

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6090431号  
(P6090431)

(45) 発行日 平成29年3月8日 (2017.3.8)

(24) 登録日 平成29年2月17日 (2017.2.17)

(51) Int.Cl.

F I

G O 6 F 12/00 (2006.01)  
 G O 6 F 13/10 (2006.01)  
 G O 6 F 3/06 (2006.01)  
 G O 6 F 12/08 (2016.01)

G O 6 F 12/00 5 1 4 K  
 G O 6 F 12/00 5 4 5 A  
 G O 6 F 13/10 3 4 0 A  
 G O 6 F 3/06 3 0 2 A  
 G O 6 F 12/08 5 5 1 C

請求項の数 11 (全 25 頁)

(21) 出願番号 特願2015-507761 (P2015-507761)  
 (86) (22) 出願日 平成25年3月27日 (2013.3.27)  
 (86) 国際出願番号 PCT/JP2013/058930  
 (87) 国際公開番号 W02014/155553  
 (87) 国際公開日 平成26年10月2日 (2014.10.2)  
 審査請求日 平成27年9月30日 (2015.9.30)

(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 100103528  
 弁理士 原田 一男  
 (72) 発明者 小池 康夫  
 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  
 (72) 発明者 藤田 和久  
 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  
 (72) 発明者 前田 敏之  
 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

最終頁に続く

(54) 【発明の名称】 分散処理のための情報処理方法、情報処理装置及びプログラム、並びに分散処理システム

(57) 【特許請求の範囲】

【請求項 1】

データの処理を行う複数の情報処理装置と管理装置とを含む分散処理システムにおいて  
 実行される情報処理方法であって、

前記複数の情報処理装置のうち第1の情報処理装置が、

前記第1の情報処理装置が管理する第1のデータに対するアクセスがあった場合に、前  
 記管理装置から、前記第1のデータに対するアクセスの後にアクセスされる可能性を有す  
る第2のデータの識別情報及び当該第2のデータを管理する情報処理装置の識別情報を取得し、

前記第2のデータを管理する情報処理装置が前記第1の情報処理装置である場合に、前  
 記第2のデータをデータ格納部から読み出し、キャッシュに展開する

処理を実行する情報処理方法。

【請求項 2】

前記第1の情報処理装置が、

前記第2のデータを管理する情報処理装置が前記第1の情報処理装置とは異なる第2の  
 情報処理装置である場合に、前記第2の情報処理装置に対し、前記第2のデータをキャッ  
 シュに展開することを要求する第1の要求を送信する

処理をさらに実行する請求項1記載の情報処理方法。

【請求項 3】

前記第1の情報処理装置が、

10

20

前記第 1 のデータに対するアクセスが終了した場合に、前記キャッシュに展開されている前記第 1 のデータを削除する

処理をさらに実行する請求項 1 又は 2 記載の情報処理方法。

【請求項 4】

データの処理を行う複数の情報処理装置と管理装置とを含む分散処理システムにおいて 実行される情報処理方法であって、

前記管理装置が、

前記複数の情報処理装置のうち第 1 のデータを管理する第 1 の情報処理装置から、前記第 1 のデータの識別情報を含む要求を受信した場合に、データの識別情報と当該データに対するアクセスの後にアクセスされる可能性を有するデータの識別情報とを対応付けて格納する関連データ格納部から、前記第 1 のデータに対するアクセスの後にアクセスされる可能性を有する第 2 のデータの識別情報を抽出し、

抽出された前記第 2 のデータの識別情報と当該第 2 のデータを管理する情報処理装置の識別情報とを、前記第 1 の情報処理装置に送信する

処理を実行する情報処理方法。

【請求項 5】

前記管理装置が、

データの内容の関連性、データの格納場所及びデータのタイムスタンプの少なくともいずれかに基づき、前記関連データ格納部に格納されているデータを更新する

処理をさらに実行する請求項 4 記載の情報処理方法。

【請求項 6】

前記管理装置が、

前記分散処理システムにおけるデータへのアクセスの履歴に基づき、前記関連データ格納部に格納されているデータを更新する

処理をさらに実行する請求項 4 記載の情報処理方法。

【請求項 7】

データの処理を行う複数の情報処理装置と管理装置とを含む分散処理システムにおいて キャッシュを管理するためのプログラムであって、

前記複数の情報処理装置のうち第 1 の情報処理装置に、

前記第 1 の情報処理装置が管理する第 1 のデータに対するアクセスがあった場合に、前記管理装置から、前記第 1 のデータに対するアクセスの後にアクセスされる可能性を有する第 2 のデータの識別情報及び当該第 2 のデータを管理する情報処理装置の識別情報を取得し、

前記第 2 のデータを管理する情報処理装置が前記第 1 の情報処理装置である場合に、前記第 2 のデータをデータ格納部から読み出し、キャッシュに展開する

処理を実行させるためのプログラム。

【請求項 8】

データの処理を行う複数の情報処理装置と管理装置とを含む分散処理システムにおいて キャッシュを管理するためのプログラムであって、

前記管理装置に、

前記複数の情報処理装置のうち第 1 のデータを管理する第 1 の情報処理装置から、前記第 1 のデータの識別情報を含む要求を受信した場合に、データの識別情報と当該データに対するアクセスの後にアクセスされる可能性を有するデータの識別情報とを対応付けて格納する関連データ格納部から、前記第 1 のデータに対するアクセスの後にアクセスされる可能性を有する第 2 のデータの識別情報を抽出し、

抽出された前記第 2 のデータの識別情報と当該第 2 のデータを管理する情報処理装置の識別情報とを、前記第 1 の情報処理装置に送信する

処理を実行させるためのプログラム。

【請求項 9】

データの処理を行う複数の情報処理装置と管理装置とを含む分散処理システムにおける

10

20

30

40

50

前記複数の情報処理装置に含まれる第1の情報処理装置であって、

第1のデータに対するアクセスがあった場合に、前記管理装置から、前記第1のデータに対するアクセスの後にアクセスされる可能性を有する第2のデータの識別情報及び当該第2のデータを管理する情報処理装置の識別情報を取得する取得部と、

前記第2のデータを管理する情報処理装置が前記第1の情報処理装置である場合に、前記第2のデータをデータ格納部から読み出し、キャッシュに展開するキャッシュ処理部と

、

を有する第1の情報処理装置。

【請求項10】

データの処理を行う複数の情報処理装置と管理装置とを含む分散処理システムにおける前記管理装置であって、

データの識別情報と当該データに対するアクセスの後にアクセスされる可能性を有するデータの識別情報とを対応付けて格納する関連データ格納部と、

第1のデータを管理する第1の情報処理装置から、前記第1のデータの識別情報を含む要求を受信した場合に、前記関連データ格納部から、前記第1のデータに対するアクセスの後にアクセスされる可能性を有する第2のデータの識別情報を抽出する抽出部と、

抽出された前記第2のデータの識別情報と当該第2のデータを管理する情報処理装置の識別情報とを、前記第1の情報処理装置に送信する送信部と、

を有する管理装置。

【請求項11】

データの処理を行う複数の情報処理装置と管理装置とを含む分散処理システムであって

、

前記複数の情報処理装置のうち第1の情報処理装置が、

前記第1の情報処理装置が管理する第1のデータに対するアクセスがあった場合に、前記管理装置から、前記第1のデータに対するアクセスの後にアクセスされる可能性を有する第2のデータの識別情報及び当該第2のデータを管理する情報処理装置の識別情報を、前記第1のデータの識別情報を含む要求を前記管理装置に送信することにより取得する取得部と、

前記第2のデータを管理する情報処理装置が前記第1の情報処理装置である場合に、前記第2のデータをデータ格納部から読み出し、キャッシュに展開するキャッシュ処理部と

、

を有し、

前記管理装置が、

データの識別情報と当該データに対するアクセスの後にアクセスされる可能性を有するデータの識別情報とを対応付けて格納する関連データ格納部と、

前記第1の情報処理装置から前記要求を受信した場合に、前記関連データ格納部から、前記第1のデータの識別情報に対応付けられている前記第2のデータの識別情報を抽出する抽出部と、

抽出された前記第2のデータの識別情報と当該第2のデータを管理する情報処理装置の識別情報とを、前記第1の情報処理装置に送信する送信部と、

を有する分散処理システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、分散処理技術に関する。

【背景技術】

【0002】

近年、ウェブサイト、センサ或いは携帯端末等において発生する大量のデータ（以下、ビッグデータと呼ぶ）を短時間で処理する技術が求められている。

【0003】

10

20

30

40

50

ビッグデータを処理する技術として、ハドゥープ (Hadoop) という分散処理フレームワークが知られている。Hadoopにおいては、マプリデュース (MapReduce) という分散処理方式に基づき1つのファイルを複数のファイルに分割し、複数のノードがファイルを並列処理する。Hadoopは、ノード数の増加に応じて処理性能が線形に高くなることを特徴とする。そのため、Hadoopを利用すれば、例えば数ペタバイト以上のデータを数十から数千のノードで処理するような大規模なシステムを比較的容易に実現することができる。

#### 【0004】

Hadoopは、分散処理を効果的に行うため、HDFS (Hadoop Distributed File System) と呼ばれる分散ファイルシステムを採用している。HDFSは、ファイルを特定の  
10  
の大きさに分割し、分割されたファイルを複数のサーバに分散して格納すると共に、分割されたファイルの複製を生成し、複製元のファイルを保持するサーバとは別のサーバに保持させる。分散処理においては、複製元のファイル及び複製されたファイルのいずれかを処理すればよいため、HDFSが作業プロセスが空いているサーバに処理を振り分けることにより、サーバのリソース (CPU (Central Processing Unit) 資源等) を有効に利用できる。

#### 【0005】

しかし、HDFS等の分散ファイルシステムは、処理の振り分けの際にOS (Operating System) のキャッシュの状態を考慮しない。そのため、分散処理システムは、あるサーバのキャッシュ上にデータがあってもそれを考慮せず別のサーバに処理を振り分けてしま  
20  
うので、ディスクI/O (Input/Output) が発生する。ディスクI/Oは時間がかかる処理であり、例えば数ギガバイトのデータの処理に要する時間の約半分をディスクのシーク時間が占めることがある。

#### 【0006】

図1を用いて、分散ファイルシステムによるビッグデータの処理について説明する。サーバAのディスクにはファイルAが格納されており、サーバBのディスクにはファイルBが格納されており、サーバCのディスクにはファイルBの複製及びファイルCが格納されている。また、サーバCのキャッシュにはファイルBが展開されている。分散処理システムにおけるいずれかのサーバに配置されたアプリケーションプログラムは、ファイルA、ファイルB、ファイルCの順に処理を行う。アプリケーションプログラムは、サーバA  
30  
におけるファイルAに対する処理をした後、サーバB又はサーバCにおけるファイルBに対して処理をする。サーバCにおけるファイルBに対して処理をすれば、キャッシュを利用できるためディスクI/Oは発生しない。しかし、サーバB又はサーバCのいずれを選択するかは、サーバB又はサーバCにおける空きプロセスの状況等によるため、サーバBを選択する場合もある。その場合にはディスクI/Oが発生するため、ファイルに対する処理の完了までの時間が大幅に遅れる。

#### 【0007】

よって、ファイルに対する処理を短時間で完了させるという観点から、分散処理システムにおけるキャッシュヒット率を向上させることが好ましい。

#### 【0008】

分散処理及びキャッシュの管理については、例えば以下のような技術が知られている。具体的には、MapReduce処理システムが、処理対象のデータを、データの更新頻度に基づき複数のグループに分け、グループに含まれるデータの更新頻度に基づきグループ更新頻度を計算する。そして、MapReduce処理システムは、グループ更新頻度が閾値以下であるグループに対するMapReduce処理段階の部分的結果を生成し、生成された部分的結果をキャッシュする。これによりMapReduce処理においてキャッシュを有効に利用する。

#### 【0009】

また、キャッシュの有効利用に関して、以下のような技術も存在する。具体的には、ファイルのオープン時にファイルの全体又は部分領域のデータを多重化して複数のドライブ  
50

に分散配置すると共に、ファイルのクローズ時にファイルの多重化データを消去する。

【 0 0 1 0 】

また、複数のクライアントから特定のファイルサーバにアクセス要求が集中することによるスループットの低下を防ぐ技術が存在する。具体的には、マスタファイルサーバが、負荷が軽いファイルサーバを選定し、選定されたファイルサーバに、クライアントから伝送されたファイルアクセス要求を分配する。

【 0 0 1 1 】

また、広域ネットワークの状態に関わらず高速なファイルアクセスを提供するための技術が存在する。具体的には、キャッシュサーバは、ファイルアクセスを記録するアクセス履歴データベースを有しており、アクセス履歴データベースによってアクセス頻度が高いファイルの更新間隔を分析し、先読みすべきファイルを決定する。そして、キャッシュサーバは、広域ネットワークが空いている時間帯にアクセス頻度が高いファイルを先読みし、クライアントからの要求に対し先読みしたファイルを渡す。

【 0 0 1 2 】

また、クライアントにログインするユーザ毎に、使用頻度が高いファイルを優先的にキャッシングすることにより、キャッシュヒット率を向上させ、ファイルアクセスを高速化する技術が存在する。

【 0 0 1 3 】

しかしながら、これらの技術を利用したとしても、分散処理システムにおけるキャッシングヒット率を十分に向上させることができない場合がある。また、サーバにおけるキャッシングを有効に利用するという観点において、従来技術は十分ではない。

【 先行技術文献 】

【 非特許文献 】

【 0 0 1 4 】

【 非特許文献 1 】 Jeffrey Dean and Sanjay Ghemawat , "MapReduce: Simplified Data Processing on Large Clusters" , [online] , 2 0 0 4 年 1 2 月 , Symposium on Operating System Design and Implementation 2004 , [平成 2 5 年 1 月 1 0 日検索] , インターネット <URL:http://research.google.com/archive/mapreduce.html >

【 特許文献 】

【 0 0 1 5 】

【 特許文献 1 】 特開 2 0 1 0 - 9 2 2 2 2 号公報

【 特許文献 2 】 特開平 6 - 3 3 2 6 2 5 号公報

【 特許文献 3 】 特開平 6 - 3 3 2 7 8 2 号公報

【 特許文献 4 】 特開平 1 1 - 2 4 9 8 1 号公報

【 特許文献 5 】 特開平 7 - 9 3 2 0 5 号公報

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 1 6 】

従って、本発明の目的は、1つの側面では、分散処理システムにおけるキャッシングヒット率を向上させるための技術を提供することである。

【 課題を解決するための手段 】

【 0 0 1 7 】

本発明の第1の態様に係る情報処理方法は、複数の情報処理装置によりデータの処理を行う分散処理システムにおいて実行される情報処理方法である。そして、本情報処理方法は、複数の情報処理装置のうちいずれかの情報処理装置である第1の情報処理装置が、第1の情報処理装置が管理する第1のデータに対するアクセスがあった場合に、複数の情報処理装置のうちデータ間の関連性を管理する情報処理装置から、第1のデータと所定の関連性を有する第2のデータの識別情報及び当該第2のデータを管理する情報処理装置の識別情報を取得し、第2のデータを管理する情報処理装置が第1の情報処理装置である場合に、第2のデータをデータ格納部から読み出し、キャッシュに展開する処理を含む。

## 【 0 0 1 8 】

本発明の第2の態様に係る情報処理方法は、複数の情報処理装置によりデータの処理を行う分散処理システムにおいて実行される情報処理方法である。そして、本情報処理方法は、複数の情報処理装置のうちいずれかの情報処理装置である第1の情報処理装置が、複数の情報処理装置のうち第1のデータを管理する情報処理装置から、第1のデータと所定の関連性を有するデータについての情報を要求する第1の要求を受信した場合に、データの識別情報と当該データと所定の関連性を有するデータの識別情報とを対応付けて格納する関連データ格納部から、第1のデータと所定の関連性を有する第2のデータの識別情報を抽出し、抽出された第2のデータの識別情報と当該第2のデータを管理する情報処理装置の識別情報とを、第1の要求の送信元の情報処理装置に送信する処理を含む。

10

## 【発明の効果】

## 【 0 0 1 9 】

分散処理システムにおけるキャッシュヒット率を向上させることができるようになる。

## 【図面の簡単な説明】

## 【 0 0 2 0 】

【図1】図1は、分散処理システムによるビッグデータの処理について説明するための図である。

【図2】図2は、本実施の形態におけるシステム概要を示す図である。

【図3】図3は、構成的関連性に基づき生成された関連ファイルデータの一例を示す図である。

20

【図4】図4は、統計的関連性に基づき生成された関連ファイルデータの一例を示す図である。

【図5】図5は、統計的関連性に基づく関連ファイルデータの生成について説明するための図である。

【図6】図6は、関連ファイルデータ格納部の詳細なデータ構造の一例を示す図である。

【図7】図7は、アプリケーションプログラムがファイルに対するアクセスを行う場合に行われる処理の処理フローを示す図である。

【図8A】図8Aは、先読み処理の処理フローを示す図である。

【図8B】図8Bは、先読み処理の処理フローを示す図である。

【図9】図9は、先読み処理について説明するための図である。

30

【図10】図10は、共有ディスク型の場合における先読み処理について説明するための図である。

【図11】図11は、アプリケーションプログラムがファイルに対するアクセスを行う場合に行われる処理の処理フローを示す図である。

【図12】図12は、破棄処理の処理フローを示す図である。

【図13】図13は、破棄処理について説明するための図である。

【図14】図14は、キャッシュヒット率の向上について説明するための図である。

【図15】図15は、ファイルを格納する場合に行われる処理の処理フローを示す図である。

【図16】図16は、ファイル格納要求について説明するための図である。

40

【図17】図17は、共有ディスク型の場合におけるファイル格納要求について説明するための図である。

【図18】図18は、ユーザが関連を指定した場合に行われる処理の処理フローを示す図である。

【図19】図19は、キャッシュのライフサイクルの一例を示す図である。

【図20】図20は、本実施の形態におけるシステム概要の他の例を示す図である。

【図21】図21は、コンピュータの機能ブロック図である。

## 【発明を実施するための形態】

## 【 0 0 2 1 】

図2に、本実施の形態における分散処理システムの概要を示す。例えばインターネット

50

であるネットワーク 5 には、管理サーバ 1 と、複数のサーバ 3 とが接続されている。図 2 に示した分散処理システムにおいては、サーバ 3 におけるアプリケーションプログラム 3 5 が、複数のサーバ 3 のうちいずれかのサーバ 3 が管理するファイルに対する処理を行う。

#### 【 0 0 2 2 】

管理サーバ 1 は、システム管理部 1 0 0 と、関連ファイルデータ格納部 1 0 8 と、ログ格納部 1 0 9 とを有する。システム管理部 1 0 0 は、通信制御部 1 0 1 と、システムデータ管理部 1 0 2 と、構成管理部 1 0 3 と、インデックス管理部 1 0 5、メタデータ管理部 1 0 6 及び統計データ管理部 1 0 7 を含むファイル管理部 1 0 4 とを有する。

#### 【 0 0 2 3 】

通信制御部 1 0 1 は、通信のための処理を行う。システムデータ管理部 1 0 2 は、分散処理システムにおいて共有するデータ（例えばファイルのブロックサイズ又は複製されたファイルの数等）を管理する。構成管理部 1 0 3 は、分散処理システムにおけるクラスタの構成の管理及び死活監視等を行う。ファイル管理部 1 0 4 は、関連ファイルデータ格納部 1 0 8 に格納されているデータ及びログ格納部 1 0 9 に格納されているデータを管理する処理等を行う。インデックス管理部 1 0 5 は、各サーバ 3 が管理しているファイルを二分木構造のファイルインデックスにより管理する。メタデータ管理部 1 0 6 は、ファイルのメタデータ（例えば、ファイルのサイズ及びファイルの所有者等）を管理する。統計データ管理部 1 0 7 は、ログ格納部 1 0 9 に格納されているデータを用いてファイル間の統計的関連を特定することにより、関連ファイルデータ格納部 1 0 8 に格納されているデータを更新する。

#### 【 0 0 2 4 】

図 3 に、関連ファイルデータ格納部 1 0 8 に格納される関連ファイルデータの一例を示す。図 3 の例では、関連ファイルデータは、ファイルの識別子とそのファイルに関連するファイル（以下、関連ファイルと呼ぶ）の識別子を含む。関連ファイルは、特定のファイルに対するアクセスがあった場合に、そのアクセスがあったタイミングの近傍のタイミングにおいてアクセスされる可能性が高いファイルである。

#### 【 0 0 2 5 】

関連性には、構成的関連性と統計的関連性とユーザの指定による関連性とがある。図 3 に示した関連ファイルデータは、構成的関連性に基づき生成された関連ファイルデータである。構成的関連性があるファイルとは、例えば、元は 1 つであったファイルを分割することにより得られた複数のファイル、同一又は近傍のディレクトリに格納されているファイル、又は同じタイムスタンプを有するファイル等である。構成的関連性があるファイルは、例えばバッチ処理のような一括処理において一遍にアクセスされる可能性が高い。図 3 に示した関連ファイルデータは、ファイル A - 1、ファイル A - 2 及びファイル A - 3 はファイル A を分割することにより得られたファイルであるため、これらのファイルが構成的関連を有していることを表している。また、ファイル B - 1 は、ファイル A - 1 と同一又は近傍のディレクトリに格納されているか又は同一のタイムスタンプを有している。

#### 【 0 0 2 6 】

図 4 に、統計的関連性に基づき生成された関連ファイルデータの一例を示す。図 4 の例では、関連ファイルデータは、ファイルの識別子と、関連ファイルの識別子と、関連ファイルにアクセスされた回数を示す数値とを含む。関連ファイルにアクセスされた回数は、対象のファイルに対するアクセスがあった後所定時間内に関連ファイルにアクセスがあった回数を表す。

#### 【 0 0 2 7 】

統計的関連性に基づき生成された関連ファイルデータは、図 5 に示すように、統計データ管理部 1 0 7 がログ格納部 1 0 9 に格納されているアクセスログに基づき生成する。図 5 の左側には、アクセスログが示されており、図 5 の右側には統計的関連性に基づき生成された関連ファイルデータが示されている。例えばバッチ処理等においては、所定の時刻にファイルにアクセスしたり、所定の順番でファイルにアクセスすることがある。そこで

10

20

30

40

50

、アクセスログを利用し、所定の条件（例えば、所定時間内のアクセスであるという条件、所定のアクセス順序であるという条件、又は所定の利用者によるアクセスであるという条件等）に合致すると判断された場合には、関連ファイルであるとみなす。

【0028】

なお、統計的関連性に基づく関連ファイルデータは、関連性が高い（例えば、アクセス回数が多い）関連ファイルほどチェーン構造における先頭に近づくように生成される。

【0029】

なお、ユーザの指定により関連性を定義すれば、アクセスパターンが固定されている場合に特に有効である。

【0030】

図6に、関連ファイルデータ格納部108の詳細なデータ構造の一例を示す。関連ファイルの情報は、分散ファイルシステムが管理するファイルの情報の属性の一つである。図6の例では、インデックス管理部105が管理するファイルインデックス50に、ファイル情報51へのリンクが張られている。ファイル情報51の構造体には、そのファイルの関連ファイルの情報へのポインタメンバが含まれる。

【0031】

関連ファイルデータ52はチェーン構造であり、関連ファイルの情報へのリンクが連結される。関連ファイルデータ52の構造体には、ファイル識別子と、関連付けられるファイル情報のアドレスと、関連ファイルの格納場所を示す情報と、アクセス回数を示すカウンタと、次の関連ファイルのアドレスとが含まれる。カウンタの値が大きいほどアクセスされる可能性が高い関連ファイルであるため、優先的にキャッシュ36に展開される。

【0032】

図2の説明に戻り、サーバ3は、クライアント管理部30と、アプリケーションプログラム35と、キャッシュ36と、ファイル格納部37とを有する。クライアント管理部30は、通信制御部31と、アクセス制御部33及び先読み処理部34を含むファイル管理部32とを有する。先読み処理部34は、取得部341と、キャッシュ処理部342とを有する。

【0033】

通信制御部31は、通信のための処理を実行する。アクセス制御部33は、ファイルアクセスを制御する処理を行う。取得部341は、関連ファイルの情報を管理サーバ1から取得する処理等を行う。キャッシュ処理部342は、関連ファイルをキャッシュ36に展開する処理等を行う。アプリケーションプログラム35は、ファイル格納部37に格納されているファイルに対する処理を行う。キャッシュ36は、例えばメモリ領域等に確保されたディスクキャッシュである。ファイル格納部37は、例えばハードディスク等に確保された、ファイルを格納するための領域である。

【0034】

次に、図7乃至図19を用いて、図2に示したシステムの動作について説明する。まず、図7乃至図14を用いて、アプリケーションプログラム35がファイルに対するアクセスを行う場合の動作について説明する。

【0035】

図2に示したシステムにおける複数のサーバ3のうちいずれかのサーバ3（ここでは、サーバAとする）におけるアプリケーションプログラム35は、特定のファイルに対するアクセスを行うとする。この場合、サーバAにおけるアプリケーションプログラム35は、アクセス対象のファイルの指定を含むファイルオープン要求を管理サーバ1に送信する（図7：ステップS1）。

【0036】

管理サーバ1における通信制御部101は、ファイルオープン要求をサーバAから受信すると（ステップS3）、ファイル管理部104に出力する。ファイル管理部104におけるインデックス管理部105は、ファイルオープン要求において指定されたファイルの格納場所をファイルインデックス50を用いて特定する（ステップS5）。インデックス

10

20

30

40

50



管理部 105 は、ファイルの格納場所を示すデータを通信制御部 101 に出力する。

【0037】

ファイル管理部 104 は、ログ格納部 109 に格納されているアクセスログ及び関連ファイルデータ格納部 108 に格納されているデータを更新する（ステップ S7）。具体的には、ファイル管理部 104 における統計データ管理部 107 が、今回のアクセスについてのデータをアクセスログに追加することでアクセスログを更新する。また、統計データ管理部 107 は、アクセスログの更新を反映するように関連ファイルデータ格納部 108 に格納されているデータを更新する。

【0038】

通信制御部 101 は、インデックス管理部 105 から受け取った、ファイルの格納場所を示すデータを、サーバ A に送信する（ステップ S9）。 10

【0039】

サーバ A におけるアプリケーションプログラム 35 は、ファイルの格納場所を示すデータを管理サーバ 1 から受信する（ステップ S11）。そして、アプリケーションプログラム 35 は、ファイルの格納場所（ここではサーバ B とする）を示すデータに従い、アクセス対象のファイルの指定を含むファイルオープン要求をサーバ B に送信する（ステップ S13）。

【0040】

サーバ B は、ファイルオープン要求をサーバ A から受信する（ステップ S15）。そして、サーバ B におけるアクセス制御部 33 は、ファイルオープン要求において指定されたファイルのオープン処理（すなわち、ファイルの使用を開始するための処理）を実行し、ファイルをキャッシュ 36 に展開する（ステップ S17）。なお、ファイルオープン要求において指定されたファイルは、ファイル格納部 37 から読み出される。 20

【0041】

そして、先読み処理部 34 は、先読み処理を実行する（ステップ S19）。先読み処理については、図 8A 乃至図 10 を用いて説明する。

【0042】

まず、先読み処理部 34 は、アクセス対象のファイルについて関連ファイルリストを要求する関連ファイル要求を管理サーバ 1 に送信する（図 8A：ステップ S61）。 30

【0043】

図 8B を用いて、関連ファイル要求を受信した管理サーバ 1 の処理について説明する。管理サーバ 1 における通信制御部 101 は、関連ファイル要求をサーバ B から受信する（図 8B：ステップ S62）。通信制御部 101 は、関連ファイル要求をファイル管理部 104 に出力する。

【0044】

ファイル管理部 104 は、アクセス対象のファイルに対応する関連ファイルの識別子及び当該関連ファイルの格納場所（すなわち、関連ファイルを管理するサーバの識別子）を関連ファイルデータ格納部 108 から抽出する（ステップ S63）。ファイル管理部 104 は、関連ファイルの識別子及び関連ファイルを管理するサーバの識別子を通信制御部 101 に出力する。なお、関連ファイルの識別子及び当該関連ファイルを管理するサーバの識別子は複数である場合もある。また、関連ファイルが無い場合もある。 40

【0045】

通信制御部 101 は、関連ファイルの識別子及び当該関連ファイルを管理するサーバの識別子を含む関連ファイルリストをサーバ B に送信する（ステップ S64）。そして処理を終了する。

【0046】

これに応じ、サーバ B における先読み処理部 34 は、関連ファイルリストを管理サーバ 1 から受信し（ステップ S65）、メインメモリ等の記憶装置に格納する。

【0047】

先読み処理部 34 は、関連ファイルリストに未処理の関連ファイルが有るか判断する（ 50

ステップS 6 6)。未処理の関連ファイルが無い場合（ステップS 6 6：N o ルート）、元の処理に戻る。未処理の関連ファイルが有る場合（ステップS 6 6：Y e s ルート）、先読み処理部 3 4 は、未処理の関連ファイルを 1 つ特定し、当該関連ファイルが自サーバ（すなわちサーバ B）に有るか判断する（ステップS 6 7）。ステップS 6 7 の判断は、関連ファイルを管理するサーバの識別子が自サーバを示しているか否かにより行う。なお、ステップS 6 7 において特定された関連ファイルを「処理対象の関連ファイル」と呼ぶ。

#### 【 0 0 4 8 】

処理対象の関連ファイルが自サーバに無い場合（ステップS 6 7：N o ルート）、先読み処理部 3 4 は、処理対象の関連ファイルを管理するサーバ（ここでは、サーバ C とする）に先読み処理要求を送信する（ステップS 6 9）。ステップS 6 9 の先読み処理要求に応じ、サーバ C は、先読み処理（ステップS 1 9）を実行する。そして処理は端子 D を介してステップS 6 6 に戻る。

#### 【 0 0 4 9 】

一方、処理対象の関連ファイルが自サーバに有る場合（ステップS 6 7：Y e s ルート）、先読み処理部 3 4 は、キャッシュ 3 6 の使用量が閾値以下であるか判断する（ステップS 7 1）。キャッシュ 3 6 の使用量が閾値以下ではない場合（ステップS 7 1：N o ルート）、キャッシュ溢れが発生する可能性があるので、処理対象の関連ファイルをキャッシュ 3 6 に展開せず、元の処理に戻る。キャッシュ 3 6 の使用量が閾値以下である場合（ステップS 7 1：Y e s ルート）、先読み処理部 3 4 は、処理対象の関連ファイルが既にキャッシュ 3 6 に展開されているか判断する（ステップS 7 3）。

#### 【 0 0 5 0 】

処理対象の関連ファイルが既にキャッシュ 3 6 に展開されている場合（ステップS 7 3：Y e s ルート）、処理対象の関連ファイルをキャッシュ 3 6 に展開しなくてもよいので、処理は端子 D を介してステップS 6 6 の処理に戻る。一方、処理対象の関連ファイルがキャッシュ 3 6 に展開されていない場合（ステップS 7 3：N o ルート）、先読み処理部 3 4 は、アクセス制御部 3 3 に処理対象の関連ファイルのオープン処理を実行することを要求する。これに応じ、アクセス制御部 3 3 は、処理対象の関連ファイルのオープン処理を実行する（ステップS 7 5）。そして処理は端子 D を介してステップS 6 6 に戻る。

#### 【 0 0 5 1 】

以上のような処理を実行すれば、特定のファイルに対してアクセスがあった場合に、そのファイルに関連するファイルがキャッシュ 3 6 に展開されるようになる。関連ファイルは、特定のファイルに対するアクセスがあった場合にアクセスされる可能性が高いファイルであるから、分散処理システムにおけるキャッシュヒット率を向上させることができるようになる。

#### 【 0 0 5 2 】

図 9 を用いて、先読み処理についてより具体的に説明する。例えば、サーバ A におけるアプリケーションプログラム 3 5 がファイル A に対する分散処理を要求したとする（図 9 における（1））。ファイル A は、分散ファイルシステムによってファイル A - 1、A - 2、・・・、A - n の n 個（n は 2 以上の自然数）に分割されており、分割されたファイルはサーバ A 乃至 C に分散して格納されている。ここで、ファイル A - 1 がサーバ A のファイル格納部 3 7 からキャッシュ 3 6 に展開され、アプリケーションプログラム 3 5 による処理に供される（図 9 における（2））。この際、サーバ B におけるファイル A - 2 及びファイル A - 3、並びにサーバ C におけるファイル A - n もキャッシュ 3 6 に展開される（図 9 における（3））。そのため、アプリケーションプログラム 3 5 がファイル A - 2 乃至 A - n について処理する際には、これらのファイルは事前にキャッシュ 3 6 に展開されているので、ディスク I / O は発生しない。従って、ファイル A の処理を高速で行うことができるようになる。

#### 【 0 0 5 3 】

なお、上で述べた例では、各サーバがファイル格納部 3 7 を有するとして説明したが、

図 10 に示すように、複数のサーバに対して共有のファイル格納部 37 を 1 つ設けるような場合（以下、共有ディスク型の場合と呼ぶ）にも先読み処理は有効である。図 10 の例では、ファイル格納部 37 が 3 つの部分に分けられている。この場合、ファイル A - 1 乃至 A - 3 のいずれかに対するアクセスがあった場合には、サーバ A 乃至サーバ C のうちいずれかのサーバが、共有のファイル格納部 37 からファイルを読み出し、キャッシュ 36 に展開する。これにより、ディスク I/O は発生しなくなるので、ファイル A の処理を高速で行うことができるようになる。なお、キャッシュ 36 への展開を行なうサーバは、例えばラウンドロビン或いはキャッシュ 36 の使用状況等に基づき分散ファイルシステムにより決定される。

【 0 0 5 4 】

10

図 7 の説明に戻り、サーバ B における通信制御部 31 は、オープン処理の完了を示す通知をサーバ A に送信する（ステップ S 21）。

【 0 0 5 5 】

サーバ A におけるアプリケーションプログラム 35 は、オープン処理の完了を示す通知を受信すると（ステップ S 23）、オープン処理が行われたファイルに対するファイルアクセス要求（例えば、参照要求）をサーバ B に送信する（ステップ S 25）。

【 0 0 5 6 】

サーバ B における通信制御部 31 は、ファイルアクセス要求を受信する（ステップ S 27）。通信制御部 31 は、ファイルアクセス要求をファイル管理部 32 に出力する。処理は端子 A 乃至 C を介して図 11 の処理に移行する。

20

【 0 0 5 7 】

図 11 の説明に移行し、ファイル管理部 32 におけるアクセス制御部 33 は、ファイル格納部 37 に格納されているファイルに対するアクセスを行う（ステップ S 29）。アクセス制御部 33 は、アクセス結果（例えば、ファイルアクセス要求が参照要求である場合には、参照した結果）を通信制御部 31 に出力する。

【 0 0 5 8 】

通信制御部 31 は、アクセス結果をファイルアクセス要求の送信元であるサーバ A に送信する（ステップ S 31）。

【 0 0 5 9 】

サーバ A におけるアプリケーションプログラム 35 は、アクセス結果を受信する（ステップ S 33）。その後、アプリケーションプログラム 35 は、アクセス対象のファイルについてファイルクローズ要求をサーバ B に送信する（ステップ S 35）。

30

【 0 0 6 0 】

サーバ B における通信制御部 31 は、ファイルクローズ要求を受信する（ステップ S 37）。通信制御部 31 は、ファイルクローズ要求をファイル管理部 32 に出力する。

【 0 0 6 1 】

ファイル管理部 32 におけるアクセス制御部 33 は、アクセス対象のファイルについてクローズ処理（すなわち、ファイルの使用を終了するための処理）を実行する（ステップ S 39）。また、ファイル管理部 32 における先読み処理部 34 は、破棄処理を実行する（ステップ S 41）。破棄処理については、図 12 及び図 13 を用いて説明する。

40

【 0 0 6 2 】

まず、先読み処理部 34 は、キャッシュ 36 に展開されているアクセス対象のファイルを削除する（図 12：ステップ S 81）。

【 0 0 6 3 】

先読み処理部 34 は、ステップ S 65 において受信した関連ファイルリストに、自サーバが管理する関連ファイルであり且つ未処理の関連ファイルが有るか判断する（ステップ S 83）。

【 0 0 6 4 】

未処理の関連ファイルが無い場合（ステップ S 85：No ルート）、元の処理に戻る。一方、未処理の関連ファイルが有る場合（ステップ S 85：Yes ルート）、先読み処理

50

部 3 4 は、未処理の関連ファイルを 1 つ特定する（以下では、特定されたファイルを「処理対象の関連ファイル」と呼ぶ）。そして、先読み処理部 3 4 は、キャッシュ 3 6 の使用量が閾値以下であるか判断する（ステップ S 8 7）。

【 0 0 6 5 】

キャッシュ 3 6 の使用量が閾値以下ではない場合（ステップ S 8 7：N o ルート）、処理対象の関連ファイルをキャッシュ 3 6 に展開することはできないので、元の処理に戻る。一方、キャッシュ 3 6 の使用量が閾値以下である場合（ステップ S 8 7：Y e s ルート）、先読み処理部 3 4 は、処理対象の関連ファイルが既にキャッシュ 3 6 に展開されているか判断する（ステップ S 8 9）。

【 0 0 6 6 】

処理対象の関連ファイルが既にキャッシュ 3 6 に展開されている場合（ステップ S 8 9：Y e s ルート）、処理対象の関連ファイルをキャッシュ 3 6 に展開しなくてもよいので、ステップ S 8 3 の処理に戻る。一方、処理対象の関連ファイルがキャッシュ 3 6 に展開されていない場合（ステップ S 8 9：N o ルート）、先読み処理部 3 4 は、アクセス制御部 3 3 に処理対象の関連ファイルのオープン処理を実行することを要求する。これに応じ、アクセス制御部 3 3 は、処理対象の関連ファイルのオープン処理を実行する（ステップ S 9 1）。そしてステップ S 8 3 に戻る。

【 0 0 6 7 】

以上のような処理を実行すれば、クローズ処理が完了したファイルについてはキャッシュ 3 6 から削除されるので、キャッシュ 3 6 を無駄に使用することが無くなる。また、図 1 3 に示すように、キャッシュ 3 6 の使用量は閾値以下になるように制御されている。図 1 3 に示す例においては、処理済みであるファイル A - 1 がキャッシュ 3 6 から削除されており、代わりに関連ファイルであるファイル A - 4 がキャッシュ 3 6 に展開される。このようにすることで、未処理の関連ファイルがキャッシュ 3 6 に展開されやすくなるので、キャッシュヒット率をさらに向上させることができるようになる。

【 0 0 6 8 】

図 1 1 の説明に戻り、サーバ B における通信制御部 3 1 は、クローズ処理の完了を示す通知をサーバ A に送信する（ステップ S 4 3）。

【 0 0 6 9 】

サーバ A におけるアプリケーションプログラム 3 5 は、クローズ処理の完了を示す通知をサーバ B から受信する（ステップ S 4 5）。そして、アプリケーションプログラム 3 5 は、アクセス対象のファイルについてファイルクローズ要求を管理サーバ 1 に送信する（ステップ S 4 7）。

【 0 0 7 0 】

管理サーバ 1 における通信制御部 1 0 1 は、サーバ A からファイルクローズ要求を受信し（ステップ S 4 9）、ファイル管理部 1 0 4 に出力する。そして、ファイル管理部 1 0 4 は、アクセス対象のファイルのステータス（ここでは、状態についての情報を意味する）を更新する（ステップ S 5 1）。具体的には、アクセス対象のファイルについて、オープン又はクローズの状態、ロックの状態、ファイルサイズ、又は更新の日時等を更新し、図示しない記憶装置に記憶する。

【 0 0 7 1 】

以上のような処理を実行すれば、分散処理システムにおけるキャッシュヒット率を向上させ、ファイルに対する処理を高速で行うことができるようになる。

【 0 0 7 2 】

通常、キャッシュはアプリケーションプログラムの都合とは無関係にリフレッシュされる。しかし、本実施の形態においては、関連性があるファイルがキャッシュに残りやすいため、高速にファイルの処理をすることを期待できる。図 1 4 を用いて、図 1 と同様の状況において本実施の形態の処理を行なった場合について説明する。本実施の形態の処理によれば、アプリケーションプログラム 3 5 がサーバ B にアクセスした際にはファイル B がキャッシュ 3 6 に展開されている可能性が高く、またアプリケーションプログラム 3 5 が

10

20

30

40

50

サーバCにアクセスした際にはファイルCがキャッシュ36に展開されている可能性が高い。よって、ファイルB及びファイルCについてディスクI/Oが発生する可能性は低くなるので、ファイルに対する処理の完了までの時間を短くすることができるようになる。

【0073】

なお、データ処理を高速化する方法として、OSのキャッシュとは別にアプリケーションプログラムのためのメモリ領域（以下、アプリケーションキャッシュと呼ぶ）を確保し、そこに予めデータを展開しておきアクセスするという方法が知られている。この方法によれば、独自のメモリ領域であればデータの取捨選択をも完全にコントロールできるため、効率的なアクセスが可能になる。但し、アプリケーションキャッシュを設けたとしてもOSのキャッシュが無くなるわけではないので、メモリ領域を2倍使用することになる。

10

【0074】

これに対し、本実施の形態において説明した方法であれば、アプリケーションキャッシュを利用する方法と比べると、メモリの使用量は1/2程度に節約できる。

【0075】

次に、図15乃至図17を用いて、ファイルを格納する際の動作について説明する。なお、図15のフローは、処理の要点を示しており、通常の分散ファイルシステムが行う処理（例えば、ステータス更新、ファイルのロック及びクラスタ構成の管理等）については省略している。

【0076】

20

まず、サーバ3（ここでは、サーバAとする）におけるファイル管理部32は、例えばアプリケーションプログラム35によりファイルが生成された場合に、当該ファイルを分散して格納することを要求するファイル格納要求を生成する。そして、ファイル管理部32は、ファイル格納要求を管理サーバ1に送信する（図15：ステップS101）。なお、ファイル格納要求には、ファイルそのものを含ませるようにしてもよいし、ファイルサイズの情報等を含ませるようにしてもよい。

【0077】

管理サーバ1における通信制御部101は、ファイル格納要求を受信すると（ステップS103）、ファイル格納要求をファイル管理部104に出力する。

【0078】

30

ファイル管理部104は、ファイル格納要求において格納が要求されているファイルの格納場所を決定する（ステップS105）。例えば、ファイル格納部37の空き容量が多いサーバをファイルの格納場所とする。また、ファイルサイズが大きい場合にはファイルを分割することになるため、複数のサーバをファイルの格納場所に決定する。

【0079】

ファイル管理部104は、関連ファイルデータ格納部108に格納されているデータを更新する（ステップS107）。ステップS107においては、構成的関連に基づき関連ファイルデータを更新する。例えばファイルが特定のディレクトリに格納される場合には、そのディレクトリに格納されている他のファイルと構成的関連性を有することになるので、新たな構成的関連性に基づき更新を行う。また、ファイルが分割される場合には、分割後の複数のファイルは構成的関連性を有することになるので、新たな構成的関連性に基づき更新を行う。

40

【0080】

ファイル管理部104は、ファイルの格納場所を示すデータを通信制御部101に出力する（ステップS109）。そして、通信制御部101は、ファイルの格納場所を示すデータ（ここでは、サーバBを示しているとする）をサーバAに送信する。

【0081】

サーバAにおけるファイル管理部32は、ファイルの格納場所を示すデータを受信すると（ステップS111）、ファイルの格納場所を示すデータに従い、ファイル格納要求をサーバBに送信する（ステップS113）。ステップS113において送信するファイル

50

格納要求には、ファイルが含まれる。なお、ファイルの格納場所となるサーバが複数ある場合には、複数のサーバに対してファイル格納要求を送信する。

【0082】

サーバBにおける通信制御部31は、ファイル格納要求を受信する(ステップS115)。通信制御部31は、ファイル格納要求をアクセス制御部33に出力する。

【0083】

アクセス制御部33は、ファイル格納要求に含まれるファイルをファイル格納部37に格納する(ステップS117)。

【0084】

以上のような処理を実行すれば、ファイルをファイル格納部37に格納する際には、ファイルの構成的関連性に基づき関連ファイルデータを更新できるようになる。

【0085】

図16を用いて、ファイルの構成的関連性に基づく関連ファイルデータの更新について具体的に説明する。例えば、いずれかのサーバにおけるアプリケーションプログラム35がファイルAについてファイル格納要求を出力したとする。ファイルAは、分散ファイルシステムによってファイルA-1、ファイルA-2及びファイルA-3の3つに分割され、分割されたファイルはサーバA乃至Cに分散して格納される。また、ファイルA-1はサーバAのファイル格納部37においてファイルB-1と同じディレクトリに格納されており、ファイルA-2はサーバBのファイル格納部37においてファイルB-2と同じディレクトリに格納されており、ファイルA-3はサーバCのファイル格納部37においてファイルB-3と同じディレクトリに格納されているとする。このような場合、管理サーバ1における関連ファイルデータ格納部108は、例えば図16の160に示した状態に更新される。この例においては、ファイルA-1の関連ファイルはファイルA-2、ファイルA-3及びファイルB-1であり、ファイルA-2の関連ファイルはファイルA-1、ファイルA-3及びファイルB-2であり、ファイルA-3の関連ファイルはファイルA-1、ファイルA-2及びファイルB-3である。

【0086】

なお、図16においては、ファイルA-1の関連ファイルにファイルB-2及びファイルB-3が含まれておらず、同じディレクトリに格納されているファイルB-1に限定されている。これは、ファイルB-1のスプリットファイル(ここでは、ファイルB-2及びファイルB-3)まで関連ファイルとすると、ファイルA-1の関連ファイルの数が膨大になるからである。ファイルA-2及びファイルA-3についても同様である。

【0087】

また、共有ディスク型の場合においても、ファイルの構成的関連性に基づく関連ファイルデータの更新を同様に行なうことができる。例えば図17に示すように、複数のサーバに対して共有のファイル格納部37が1つ設けられており、ファイル格納部37が3つの部分に分けられているとする。ファイルAは、分散ファイルシステムによってファイルA-1、ファイルA-2及びファイルA-3の3つに分割され、分割されたファイルは共有のファイル格納部37における部分p1乃至p3に分散して格納される。また、ファイルA-1はファイル格納部37の部分p1においてファイルB-1と同じディレクトリに格納されており、ファイルA-2はファイル格納部37の部分p2においてファイルB-2と同じディレクトリに格納されており、ファイルA-3はファイル格納部37の部分p3においてファイルB-3と同じディレクトリに格納されているとする。このような場合にも、管理サーバ1における関連ファイルデータ格納部108は、図17の170に示した状態に更新される。すなわち、ファイルA-1の関連ファイルはファイルA-2、ファイルA-3及びファイルB-1であり、ファイルA-2の関連ファイルはファイルA-1、ファイルA-3及びファイルB-2であり、ファイルA-3の関連ファイルはファイルA-1、ファイルA-2及びファイルB-3である。

【0088】

次に、図18を用いて、ユーザが関連を指定する場合の動作について説明する。

## 【 0 0 8 9 】

まず、ユーザは、対象ファイル及び当該対象ファイルに関連する関連ファイルをサーバ 3 における入力部（例えばキーボード又はマウス等）を使用して入力する。サーバ 3 におけるファイル管理部 3 2 は、対象ファイル及び関連ファイルの指定を受け付ける。そして、ファイル管理部 3 2 は、対象ファイル及び関連ファイルの指定を含む更新要求を管理サーバ 1 に送信する（図 1 8：ステップ S 1 2 1）。

## 【 0 0 9 0 】

管理サーバ 1 における通信制御部 1 0 1 は、サーバ 3 から更新要求を受信し（ステップ S 1 2 3）、ファイル管理部 1 0 4 に出力する。

## 【 0 0 9 1 】

ファイル管理部 1 0 4 は、更新要求において指定されている対象ファイルについての関連ファイルデータを関連ファイルデータ格納部 1 0 8 から特定する（ステップ S 1 2 5）。そして、ファイル管理部 1 0 4 は、更新要求におけるユーザの指定に従い関連ファイルデータを生成し、特定された関連ファイルデータを生成された関連ファイルデータで更新する（ステップ S 1 2 7）。なお、指定された対象ファイルについての関連ファイルデータが関連ファイルデータ格納部 1 0 8 に無い場合には、生成された関連ファイルデータを関連ファイルデータ格納部 1 0 8 に格納する。

## 【 0 0 9 2 】

ファイル管理部 1 0 4 は、更新完了を示す通知を通信制御部 1 0 1 に出力する。通信制御部 1 0 1 は、更新完了を示す通知をサーバ 3 に送信する（ステップ S 1 2 9）。

## 【 0 0 9 3 】

サーバ 3 における通信制御部 3 1 は、更新完了を示す通知を管理サーバ 1 から受信する（ステップ S 1 3 1）。通信制御部 3 1 は、更新完了を示す通知をファイル管理部 3 2 に出力する。

## 【 0 0 9 4 】

ファイル管理部 3 2 における先読み処理部 3 4 は、先読み処理を実行する（ステップ S 1 3 3）。先読み処理については、図 8 A 乃至図 1 0 を用いて説明をしたとおりである。なお、このようなタイミングで先読み処理を実行するのは、ユーザが関連を指定する場合には即ファイルに対する処理を行なう場合が多いと考えられるからである。

## 【 0 0 9 5 】

以上のような処理を実行すれば、ユーザの意図を関連ファイルデータ格納部 1 0 8 に適切に反映することができるようになる。

## 【 0 0 9 6 】

図 1 9 に、キャッシュのライフサイクルの一例を示す。図 1 9 の例は、サーバ A 及びサーバ B におけるキャッシュ 3 6 のライフサイクルを示している。図 1 9 において、キャッシュ 3 6 にファイルが展開されている時間を、白色の長方形で表す。

## 【 0 0 9 7 】

例えばサーバ A における特定のファイルに対するアクセスがあった場合、サーバ A における先読み処理部 3 4 は、管理サーバ 1 に関連ファイル要求を送信する（図 1 9：1 0 0 1）。これに応じ、管理サーバ 1 におけるファイル管理部 1 0 4 は、関連ファイルリストをサーバ A に送信する（図 1 9：1 0 0 2）。ここで、関連ファイルリストには、サーバ A が管理するファイルの識別情報と、サーバ B が管理するファイルの識別情報とが含まれるとする。

## 【 0 0 9 8 】

サーバ A における先読み処理部 3 4 は、サーバ B に対し、先読み処理要求を送信する（図 1 9：1 0 0 4）。サーバ B における先読み処理部 3 4 は、先読み処理要求を受信すると、応答を返す（図 1 9：1 0 0 5）。

## 【 0 0 9 9 】

サーバ B における先読み処理部 3 4 は、関連ファイル要求を管理サーバ 1 に送信する（図 1 9：1 0 0 6）。これに応じ、管理サーバ 1 におけるファイル管理部 1 0 4 は、関連

10

20

30

40

50

ファイルリストをサーバ B に送信する (図 19 : 1007)。

【0100】

一方、サーバ A における先読み処理部 34 は、サーバ A が管理する関連ファイルをキャッシュ 36 に展開する (図 19 : 1008)。また、サーバ B における先読み処理部 34 は、サーバ B が管理する関連ファイルをキャッシュ 36 に展開する (図 19 : 1009)。

【0101】

サーバ A におけるアプリケーションプログラム 35 は、ファイルクローズ要求を管理サーバ 1 に送信する (図 19 : 1010)。管理サーバ 1 におけるファイル管理部 104 は、ファイルクローズ要求を受信すると、ファイルのステータスの更新等を行い、クローズの完了をサーバ A に通知する (図 19 : 1011)。

10

【0102】

サーバ A における先読み処理部 34 は、クローズの完了を管理サーバ 1 から通知されたことを契機として、キャッシュ 36 に展開されているファイルを削除する (図 19 : 1012)。また、先読み処理部 34 は、残りの関連ファイルをキャッシュ 36 に展開する (図 19 : 1013)。

【0103】

一方、サーバ B におけるアプリケーションプログラム 35 は、ファイルクローズ要求を管理サーバ 1 に送信する (図 19 : 1014)。管理サーバ 1 におけるファイル管理部 104 は、ファイルクローズ要求を受信すると、ファイルのステータスの更新等を行い、クローズの完了をサーバ B に通知する (図 19 : 1015)。

20

【0104】

サーバ B における先読み処理部 34 は、クローズの完了を管理サーバ 1 から通知されたことを契機として、キャッシュ 36 に展開されているファイルを削除する (図 19 : 1016)。また、先読み処理部 34 は、残りの関連ファイルをキャッシュ 36 に展開する (図 19 : 1017)。

【0105】

以上のような処理が繰り返し行われることで、分散処理システムにおいてファイルに対する処理が進行する。

【0106】

30

以上本発明の一実施の形態を説明したが、本発明はこれに限定されるものではない。例えば、上で説明した管理サーバ 1 及びサーバ 3 の機能ブロック構成は実際のプログラムモジュール構成に一致しない場合もある。

【0107】

また、上で説明した各テーブルの構成は一例であって、上記のような構成でなければならないわけではない。さらに、処理フローにおいても、処理結果が変わらなければ処理の順番を入れ替えることも可能である。さらに、並列に実行させるようにしても良い。

【0108】

なお、上で述べた例では、ファイルアクセス要求として参照要求を示したが、参照要求に限られるわけではない。例えば、ファイルに対する書き込み要求であってもよい。

40

【0109】

また、図 7 及び図 11 に示したフローにおいては、サーバ B が管理サーバ 1 から関連ファイルリストを取得する例を示した。しかし、このような例に限られるわけではない。例えば、サーバ A が管理サーバ 1 からファイルの格納場所を示す情報を受信する際に関連ファイルリストを受信し、サーバ A が関連ファイルリストをサーバ B に送信するようにしてもよい。

【0110】

また、上では管理サーバ 1 がサーバ 3 の動作を行う例を示していないが、管理サーバ 1 がサーバ 3 の動作をも行なうようにしてもよい。

【0111】

50



また、上で述べたように、共有ディスク型の場合であっても、本実施の形態の処理を適用可能である。共有ディスク型の場合のシステムは、例えば図20に示すようなシステムであればよい。具体的には、複数のサーバ3に対して1つのファイル格納部70を設ければよい。

#### 【0112】

なお、上で述べた管理サーバ1及びサーバ3は、コンピュータ装置であって、図21に示すように、メモリ2501とCPU(Central Processing Unit)2503とハードディスク・ドライブ(HDD:Hard Disk Drive)2505と表示装置2509に接続される表示制御部2507とリムーバブル・ディスク2511用のドライブ装置2513と入力装置2515とネットワークに接続するための通信制御部2517とがバス2519で接続されている。オペレーティング・システム(OS:Operating System)及び本実施例における処理を実施するためのアプリケーション・プログラムは、HDD2505に格納されており、CPU2503により実行される際にはHDD2505からメモリ2501に読み出される。CPU2503は、アプリケーション・プログラムの処理内容に応じて表示制御部2507、通信制御部2517、ドライブ装置2513を制御して、所定の動作を行わせる。また、処理途中のデータについては、主としてメモリ2501に格納されるが、HDD2505に格納されるようにしてもよい。本発明の実施例では、上で述べた処理を実施するためのアプリケーション・プログラムはコンピュータ読み取り可能なリムーバブル・ディスク2511に格納されて頒布され、ドライブ装置2513からHDD2505にインストールされる。インターネットなどのネットワーク及び通信制御部2517を経由して、HDD2505にインストールされる場合もある。このようなコンピュータ装置は、上で述べたCPU2503、メモリ2501などのハードウェアとOS及びアプリケーション・プログラムなどのプログラムとが有機的に協働することにより、上で述べたような各種機能を実現する。

#### 【0113】

以上述べた本発明の実施の形態をまとめると、以下のようになる。

#### 【0114】

本実施の形態の第1の態様に係る情報処理方法は、複数の情報処理装置によりデータの処理を行う分散処理システムにおいて実行される情報処理方法である。そして、本情報処理方法は、複数の情報処理装置のうちいずれかの情報処理装置である第1の情報処理装置が、(A)第1の情報処理装置が管理する第1のデータに対するアクセスがあった場合に、複数の情報処理装置のうちデータ間の関連性を管理する情報処理装置から、第1のデータと所定の関連性を有する第2のデータの識別情報及び当該第2のデータを管理する情報処理装置の識別情報を取得し、(B)第2のデータを管理する情報処理装置が第1の情報処理装置である場合に、第2のデータをデータ格納部から読み出し、キャッシュに展開する処理を含む。

#### 【0115】

第1のデータと所定の関連性を有する第2のデータがアクセスされるタイミングは、第1のデータがアクセスされるタイミングの近傍である可能性が高い。そこで、上で述べたようにすれば、分散処理システムにおけるキャッシュヒット率を向上させることができるようになる。

#### 【0116】

また、上で述べた本情報処理方法が、(C)第2のデータを管理する情報処理装置が第1の情報処理装置ではない場合に、複数の情報処理装置のうち第2のデータを管理する情報処理装置に対し、第2のデータをキャッシュに展開することを要求する第1の要求を送信する処理をさらに含むようにしてもよい。このようにすれば、第1の情報処理装置が第2のデータを管理していない場合にも対処できるようになる。

#### 【0117】

また、上で述べた本情報処理方法が、(D)第1のデータに対するアクセスが終了した場合に、キャッシュに展開されている第1のデータを削除する処理をさらに含むようにし

てもよい。このようにすれば、キャッシュ溢れの発生を抑制できるようになる。

【 0 1 1 8 】

また、上で述べた所定の関連性が、データの内容の関連性、データの格納場所に基づく関連性、データのタイムスタンプに基づく関連性、データに対するアクセスの履歴に基づく関連性及びユーザにより指定された関連性の少なくともいずれかであってもよい。

【 0 1 1 9 】

本実施の第2の態様に係る情報処理方法は、複数の情報処理装置によりデータの処理を行う分散処理システムにおいて実行される情報処理方法である。そして、本情報処理方法は、複数の情報処理装置のうちいずれかの情報処理装置である第1の情報処理装置が、(E) 複数の情報処理装置のうち第1のデータを管理する情報処理装置から、第1のデータと所定の関連性を有するデータについての情報を要求する第1の要求を受信した場合に、データの識別情報と当該データと所定の関連性を有するデータの識別情報とを対応付けて格納する関連データ格納部から、第1のデータと所定の関連性を有する第2のデータの識別情報を抽出し、(F) 抽出された第2のデータの識別情報と当該第2のデータを管理する情報処理装置の識別情報とを、第1の要求の送信元の情報処理装置に送信する処理を含む。

10

【 0 1 2 0 】

このようにすれば、第1のデータにアクセスがあった場合に第2のデータをキャッシュに展開できるようになるので、分散処理システムにおけるキャッシュヒット率を向上させることができるようになる。

20

【 0 1 2 1 】

また、上で述べた本情報処理方法が、(G) データの内容の関連性、データの格納場所及びデータのタイムスタンプの少なくともいずれかに基づき、関連データ格納部に格納されているデータを更新する処理をさらに含むようにしてもよい。このような事項はデータの関連性に関わっているため、このような事項に基づき更新を行えば、関連性を有するデータとして適切なデータを特定できるようになる。

【 0 1 2 2 】

また、上で述べた本情報処理方法が、(H) 分散処理システムにおけるデータへのアクセスの履歴に基づき、関連データ格納部に格納されているデータを更新する処理をさらに含むようにしてもよい。例えばアクセスの順序或いはアクセスパターン等はデータの関連性に関わっているため、アクセスの履歴に基づき更新を行えば、関連性を有するデータとして適切なデータを特定できるようになる。

30

【 0 1 2 3 】

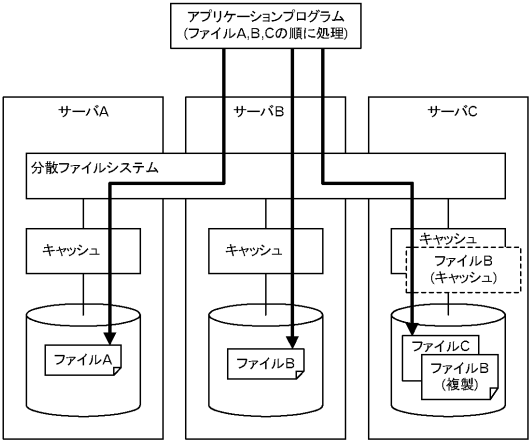
また、上で述べた本情報処理方法が、(I) ユーザからの指定を受け付けた場合に、当該指定に従い、関連データ格納部に格納されているデータを更新する処理をさらに含むようにしてもよい。このようにすれば、ユーザの意図どおりにキャッシュへの展開を行えるようになる。

【 0 1 2 4 】

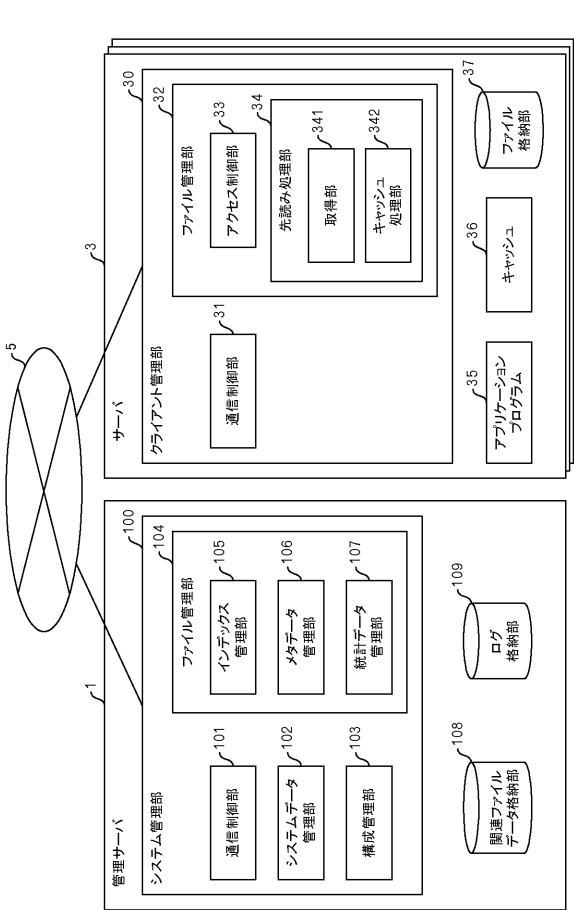
なお、上記方法による処理をコンピュータに行わせるためのプログラムを作成することができ、当該プログラムは、例えばフレキシブルディスク、CD-ROM、光磁気ディスク、半導体メモリ、ハードディスク等のコンピュータ読み取り可能な記憶媒体又は記憶装置に格納される。尚、中間的な処理結果はメインメモリ等の記憶装置に一時保管される。

40

【図 1】



【図 2】



【図 3】

ファイル	関連ファイル
A-1	A-2,A-3,B-1
A-2	A-1,A-3,B-2
A-3	A-1,A-2,B-3
⋮	⋮

【図 5】

時刻	ファイル	ユーザ
12:00.000	A	ユーザ1
12:10.000	B	ユーザ1
12:12.000	A	ユーザ1
12:13.000	C	ユーザ2
14:30.000	D	ユーザ1
⋮	⋮	⋮
14:30.000	Z	ユーザ3

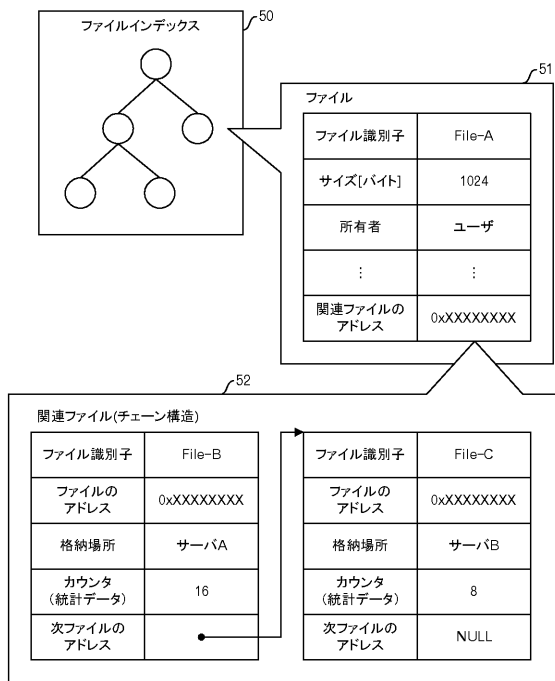
→

ファイル	関連ファイル
A	B(a回),C(b回)
B	A(c回)
C	D(d回)
⋮	⋮

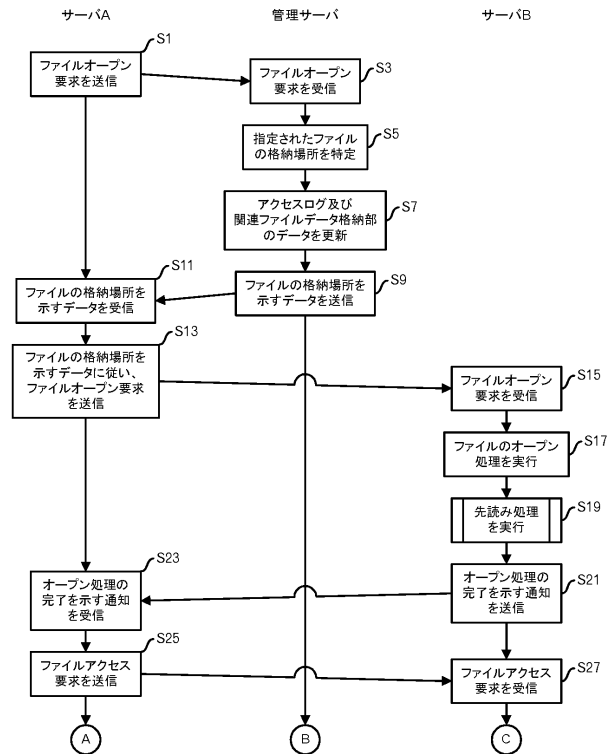
【図 4】

ファイル	関連ファイル
A	B(a回),C(b回)
B	A(c回)
C	D(d回)
⋮	⋮

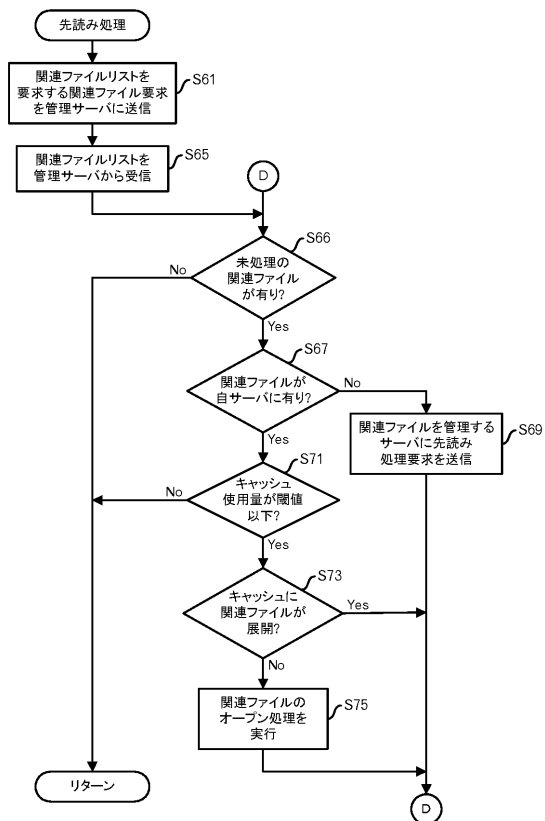
【 図 6 】



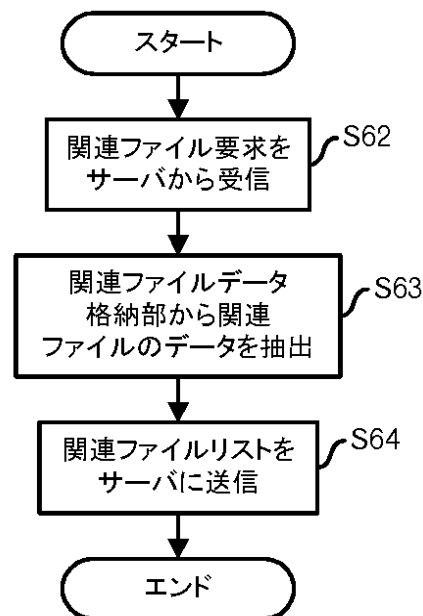
【圖 7】



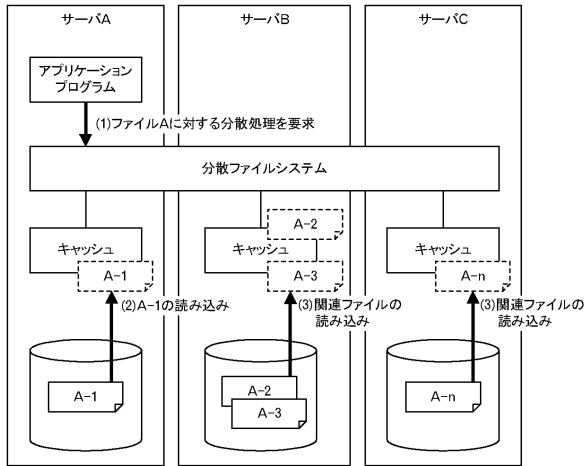
【 図 8 A 】



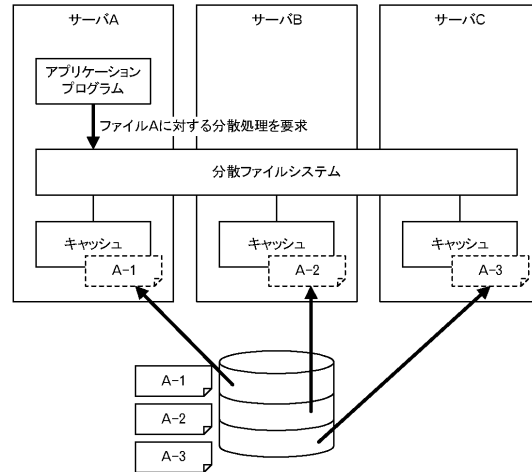
【 図 8 B 】



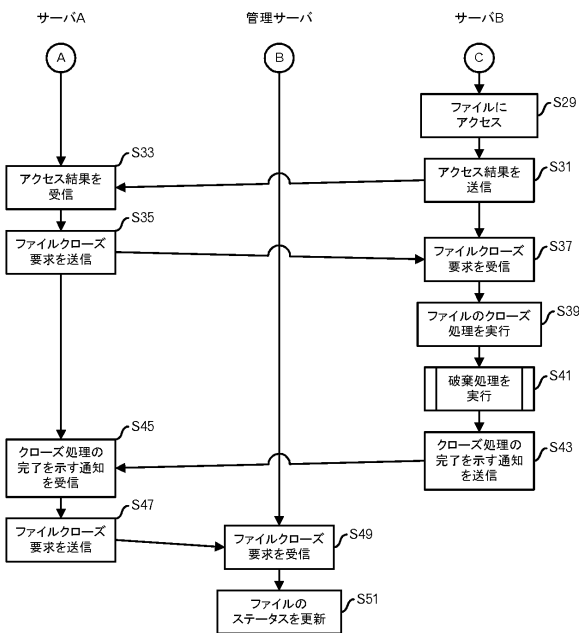
【図 9】



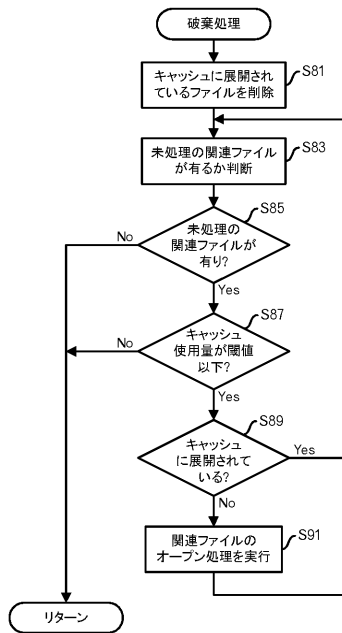
【図 10】



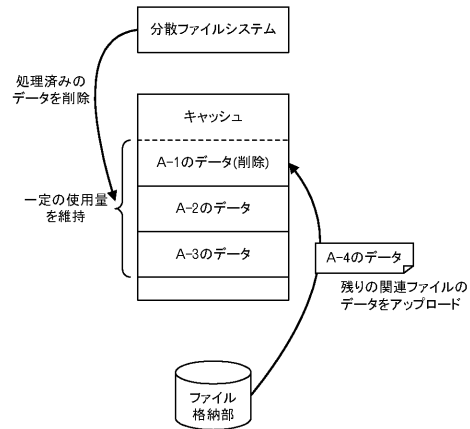
【図 11】



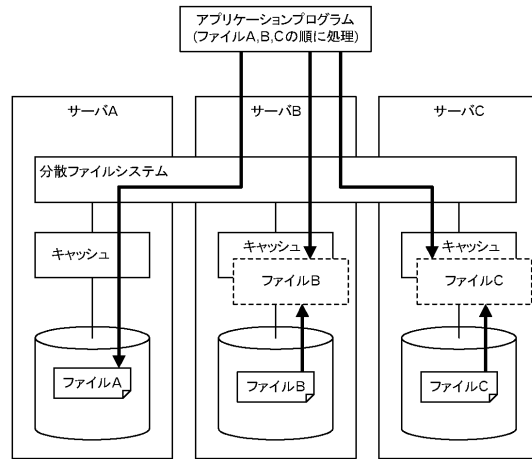
【図 12】



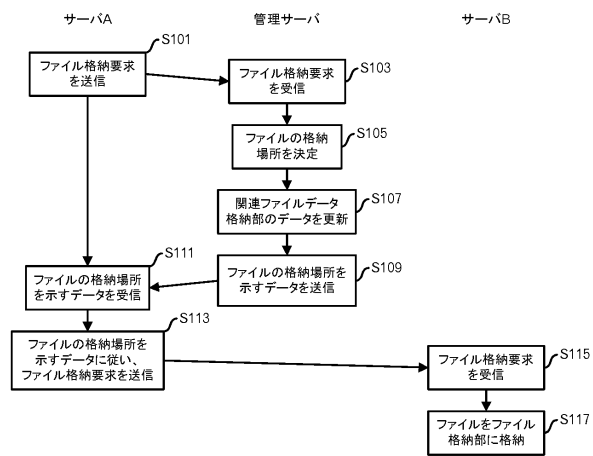
【図 13】



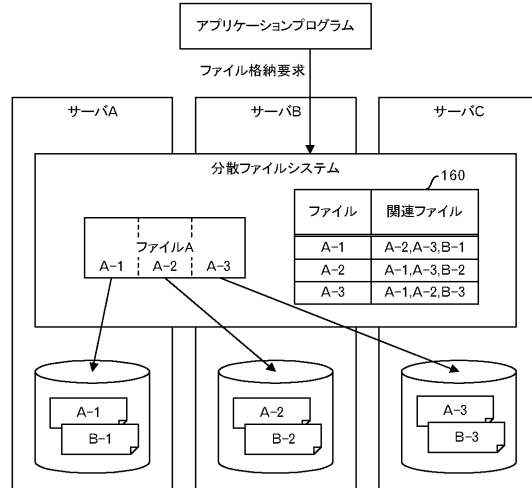
【図 14】



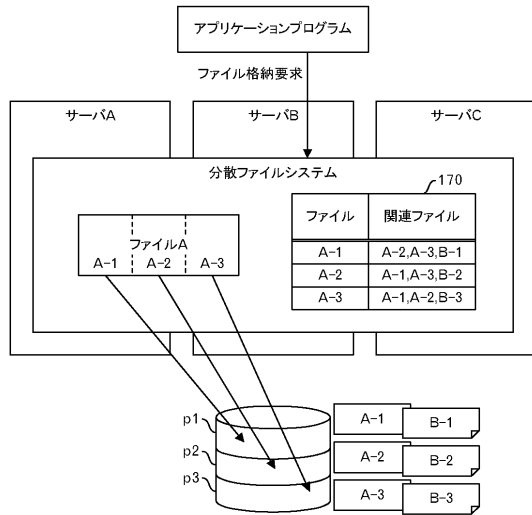
【図 15】



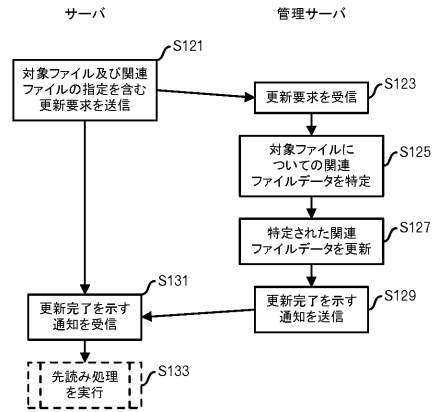
【図 16】



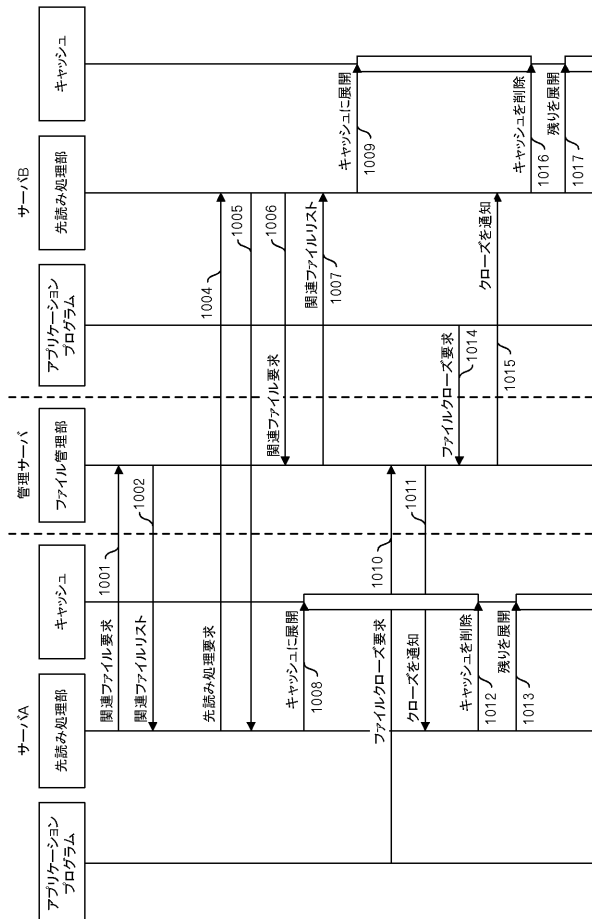
【図 17】



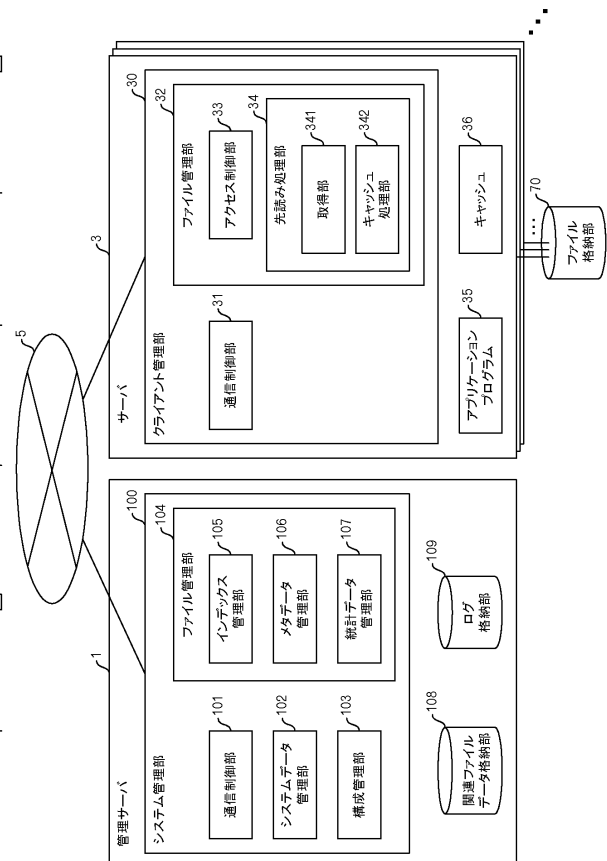
【図 18】



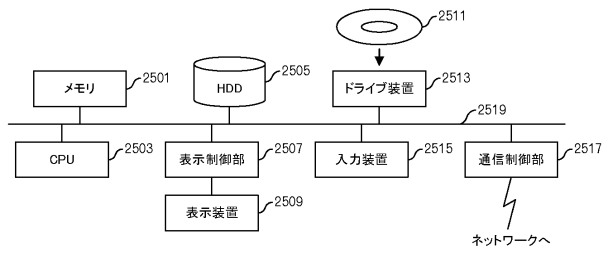
【図 19】



【図 20】



【図 21】





---

フロントページの続き

- (72)発明者 宮路 忠宏  
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
- (72)発明者 古田 智徳  
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
- (72)発明者 伊藤 史昭  
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 桜井 茂行

- (56)参考文献 特開2011-197977(JP,A)  
特開2004-118482(JP,A)  
特開2005-285058(JP,A)  
特開2008-146380(JP,A)

- (58)調査した分野(Int.Cl., DB名)
- |         |           |
|---------|-----------|
| G 0 6 F | 1 2 / 0 0 |
| G 0 6 F | 1 2 / 0 8 |
| G 0 6 F | 3 / 0 6   |
| G 0 6 F | 1 3 / 1 0 |