 \times 10 + 21) / (10 + 1)$$

20.1

上述のようにしてボリューム毎に算出された「平均時間」は、時間管理テーブル内の対応する「前日までの平均時間」にそれぞれ設定される。更に、この後、時間管理テーブルにおいて各ボリュームの「バックアップ回数」に「1」が加算される。

20

【0035】

このように、本実施形態においては、バックアップ処理を開始する前に、各ジョブの処理時間の予測値を予め求めておき、これらの予測値に従ってジョブが適切に並べ替えられて並列的に実行される。このとき、これらのジョブは、全体のバックアップ時間が短くなるように、或いは並列に実行される複数の処理シーケンスの実行時間が均一または略均一になるように、並べ替えられる。以下、複数のジョブを複数の処理シーケンスに振り分ける方法の実施例を説明する。

【0036】

図5は、複数のジョブを複数の処理シーケンスに振り分ける方法の実施例であり、図4のステップS16に対応する。ここでは、図2に示した時間管理テーブルに基づいてバックアップ処理が実行されるものとする。また、複数の磁気テープ装置40へデータを書き込むための3つの処理シーケンス（図5に示す処理1～処理3）が並列的に実行されるものとする。

30

【0037】

この実施例の方法では、まず、各ジョブは、対応するボリュームの「当日の予測時間」が長い順番に並べられる。ここでは、図2に示す時間管理テーブルに従って、ジョブ（A03）、ジョブ（B01）、ジョブ（A01）、ジョブ（C01）、ジョブ（B02）、ジョブ（A02）、ジョブ（C03）、ジョブ（B03）、ジョブ（C02）の順番に並べられる。

【0038】

続いて、上述のようにして並べられたジョブは、図5に示すように、1つずつ順番に実行順序管理テーブルに設定されていく。このとき、これらのジョブは、まず昇り順（処理1、処理2、処理3の順番）に1つずつ順番に割り当てられ、次に、下り順（処理3、処理2、処理1の順番）に1つずつ順番に割り当てられる。以降、昇り順／下り順を交互に繰り返しながら、ジョブが1つずつ対応する処理シーケンスに割り当てられていく。これにより、図5に示す実行順序管理テーブルが得られる。

40

【0039】

この後、処理番号が偶数の処理シーケンスにおいて、ジョブの実行順序を反転させる。例えば、図5に示す例では、この処理により、「処理2」において、第1番目にジョブ（B03）、2番目にジョブ（B02）、3番目にジョブ（B01）が設定されることになる

50

。

## 【0040】

この処理は、本実施形態の方法において必須ではないが、並列処理の弊害を小さくする上で有効である。たとえば、大規模な磁気テープ装置の中には、多数の磁気テープを収容する収容棚および複数の書き込み／読み出し装置を備え、ロボットが、指定された磁気テープをその収容棚から取り出して指定された書き込み／読み出し装置定にマウントするものが知られている。このような構成の磁気テープ装置においては、あるジョブが終了して次のジョブが開始される際に、書き込み／読み出し装置から磁気テープを取り出して他の磁気テープをその装置にマウントする処理が発生する。このとき、複数のジョブが並列的に実行される環境においては、上述のマウント動作が競合する場合がある。ここで、マウントの競合が発生すると、各処理シーケンスのマウント動作は、他の処理シーケンスのマウント動作が終了するまで待たされることになる。よって、上記並列処理環境においては、あるジョブが終了して次のジョブが開始されるタイミングが互いに重なり合わないことが望ましい。

## 【0041】

ところが、本実施形態の方法で複数のジョブを複数の処理シーケンスに振り分けると、バックアップ処理の終了間際に実行時間の短いジョブが集中し、あるジョブが終了して次のジョブが開始されるタイミングが互いに重なり合う可能性が高くなってしまう。このため、処理番号が偶数の処理シーケンスにおいてジョブの実行順序を反転させることにより、上述の問題による弊害を軽減している。

## 【0042】

この後、ホストコンピュータ10は、上述のようにして作成した実行順序管理テーブルに従ってバックアップ処理を実行する。すなわち、ジョブ(A03)、ジョブ(A02)、ジョブ(C03)を順番に実行する処理シーケンス1、ジョブ(B03)、ジョブ(B02)、ジョブ(B01)を順番に実行する処理シーケンス2、およびジョブ(A01)、ジョブ(C01)、ジョブ(C02)を順番に実行する処理シーケンス3を並列的に実行する。

## 【0043】

図6は、複数のジョブを複数の処理シーケンスに振り分ける処理のフローチャートであり、図4のステップS16に対応する。ここでは、複数の磁気テープ装置40へデータを書き込むためのn個の処理シーケンスが並列的に実行されるものとする。

## 【0044】

ステップS21では、時間管理テーブルに設定されている「当日の予測時間」の長い順番にジョブをソートし、ジョブ列を作成する。ステップS22では、ステップS21で作成したジョブ列の先頭からn個のジョブを抽出する。ステップS23では、ステップS22で抽出したn個のジョブを、n個の処理シーケンスに対して昇り順に割り当てる。ステップS24では、ステップS21で作成したジョブ列に未処理のジョブが残っているか否かを調べる。そして、未処理のジョブが残っている場合には、ステップS25へ進み、そうでない場合にはステップS28へ進む。

## 【0045】

ステップS25では、上記ジョブ列の先頭から未だ抽出されていないn個のジョブを抽出する。ステップS26では、ステップS25で抽出したn個のジョブを、n個の処理シーケンスに対して下り順に割り当てる。ステップS27では、ステップS21で作成したジョブ列に未処理のジョブが残っているか否かを調べる。そして、未処理のジョブが残っている場合には、ステップS22に戻り、そうでない場合にはステップS28へ進む。なお、ステップS22またはS25において、上記ジョブ列に残っているジョブの数がn個未満のときは、それらのジョブをすべて抽出する。この後、ステップS28において、処理シーケンス番号が偶数の処理シーケンスにおいて、ジョブの実行順序を反転させる。

## 【0046】

なお、上記実施例では、「当日の予測時間」の長い順番にジョブを各処理シーケンスに振り分けているが、「当日の予測時間」の短い順番にジョブを各処理シーケンスに振り分け

10

20

30

40

50

るようにしてよい。

【0047】

図7は、上記実施例による効果を説明する図である。ここでは、図2に示す時間管理テーブルに登録されている9つのボリュームについてバックアップ処理を行う場合を想定し、並列的に実行される3つの処理シーケンスについて、それぞれ「当日の予測時間」の合計値を示す。

【0048】

図7(a)は、本実施形態の方法を使用せずに、ボリュームの識別番号の並び順に従って9つのジョブを単純に3つの処理シーケンスに割り当てた場合を示している。この場合、処理シーケンス2、3については、実行時間が比較的短くなることが期待されるが、処理シーケンス1の実行時間はかなり長くなるものと予想される。このように、幾つかの処理シーケンスが比較的早く終了したとしても、実行時間の長い処理シーケンスが1つでも含まれていると、すべてのバックアップ処理が終了するまでの時間は結果として長くなってしまう。

【0049】

従来は、この問題を解決しようとした場合、ユーザが手作業で、ジョブの組合せを変更していた。たとえば、実行時間が最も長かった処理シーケンスの所定のジョブと実行時間が最も短かった処理シーケンスの所定のジョブとを入れ替えてバックアップ処理を実行し、その結果に基づいてさらにジョブの入れ替えが必要か否かを判断しながらジョブの入れ替え作業を繰り返していた。このため、ジョブ数が多くなった場合、あるいは並列に実行される処理シーケンスの数が多くなった場合には、そのような入れ替え作業は困難であった。また、各ボリュームのデータ量が変わった場合、あるいは新たなボリュームが追加された場合には、ジョブを組み合わせる作業をやり直す必要があり、ユーザの負担が重かった。

【0050】

図7(b)は、本実施形態の方法を使用した場合のバックアップ処理の予測時間を示す。このグラフは、図5に示した実行順序管理テーブルに従っている。

このように、本実施形態の方法を使用すると、本実施形態の方法を使用しない場合と比べて、処理シーケンス間のばらつきが小さくなると共に、実行時間の最大値が小さくなっている。これにより、すべてのバックアップ処理が終了するまでの時間が短くなる。

【0051】

図8は、複数のジョブを複数の処理シーケンスに振り分ける方法の他の実施例であり、図4のステップS16に対応する。ここでも、図2に示した時間管理テーブルに基づいてバックアップ処理が実行されるものとする。また、複数の磁気テープ装置40へデータを書き込むための3つの処理シーケンス(図5に示す処理1～処理3)が並列的に実行されるものとする。

【0052】

この実施例の方法においても、まず、各ジョブは、図5～図6に示した実施例の方法と同様に、対応するボリュームの「当日の予測時間」が長い順番に並べられる。

【0053】

続いて、各処理シーケンスの「目標バックアップ時間」を算出する。この「目標バックアップ時間」は、下記の式により算出される。

「目標バックアップ時間」 = 「当日の予測時間」 / 「並列に実行される処理シーケンスの数」

例えば、図2に示した例では、下記のようになる。

$$\begin{aligned}
 \text{「目標バックアップ時間」} &= (15+12+20+18+14+8+15+6+12) / 3 \\
 &= 40 \text{ (分)}
 \end{aligned}$$

そして、上述のようにして並べられたジョブは、図8に示すように、1つずつ順番に実行順序管理テーブルに設定されていく。このとき、これらのジョブは、各処理シーケンスの

10

20

30

40

50

実行時間が「目標バックアップ時間」に対して一定の範囲内に収まるように適切に選択されて対応する処理シーケンスに振り分けられる。図8に示す例では、一定の範囲として「± 20 %」が設定されている。

#### 【0054】

具体的に説明する。まず、始めに、処理シーケンス1に割り当てるべきジョブを決定する。この場合、まず、時間管理テーブルに登録されているボリュームの中で「当日の予測時間」が最も長いボリュームに対応するジョブを処理シーケンス1に割り当てる。ここでは、処理シーケンス1にジョブ(A03)が割り当たられる。そして、このジョブの「当日の予測時間」が「目標範囲」に入っているか否を調べる。なお、「目標範囲」は、「目標バックアップ時間±20 %」、すなわち32分~48分であるものとする。ここでは、ジョブ(A03)の「当日の予測時間」は「20分」であり、上記目標範囲には入っていない。

10

#### 【0055】

この場合、処理シーケンス1に、「当日の予測時間」が2番目に長いボリュームに対応するジョブを割り当てる。ここでは、処理シーケンス1にジョブ(B01)が割り当たられる。そして、このジョブ(B01)及び先に処理シーケンス1に割り当たっているジョブ(A03)の「当日の予測時間」の和が、目標範囲に入っているか否かを調べる。ここでは、ジョブ(B01)の「当日の予測時間」は「18分」なので、上記2つのジョブの「当日の予測時間」の和は「38分」である。すなわち、これら2つのジョブの「当日の予測時間」の和は、目標範囲に入っている。したがって、処理シーケンス1にジョブを割り当てる作業を終了する。

20

#### 【0056】

以下、処理シーケンス2、3についても、同様の手順でジョブが割り当たられる。これにより、図8に示す実行順序管理テーブルが得られる。

なお、目標範囲を変更すると、ジョブの割当てが変わることがある。例えば、目標範囲として、「目標バックアップ時間±5 %」、すなわち、38分~42分とした場合の例を図9に示す。このとき、もし、処理シーケンス2に対して、ジョブ(A01)およびジョブ(C01)を割り当てる後にジョブ(B02)を割り当てるとき、それらのジョブの「当日の予測時間」の合計値は、「44分」となり、目標範囲を越えてしまう。したがって、この場合は、処理シーケンス2に対して、ジョブ(A01)およびジョブ(C01)を割り当てる後に、「当日の予測時間」がジョブ(B02)よりも短いジョブを選択して、それを処理シーケンス2に割り当てる。ここでは、処理シーケンス2にジョブ(A02)が割り当たっている。これにより、処理シーケンス2の予測時間は、「42分」になり、目標範囲に収まることになる。

30

#### 【0057】

図10は、図8または図9に示す実施例の動作を説明するフローチャートであり、図4のステップS16に対応する。

ステップS31では、図6に示した図21と同様に、時間管理テーブルに設定されている「当日の予測時間」の長い順番にジョブをソートし、ジョブ列を作成する。ステップS32では、時間管理テーブルを参照し、並列に実行すべき処理シーケンス処理についての「目標バックアップ時間」を算出する。そして、ステップS33において、ステップS32で算出した「目標バックアップ時間」に基づいて目標範囲を設定する。

40

#### 【0058】

ステップS34では、1以上のジョブが割り当たるべき処理シーケンスを1つ抽出する。ステップS35では、ステップS34で抽出した処理シーケンスに割り当てるべきジョブを1つ抽出する。ステップS36では、ステップS34で抽出したジョブの「当日の予測時間」の合計値を算出する。すなわち、ステップS35~S37、S41~S43の処理が繰り返し実行されると、複数のジョブが抽出されることになるので、その場合に、それら複数のジョブの「当日の予測時間」の合計値を算出する。もっとも、第1番目のジョブが抽出された時点では、その抽出されたジョブの「当日の予測時間」がステップS3

50

6により算出された合計値として扱われる。

【0059】

ステップS37では、ステップS36で算出した合計値がステップS33で設定した目標範囲内であるか否かを調べる。そして、その合計値が目標範囲内であれば、ステップS38において、抽出された1以上のジョブを、ステップS34で抽出した処理シーケンスに割り当てる。

【0060】

ステップS39は、並列に実行されるすべての処理シーケンスについて上記ステップS35～S38の処理を実行するために設けられている。すなわち、未処理の処理シーケンスが残っていれば、ステップS34に戻って次の処理シーケンスを選択する。

10

【0061】

すべての処理シーケンスについて上記ステップS35～S38を実行すると、ステップS40において、いずれの処理シーケンスにも割り当てられていないジョブが残っていないかを調べる。このとき、もし、ジョブが残っていれば、ステップS51においてエラー処理を実行する。

【0062】

ステップS36で算出した合計値がステップS33で設定した目標範囲外であった場合(ステップS37:No)には、ステップS41において、いずれの処理シーケンスにも割り当てられていないジョブ或いは未だ抽出されていないジョブが残っているか否かを調べる。そして、もし、ジョブが残っていなければ、当該処理シーケンスの処理時間を目標範囲内にすることのできないものとみなし、ステップS51のエラー処理を実行する。

20

【0063】

一方、ジョブが残っている場合(ステップS41:Yes)には、ステップS36で算出した合計値が上記目標範囲に達していないのか、或いはそれを越えてしまっているのかを調べる。そして、その合計値が目標範囲に達していないのであれば、ステップS35に戻って次のジョブを抽出する。また、その合計値が目標範囲を越えてしまっていた場合には、ステップS43において直前に抽出したジョブをジョブ列に戻し、その後にステップS35において次のジョブを抽出する。

【0064】

ステップS51のエラー処理は、例えば、ステップS33に戻って目標範囲を変更する処理である。このようなエラー処理を導入する場合には、初期段階において、目標範囲の幅を狭く設定する。このとき、もし、各処理シーケンスの実行時間がそのような目標範囲に収まれば、それらの処理シーケンスの実行時間のばらつきは小さくなり、全体のバックアップ時間は論理的に可能な最短時間に近くなる。一方、各処理シーケンスの実行時間がそのような目標範囲に収まらなかった場合には、目標範囲を少しづつ広げていく。このような構成とすれば、処理シーケンス間のばらつきが必要以上に大きくなることはなく、また、全体のバックアップ時間が必要以上に長くなることもない。

30

【0065】

なお、上記実施例では、ステップS31において「当日の予測時間」が長い順番にジョブをソートし、その順番に従ってジョブを対応する処理シーケンスに割り当てるが、本発明はこれに限定されるものではない。すなわち、例えば、「当日の予測時間」が短い順番にジョブを対応する処理シーケンスに割り当てる手順でもよい。また、図10のフローチャートにおいて、ステップS31の処理を省略し、任意の順番でジョブが抽出されてステップS34以降の処理が実行されるような手順であってもよい。

40

【0066】

図11は、図8～図10を参照しながら説明した実施例による効果を説明する図である。図11(a)は、目標範囲として「目標バックアップ時間±20%」が設定された場合を示し、図11(b)は、目標範囲として「目標バックアップ時間±5%」が設定された場合を示している。このように、目標範囲の幅を狭くすると、各処理シーケンスの処理時間が均一化され、必要以上に処理時間が長くなる処理シーケンスがなくなるので、全体のバック

50

アップ時間を短くすることができる。

【0067】

なお、図2～図11に示した実施例では、バックアップ処理が実行される際、その都度、各ジョブ毎にバックアップ時間が計測される。そして、並列的に実行される複数の処理シーケンスにジョブを振り分ける際、それらのジョブの過去のバックアップ時間に基づいて次回のバックアップ処理に要するであろう時間を予測し、それらの予測時間を参照していた。しかし、本発明はこの方法に限定されるものではない。すなわち、例えば、バックアップ処理の実行時に各ジョブのバックアップ時間を計測することなく、各ボリュームに蓄積されているデータの量およびバックアップ能力のみに基づいて、並列的に実行される複数の処理シーケンスに振り分けるべきジョブを決定してもよい。

10

【0068】

また、上述の実施例では、磁気ディスク装置などのDASDに格納されているデータを磁気テープ装置を用いてバックアップするケースを示したが、本発明はこの構成に限定されるものではなく、任意の記録媒体に格納されているデータを任意の記録媒体を用いてバックアップする場合に適用可能である。

【0069】

上述したバックアップ処理を実行する機能は、コンピュータを用いて上述のフローチャートに示した処理を記述したプログラムを実行することにより実現される。上述の実施例では、このプログラムは、ホストコンピュータ10により実行される。

【0070】

図12は、ホストコンピュータ10のブロック図である。CPU101は、上述のフローチャートに示した処理を記述したプログラムを記憶装置102からメモリ103にロードして実行する。記憶装置102は、たとえばハードディスクであり、上記プログラムを格納する。一方、メモリ103は、例えば半導体メモリであり、CPU101の作業領域として使用される。なお、時間管理テーブルおよび実行順序管理テーブルは、このメモリ103に作成される。

20

【0071】

記録媒体ドライバ104は、CPU101の指示に従って可搬性記録媒体105にアクセスする。可搬性記録媒体105は、例えば、半導体デバイス(PCカード等)、磁気的作用により情報が入出力される媒体(フロッピーディスク、磁気テープなど)、光学的作用により情報が入出力される媒体(光ディスクなど)を含む。通信制御装置106は、CPU101の指示に従って網との間でデータを送受信する。

30

【0072】

図13は、本発明に係わるソフトウェアプログラムなどの提供方法を説明する図である。本発明に係わるプログラムは、例えば、以下の3つの方法の中の任意の方法により提供される。

【0073】

(a) ホストコンピュータ10にインストールされて提供される。この場合、プログラム等は、たとえば、出荷前にブレインストールされる。

(b) 可搬性記録媒体に格納されて提供される。この場合、可搬性記録媒体105に格納されているプログラム等は、基本的に、記録媒体ドライバ104を介して記憶装置102にインストールされる。

40

【0074】

(c) 網上のサーバから提供される。この場合、基本的には、ホストコンピュータ10がサーバに格納されているプログラム等をダウンロードすることによってそのプログラム等を取得する。

【0075】

【発明の効果】

本発明によれば、バックアップ手順が最適化されるので、バックアップ時間が短くなる。また、このバックアップ手順の最適化処理は、コンピュータにより実行されるので、ユー

50

ザに負担がかかることはない。

### 【図面の簡単な説明】

【図1】本発明の実施形態のシステム構成図である。

【図2】時間管理テーブルの例である。

【図3】新規ボリュームが追加された際、あるいは任意のボリュームのデータが更新された際の動作を説明するフロー・チャートである。

【図4】バックアップ処理のフローチャートである。

【図5】複数のジョブを複数の処理シーケンスに振り分ける方法の実施例である。

【図6】図5に示す実施例の動作を説明するフローチャートである。

【図7】(a)は、本実施形態の方法を使用しない場合、(b)は、本実施形態を使用した場合のバックアップ処理の予測時間を示す図である。

【図8】複数のジョブを複数の処理シーケンスに振り分ける方法の他の実施例である。

【図9】目標範囲を狭くした場合の例である。

【図10】図8に示す実施例の動作を説明するフローチャートである。

【図11】(a)は、目標範囲の幅が広い場合、(b)は、目標範囲の幅が狭い場合の例である。

【図12】本発明の機能を記述したプログラムを実行するコンピュータのブロック図である。

【図13】本発明に係わるソフトウェアプログラムなどの提供方法を説明する図である。

## 【符号の説明】

## 1.0 来ストヨンピュータ

## 2.0 磁気ディスク装置 (PASP)

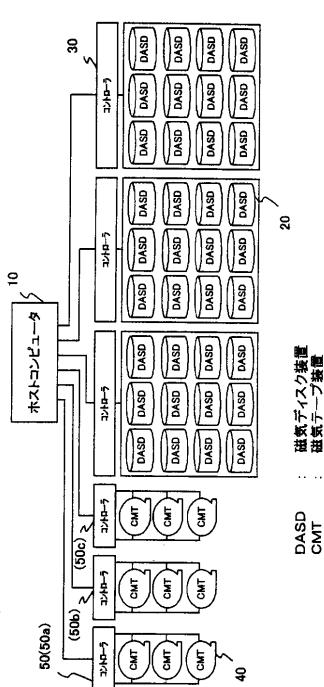
## 4.0 磁気テープ装置

【圖 1】

【圖2】

### 本発明の実施形態のシステム構成図

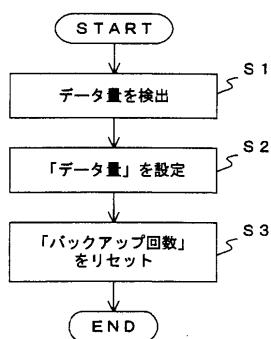
### 時間管理 テーブルの例



VOL=A01	VOL=A02	VOL=A03	VOL=B01	VOL=B02	VOL=B03	VOL=C01	VOL=C02	VOL=C03
日本式の平均時間								
午前の予習時間 (分)	15	12	20	8	14	8	15	6
日本の予習時間								
午前の予習時間 (分)	15	12	20	18	14	8	15	6
米国式の予習時間								
午前の予習時間 (分)	10	10	10	10	10	10	10	10
アーチャー回数								
(回)	1.5	1.8	1.6	1.4	0.8	1.5	0.6	1.2
アーチャー回数								
(回)	1.5	1.8	1.6	1.4	0.8	1.5	0.6	1.2

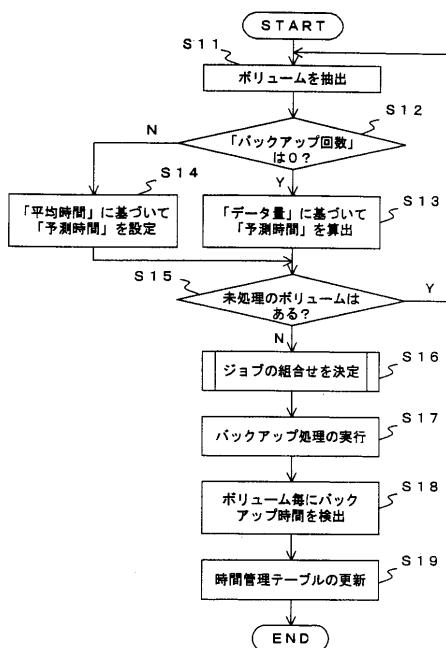
【図3】

新規ボリュームが追加された際、あるいは  
任意のボリュームのデータが更新された際の  
動作を説明するフローチャート



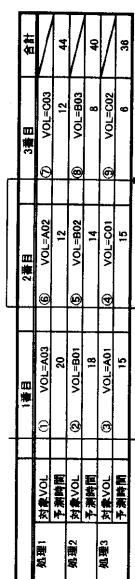
【図4】

バックアップ処理のフローチャート



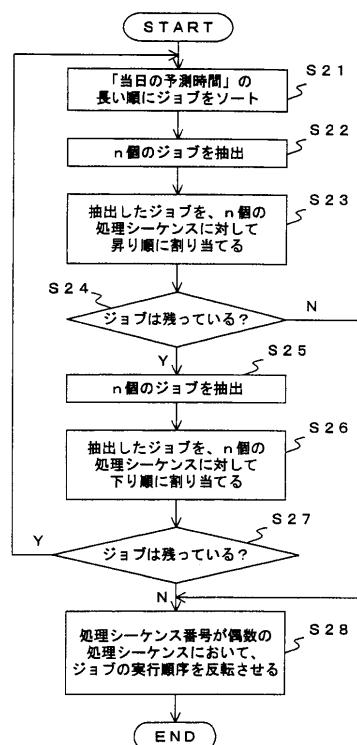
【図5】

複数のジョブを複数の処理シーケンスに  
振り分ける方法の実施例



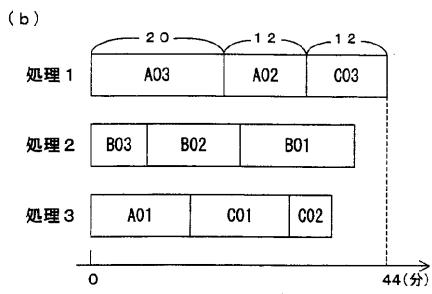
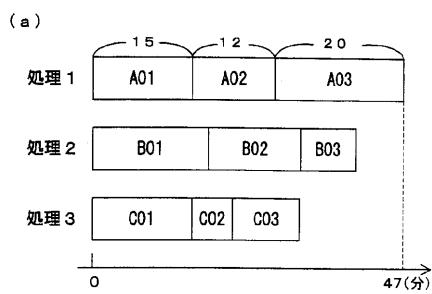
【図6】

図5に示す実施例の動作を説明する  
フローチャート



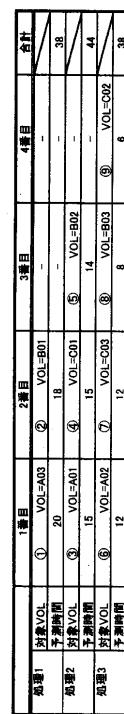
【図7】

(a)は、本実施形態の方法を使用しない場合、  
(b)は、本実施形態を使用した場合のバックアップ処理の予測時間を示す図



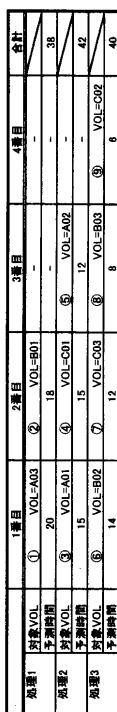
【図8】

複数のジョブを複数の処理シーケンスに  
振り分ける方法の他の実施例



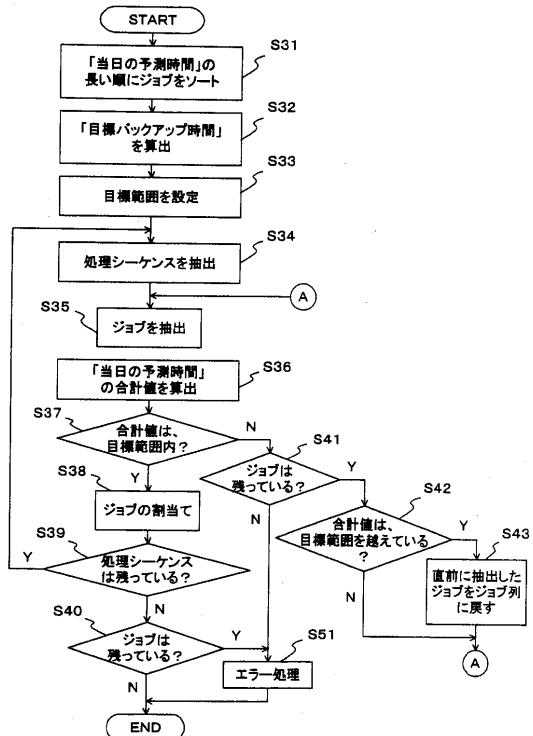
【図9】

目標範囲を狭くした場合の例



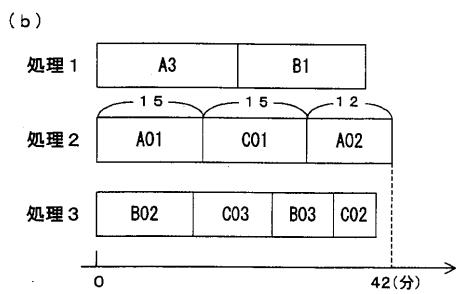
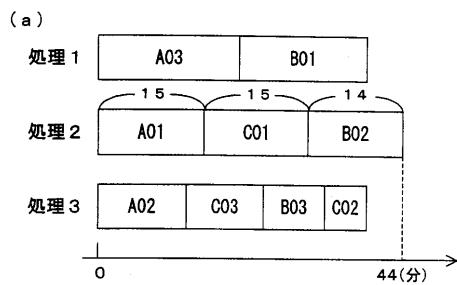
【図10】

図8に示す自効例の動作を説明するフローチャート



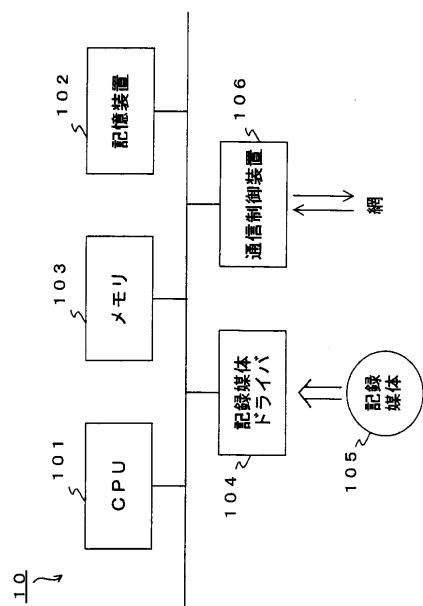
【図 1 1】

(a) は、目標範囲の幅が広い場合、  
(b) は目標範囲の幅が狭い場合の例



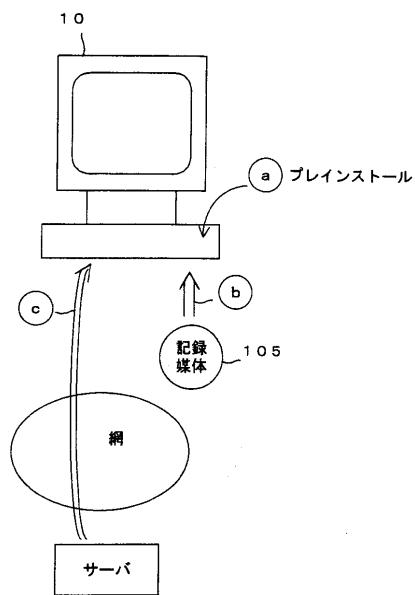
【図 1 2】

本発明の機能を記述したプログラムを  
実行するコンピュータのブロック図



【図 1 3】

本発明に係わるソフトウェアプログラム等の  
提供方法を説明する図



---

フロントページの続き

(56)参考文献 特開平10-027070(JP,A)  
特開平07-210430(JP,A)  
特開平09-212376(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 12/00  
G06F 3/06  
G06F 12/16  
JSTPplus(JDream2)