

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-361994

(P2004-361994A)

(43) 公開日 平成16年12月24日(2004. 12. 24)

(51) Int.Cl.⁷

G06F 9/46

G06F 11/34

F I

G06F 9/46

350

G06F 9/46

330C

G06F 11/34

A

テーマコード (参考)

5B042

5B098

審査請求 有 請求項の数 22 O L (全 24 頁)

(21) 出願番号 特願2003-155928 (P2003-155928)

(22) 出願日 平成15年5月30日 (2003. 5. 30)

(71) 出願人 000003078

株式会社東芝

東京都港区芝浦一丁目1番1号

(74) 代理人 100058479

弁理士 鈴江 武彦

(74) 代理人 100091351

弁理士 河野 哲

(74) 代理人 100088683

弁理士 中村 誠

(74) 代理人 100108855

弁理士 蔵田 昌俊

(74) 代理人 100084618

弁理士 村松 貞男

(74) 代理人 100092196

弁理士 橋本 良郎

最終頁に続く

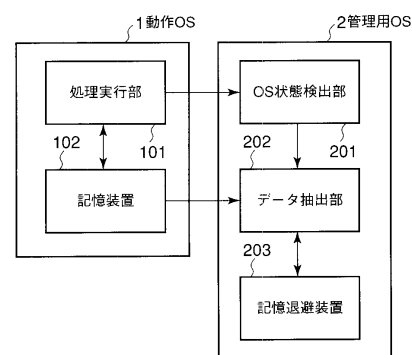
(54) 【発明の名称】 データ管理装置、データ管理方法及びプログラム

(57) 【要約】

【課題】 計算機の障害復旧に必要なデータを従来よりも確実に保存することのできるデータ管理装置を提供すること。

【解決手段】 管理対象となる動作OS 1よりも安全性を高めた管理用OS 2を用意し、この管理用OS 2は、OS状態検出部201が動作OS 1の状態を検出するとともに、データ抽出部202が動作OS 1の記憶装置102のデータを読み出すという形で両OSが連携して動作し、障害復旧に必要な全ての情報が逐次管理OS 2の持つ記憶退避装置203に残ることが保証される。これによって、データ保存のスケジューリングの難しさを取り除き、コンピュータウイルス被害等によってデータ管理システムそのものが障害を受ける危険性があるという問題点が解決される。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

対象オペレーティングシステムの持つ記憶装置のデータを管理するデータ管理システムにおいて、

前記対象オペレーティングシステムとは独立した管理用オペレーティングシステムを設け、

前記管理用オペレーティングシステムが、

前記対象オペレーティングシステムの動作状態が予め定められた複数の動作状態のいずれかに該当する場合に、該対象オペレーティングシステムの動作状態を検出する動作状態検出手段と、

前記動作検出手段により検出された前記動作状態に応じて前記記憶装置から退避させるべきデータを抽出する抽出手段と、

抽出された前記データを退避させるための退避用記憶手段とを備えたことを特徴とするデータ管理システム。

【請求項 2】

前記管理用オペレーティングシステムは、前記対象オペレーティングシステムに比較して、よりセキュリティ・ホールが少ないものであることを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 3】

前記管理用オペレーティングシステムは、前記対象オペレーティングシステムに比較して、より機能が制限されたものであることを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 4】

前記対象オペレーティングシステムは外部のネットワークへ接続する機能を包含するものであるのに対し、前記管理用オペレーティングシステムは外部のネットワークへ接続する機能を包含しないものであることを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 5】

外部のネットワークから前記対象オペレーティングシステムへのアクセスに比較して外部のネットワークから前記管理用オペレーティングシステムへのアクセスをより厳しく制限することを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 6】

前記管理用オペレーティングシステムのソフトウェアは、前記対象オペレーティングシステムのソフトウェアが動作する計算機と同一の計算機上で動作することを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 7】

前記管理用オペレーティングシステムは、前記対象オペレーティングシステムのソフトウェアを実行可能な仮想計算機のソフトウェアを含み、前記対象オペレーティングシステムのソフトウェアは、該仮想計算機上で動作することを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 8】

前記管理用オペレーティングシステムのソフトウェアは、前記対象オペレーティングシステムのソフトウェアが動作する計算機とは異なる計算機上で動作することを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 9】

前記データ管理システムは、前記管理用オペレーティングシステムとは独立して設けられた 1 又は複数の遠隔管理装置を更に備え、

前記遠隔管理装置は、前記抽出手段により抽出された前記データの全部又は一部を退避させるための遠隔退避用記憶手段を含むことを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 10】

前記管理用オペレーティングシステムは、前記記憶手段の内容を過去の或る時点の状態に復旧させるべく前記退避用記憶手段に退避されたデータを前記記憶手段へ書き戻す手段を更に備えたことを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 1 1】

前記管理用オペレーティングシステムは、前記記憶手段の内容を過去の或る時点の状態に復旧させるべく前記退避用記憶手段又は前記遠隔退避用記憶手段に退避されたデータを前記記憶手段へ書き戻す手段を更に備えたことを特徴とする請求項 9 に記載のデータ管理システム。

【請求項 1 2】

前記遠隔管理装置は、前記記憶手段の内容を過去の或る時点の状態に復旧させるべく前記遠隔退避用記憶手段に退避されたデータを前記記憶手段へ書き戻す手段を更に備えたことを特徴とする請求項 9 に記載のデータ管理システム。 10

【請求項 1 3】

前記対象オペレーティングシステムは、前記退避用記憶手段が接続された場合に、該接続された退避用記憶手段に退避されたデータを、過去の或る時点に前記記憶装置に記憶されていたデータとして読み出す手段を含むことを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 1 4】

前記対象オペレーティングシステムは、前記退避用記憶手段又は前記遠隔退避用記憶手段が接続された場合に、該接続された退避用記憶手段又は遠隔退避用記憶手段に退避されたデータを、過去の或る時点に前記記憶装置に記憶されていたデータとして読み出す手段を含むことを特徴とする請求項 1 に記載のデータ管理システム。 20

【請求項 1 5】

前記抽出手段は、抽出した前記データを退避するのに先立って、既に退避されている退避データに基づいて該抽出したデータをよりデータ量の少ない形態に変換することを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 1 6】

前記変換は、抽出した前記データに対応する過去の時点のデータが既に退避されている場合に、該過去の時点のデータを基準にして該抽出したデータを圧縮するものであることを特徴とする請求項 1 5 に記載のデータ管理システム。 30

【請求項 1 7】

前記予め定められた動作状態は、前記対象オペレーティングシステムがその実行を終了することを示す第 1 の動作状態を含むことを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 1 8】

前記予め定められた動作状態は、前記対象オペレーティングシステムがその実行を開始することを示す第 2 の動作状態を含むことを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 1 9】

前記予め定められた動作状態は、前記対象オペレーティングシステムがアプリケーションプログラムのインストールを実行することを示す第 3 の動作状態を含むことを特徴とする請求項 1 に記載のデータ管理システム。 40

【請求項 2 0】

前記予め定められた動作状態は、前記対象オペレーティングシステムが前記記憶装置内のデータに変更を加えたことを示す第 4 の動作状態を含むことを特徴とする請求項 1 に記載のデータ管理システム。

【請求項 2 1】

対象オペレーティングシステムの持つ記憶装置のデータを、前記対象オペレーティングシステムとは独立した管理用オペレーティングシステムにより管理するデータ管理方法であって、 50

前記対象オペレーティングシステムの動作状態が予め定められた複数の動作状態のいずれかに該当する場合に、該対象オペレーティングシステムの動作状態を検出するステップと、
検出された前記動作状態に応じて前記記憶装置から退避させるべきデータを抽出するステップと、
抽出された前記データを退避用記憶装置に退避させるステップとを有することを特徴とするデータ管理方法。

【請求項 22】

対象オペレーティングシステムの持つ記憶装置のデータを、前記対象オペレーティングシステムとは独立した管理用オペレーティングシステムにより管理するデータ管理装置としてコンピュータを機能させるためのプログラムであって、
前記対象オペレーティングシステムの動作状態が予め定められた複数の動作状態のいずれかに該当する場合に、該対象オペレーティングシステムの動作状態を検出する機能と、
検出された前記動作状態に応じて前記記憶装置から退避させるべきデータを抽出する機能と、
抽出された前記データを退避用記憶装置に退避させる機能をコンピュータに実現させるためのプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、計算機に係る障害を復旧させるために計算機に係るデータの待避や復元を行うデータ管理装置、データ管理方法及びプログラムに関する。

【0002】

【従来の技術】

計算機の障害は、ハードウェアの故障またはソフトウェア（記憶装置のデータ）の誤った修正等によって起こるが、障害復旧にあたっては、障害発生以前におけるソフトウェアの状態を正確に復元させることができるか否かが最も重要なポイントである。従来より知られている計算機障害復旧装置においては、大まかに3つの手法がある。

【0003】

まず、第1の方法は、記憶装置を多重化し、動作OSが第1の記憶装置に書き込んだのと同じデータを第2の記憶装置にも書き込むことにより、万一、第1の記憶装置のデータが破壊されても第2の記憶装置に保存されたデータを利用し復元させる方法である。例えば、動作OSから第2の記憶装置は直接アクセスできない仕組みにしておき、障害が発生した時点で起動記憶装置（起動ディスク）を第1の記憶装置から第2の記憶装置に切替えることにより、瞬時に障害復旧することが可能となる（例えば、特許文献1参照）。この方式は、ハードディスク等の記憶装置が物理的に動作不能になった場合など、即座に直前の障害発生時に復旧できるという利点のあるものである。しかし、計算機の障害発生後に直ちに完全に動作不能となるタイプの障害発生はあまり多くなく、実際には、障害発生後も動作を続け、例えば数日後に障害報告を受けて障害復旧を試みる場合が多いのであるが、この方式によれば過去の障害発生時のデータは既に廃棄されているから、障害を復旧することはできない。すなわち、どの時点が障害発生のタイミングであるかを、リアルタイムには検出できないから、この方式では障害復旧に必要なデータは失われてしまう可能性が極めて高い。

【0004】

次に、第2の方法として、動作OS上に、データのバックアップのためのソフトウェアを常時動作させ、または必要と思われるタイミングに動作させることにより、動作OSが変更した記憶装置内のデータを逐次第2の記憶装置に保存していく方法がある。例えば、新しいアプリケーションのインストールによって障害が発生することを防ぐために、アプリケーションのインストール直前の状態（スナップショット）を保存しておくという方式がある。例えば、米国MicrosoftTM社製品のWindows MeTM（200

10

20

30

40

50

0年9月発売)に「システムの復元」という機能として実装されている(例えば、非特許文献2参照)。この方式を利用すれば、第1の方法とは異なり、複数の状態を保存しておくことが可能であるから、障害復旧に必要なデータが残っている可能性が高まる。しかし、この方式は、障害復旧に必要なデータもまた動作OSのデータとして管理されているため、障害復旧のためのソフトウェア自身が障害を受ける可能性があり、やはり障害復旧に必要なデータは失われている可能性が高い。例えば、今日多く見られるコンピュータウイルスと呼ばれるシステム破壊を目的としたソフトウェアによって受ける被害に対してはほとんど無力であることが問題となっている。

【0005】

次に、第3の方法として、動作OSとは別に、データのバックアップを目的とするOSを別途動作させることにより、バックアップ作業の際に一旦動作OSを停止(シャットダウン)させ、停止している状態のデータを保存しておく方法がある。例えば、米国 SymantecTM社の「Norton GhostTM」という製品などに採用されている(例えば、非特許文献3参照)。これは現在最も広く用いられている方法であり、動作OSの挙動に全く影響されることなく障害復旧に必要なデータが確実に保存できるという利点がある。しかし、この方式は、保存すべき「動いている状態」が今であるということを予め知っていなければならないため、例えばOSを破壊するかもしれない危険な操作をする前に保存(バックアップ)しておくという目的には利用できるものの、一般にいつ発生するか判らない障害に対し、その障害発生前の状態に戻すという一番重要な目的にはほとんど役に立たない。

【0006】

【非特許文献1】

Apparatus and method for providing a transparent disk drive back-up USP 6,175,904 (2001/1/16)

【0007】

【非特許文献2】

<http://www.microsoft.com/japan/enable/training/kblight/t006/3/17.htm>

【0008】

【非特許文献3】

<http://www.symantec.com/region/jp/index.html>

【0009】

【発明が解決しようとする課題】

上述したように従来、計算機の障害復旧に必要なデータが保存されているか否かは、さまざまな状況に依存しており、必ずしも復旧することができる保証はないという問題があった。

【0010】

本発明は、上記事情を考慮してなされたもので、計算機の障害復旧に必要なデータを従来よりも確実に保存することのできるデータ管理装置、データ管理方法及びプログラムを提供することを目的とする。

【0011】

【課題を解決するための手段】

本発明は、対象オペレーティングシステムの持つ記憶装置のデータを管理するデータ管理システムにおいて、前記対象オペレーティングシステムとは独立した管理用オペレーティングシステムを設け、前記管理用オペレーティングシステムが、前記対象オペレーティングシステムの動作状態が予め定められた複数の動作状態のいずれかに該当する場合に、該対象オペレーティングシステムの動作状態を検出する動作状態検出手段と、前記動作検出手段により検出された前記動作状態に応じて前記記憶装置から退避させるべきデータを抽

出する抽出手段と、抽出された前記データを退避させるための退避用記憶手段とを備えたことを特徴とする。

【0012】

好ましくは、前記管理用オペレーティングシステムは、前記対象オペレーティングシステムより高い安全性を有するものであるようにしてもよい。

【0013】

また、本発明は、対象オペレーティングシステムの持つ記憶装置のデータを、前記対象オペレーティングシステムとは独立した管理用オペレーティングシステムにより管理するデータ管理方法であって、前記対象オペレーティングシステムの動作状態が予め定められた複数の動作状態のいずれかに該当する場合に、該対象オペレーティングシステムの動作状態を検出するステップと、検出された前記動作状態に応じて前記記憶装置から退避させるべきデータを抽出するステップと、抽出された前記データを退避用記憶装置に退避させるステップとを有することを特徴とする。 10

【0014】

また、本発明は、対象オペレーティングシステムの持つ記憶装置のデータを、前記対象オペレーティングシステムとは独立した管理用オペレーティングシステムにより管理するデータ管理装置としてコンピュータを機能させるためのプログラムであって、前記対象オペレーティングシステムの動作状態が予め定められた複数の動作状態のいずれかに該当する場合に、該対象オペレーティングシステムの動作状態を検出する機能と、検出された前記動作状態に応じて前記記憶装置から退避させるべきデータを抽出する機能と、抽出された前記データを退避用記憶装置に退避させる機能とをコンピュータに実現させるためのプログラムである。 20

【0015】

なお、装置に係る本発明は方法に係る発明としても成立し、方法に係る本発明は装置に係る発明としても成立する。

また、装置または方法に係る本発明は、コンピュータに当該発明に相当する手順を実行させるための（あるいはコンピュータを当該発明に相当する手段として機能させるための、あるいはコンピュータに当該発明に相当する機能を実現させるための）プログラムとしても成立し、該プログラムを記録したコンピュータ読取り可能な記録媒体としても成立する。 30

【0016】

本発明では、管理用オペレーティングシステムは対象オペレーティングシステムの動作状態を知り、例えば動作OSが停止したことや新規アプリケーションをインストールしようとしていることなどの情報を得て記憶装置からデータを取り出すことができるため、ユーザが対象オペレーティングシステムの動作状態を意識することなく、障害復旧に必要なデータをより確実に退避させておくことが可能となり、障害発生以前の或る時点に復旧させるためのデータを持つことが可能になる。

【0017】

【発明の実施の形態】

以下、図面を参照しながら発明の実施の形態を説明する。 40

【0018】

（第1の実施形態）

図1に、本発明の第1の実施形態に係るデータ管理システムを含む計算機システムの構成例を示す。

【0019】

図中、1は動作OS（対象オペレーティングシステム）、2は管理用OS（管理用オペレーティングシステム）（データ管理システム）である。動作OS1は、障害を復旧させたい対象のOSである。管理用OS2は、専ら動作OS1の障害復旧作業のために用意され、動作OS1の障害の障害復旧を目的として、動作OS1に関係するデータ（若しくはファイル）の退避などを行うためのOSである。詳しくは後述するように、管理用OSは、 50

動作OS 1に比較してより高い安全性を持たせるようにするのが好ましい。

【0020】

なお、一般にOSとは計算機を管理するソフトウェアの部分のみ指すこともあるが、本実施形態においては、記憶装置等の周辺機器をも含んだシステムを指すものとする。ただし、計算機システムは複数のOSがインストールされ稼働する場合があるので、ここでは、その一主体である動作OSが利用する部分のみや管理用OSが利用する部分のみを指すものとなる。

【0021】

システム構成としては、大きく分けて、図2に示すように動作OS 1と管理用OS 2とが別々の計算機A、B上で実現される構成と、図3に示すように動作OS 1と管理用OS 2とが同一の計算機C上で実現される構成とが可能である。前者の場合、動作OS 1と管理用OS 2とを、LAN等で接続して同一の計算機室や同一のビル内などの近接した場所に設置する形態や、広域ネットワーク等で接続して物理的に離れた場所に設置する形態など、種々の設置形態が可能である。

【0022】

動作OS 1は、プログラムの実行等を行うための処理実行部101と、処理実行部101がデータの書き込みや読み出し等を行う記憶装置102とを有する。

【0023】

記憶装置102は、典型的には、ハードディスクやフラッシュメモリ装置などの物理的な記憶媒体であるが、処理実行部101がデータの書き込みや読み出しを行うものであれば、記憶媒体を持たないものであってもよい。例えば、動作OS 1は通信回線を持ち、その通信回線を経由して外部の記憶装置にデータを書き込みおよび外部の記憶装置からデータを読み込むような場合も可能である。

【0024】

管理用OS 2は、OS状態検出部201とデータ抽出部202と記憶退避装置203とを有する。

【0025】

図4に、管理用OS 2が記憶装置102のデータを退避する手順の一例を示す。

【0026】

OS状態検出部201は、対象となる動作OS 1の動作状態（動作状況）を検出する（ステップS11）。すなわち、対象となる動作OS 1の処理実行部101の動作状態に関する情報を取得する。

【0027】

データ抽出部202は、検出された動作状態に応じて記憶装置102から退避すべきデータを抽出する（読み出す）（ステップS12）。

【0028】

データ抽出部202は、抽出したデータを、記憶退避装置203へ退避させる（書き込む）（ステップS13）。

【0029】

以下、OS状態検出部201による動作OS 1の動作状態の検出について詳しく説明する。

【0030】

動作の状態に関する情報としては、例えば、「処理実行部101が動作OS 1の終了処理を行い、動作を停止する（すなわち、シャットダウンする）」という情報がある。

【0031】

例えば、動作OS 1が動作を停止する直前に停止の予告メッセージをネットワーク上に流す場合に、管理用OS 2は、その予告メッセージを受信し、OS状態検出部201は、動作OS 1が間もなく停止することを知る。

【0032】

なお、この通知をより確実に行うために、例えば、予告メッセージを受信した管理用OS

2が動作OS 1のネットワークインターフェースに対するICMPメッセージ（INTERNET CONTROL MESSAGE PROTOCOL）を流すなどして、動作OS 1が実際に停止したことを確認するようにしてもよい。

【0033】

ここでは、予告メッセージがネットワーク上に流される場合を例にとって説明したが、管理用OS 2に動作OS 1の停止を伝える手段あるいはOS状態検出部201が動作OS 1の停止を検出する手段には特に限定はなく、どのような方法によってもよい。

【0034】

動作の状態に関する情報の他の例としては、「処理実行部101が動作OS 1の処理を開始する（すなわち、OSをブートする）」という情報がある。

10

【0035】

例えば、動作OS 1が動作を開始した直後にOSの再起動メッセージをネットワーク上に流す場合に、管理用OS 2は、その再起動メッセージを受信し、OS状態検出部201は、動作OS 1が処理を開始したことを知る。

【0036】

この再起動メッセージには、動作OS 1の起動オプションを含むとより効果的である。起動オプションは、例えば、動作OS 1がどのようなサービス（デーモン）を実行しようとしているか、OSのバージョンは何か、何の目的で動作OS 1が起動されたかなどを示す情報である。例えば、この動作OS 1がCADシステムの利用を目的として起動されたことが予め判っていれば、その情報を後述するデータ抽出部202が利用し、CADファイルの更新を優先して処理するといった効率化が可能になる。また、起動オプションに相当する情報は、再起動メッセージに含む方法をとるのではなく、単独のメッセージとして随時送信する方法をとっても構わない。

20

【0037】

ここでは、再起動メッセージがネットワーク上に流される場合を例にとって説明したが、管理用OSに動作OS 1の再起動を伝える手段あるいはOS状態検出部201が動作OS 1の再起動を検出する手段には特に限定はなく、どのような方法によってもよい。

【0038】

また、動作の状態に関する情報のさらに他の例としては、動作OS 1そのものの内部動作に関する情報が考えられる。

30

【0039】

例えば、「処理実行部101が記憶装置102に対してデータの書き込みを行った」という情報である。この場合、全てのデータ書き込み情報を逐一利用してもよいし、処理を効率化するために特定のデータ書き込み情報のみを利用してもよい。

【0040】

後者の特定のデータ書き込みとは、例えば、OSのシステム領域の書き換えなどの重大な書き込みであったり、あるいは、アプリケーションのインストール時に作成されるファイルへの書き込みであったりする。何を特定のデータ書き込みとして扱うべきかについては、動作OS 1の性質やまたその利用目的などに応じて決定するのが好ましい。これは、例えば、動作OS 1の性質や利用目的などから特定のデータ書き込みを求めるためのルールを定義ファイルに記述するなどして実現することができる。

40

【0041】

また、動作の状態に関する情報に含まれるデータの中身に関しても、書き込みしたファイル名であることも考えられるし、また、その記憶装置上の場所（トラック番号やファイルのノード番号等）であることも考えられるし、また、場合によってはファイルの更新内容（3行目のデータをこのようなデータに書き換えた等）であることも考えられる。こういったルールも同様に定義ファイルに記述するなどして実現するようにしてもよい。

【0042】

動作の状態に関する情報のさらに他の例としては、動作OS 1が他の計算機と通信を行ったというものが考えられる。

50

【0043】

これも、例えば、前述したデータ書き込みの場合と同様、全ての通信記録の中からルールに従って必要な情報を採用すればよい。例えば、OSのアップデートをするための通信（すなわち、アップデートサイトとの通信）があれば、重要な変更がなされると予測することができるし、危険なWWWサイト（例えば、多くの複雑なスクリプトが書かれたサイトや、危険だと予め判っているサイトなど）へのWWWアクセスがあれば、コンピュータウイルスが入った可能性があるかと推定することができる。

【0044】

なお、上記では、予め定められた動作状態が検出された場合に、該動作状態に応じて記憶装置102から退避すべきデータを抽出し、これを記憶退避装置203へ退避させるものであったが、これに加えて、予め定められた時間が経過するごとにも、記憶装置102から退避すべきデータを抽出し、これを記憶退避装置203へ退避させるようにしてもよい。なお、予め定められた動作状態が検出されたことと、予め定められた時間が経過したことが同時に発生した場合には、例えば、前者を優先するものとしておけばよい。

【0045】

次に、データ抽出部202による退避すべきデータの記憶装置102からの読み出し及び記憶退避装置203へのデータ退避について詳しく説明する。

【0046】

データを記憶装置102から読み出す方法は主として2通りの方法が可能である。1つは、実際に記憶装置102に記憶されているデータを通常のデータ読み出しと同様の方法で読み出す方法であり、もう1つは、処理実行部101から記憶装置102への書き込み命令が出た際にその信号を直接読み取ることによってデータを読み出すのと同様の効果を得る方法である。ただし、前記したような物理的な記憶装置を内包しない記憶装置の場合には後者の方法を用いる。

【0047】

データ抽出部202は、抽出したデータを、記憶退避装置203へ書き込む。この書き込みは、通常のファイルシステム等の書き込みとは異なり、時系列を考慮した書き込みとなる（例えば、あるデータを記憶装置102から読み出した時刻もしくは記憶退避装置203へ書き込んだ時刻を、該あるデータに対応付けて保存しておく）。すなわち、当該データの更新履歴を管理するのと同様の処理を行う。

【0048】

例えば、図5に示すように、3日前にデータ抽出部202が取り出し記憶退避装置203に書き込んだ“foo”という名前のファイルのデータ（このときの内容をaで表す）が存在する場合、その後（例えば、1日前）に再度データ抽出部202が“foo”という名前のファイルのデータ（このときの内容をbで表す）を取り出して記憶退避装置203の同一名のファイルのデータに上書きすると、3日前の状態に復旧させるためのデータ（すなわち、内容aのデータ）が欠落してしまうから、例えば障害発生が2日前であったとすると、その障害を復旧させることができない可能性が出てしまう。そこで、各時点（例えば、各退避時点）における同一名のファイルのデータはそれぞれ区別して、それぞれが保存されるような方法によって、この書き込みを行う。なお、この区別は、例えば、記憶退避装置203に退避した時刻や、バージョン番号等によって行えばよい。図6は、“foo”という名前のファイルのバージョンで内容を区別して、各バージョンのデータを保存するようにした例である。

【0049】

また、例えば、ファイル“foo”が一旦削除された後に、“foo”という名前のファイルが再度作成される場合があるので（削除されてから再度作成されるまでの間、ファイル“foo”は存在しないことになる）、“foo”が削除された場合には、その旨を示す情報を記憶退避装置203に保存しておくのが望ましい。図7は、t3の時点でファイル“foo”が削除されたことを示す情報を退避した例を示す（なお、foo(t4, ver1)は、foo(t1, ver1)とは、バージョン番号は同じであるが、時刻情

報が異なるので、内容は異なる）。

【0050】

また、ファイル“foo”が新たに作成されたものか、修正されたものか、削除されたもののかの区別を示す情報を、ファイル“foo”に対応付けて保存しておくようにしてもよい。

【0051】

また、例えば、パス名の異なる“foo”という名前のファイルが複数存在し得る場合には、それらは異なるファイルとして扱うものとする。

【0052】

データ抽出部202が記憶装置102より抽出したデータを記憶退避装置203に書き込むにあたっては、OS状態検出部201によって検出された動作OS1の動作状態を示す情報を利用すると好ましい。これは動作OS1の状態を保存するには動作OS1の現況（現在の動作状態）を意識しておくのが望ましいからである。 10

【0053】

例えば、動作OS1が完全に停止した状態であれば、データ抽出部202が記憶装置102よりデータを取り出している間に記憶装置102のデータが変更されることがないことが保証されることになる。このような場合には、データの取り出しのスケジューリングを意識する必要がなくなり、とにかく考えられる全てのデータを保存しておけばよい。そのデータに基づけば、次に動作OS1が起動（電源オン）するであろう時点での状態を将来いつでも取り出すことができる。 20

【0054】

一方、動作OS1が動作している場合にはデータ抽出部202が記憶装置102よりデータを取り出している間にも、記憶装置102のデータが変更されることがある。このような場合には、データ抽出部202がやみくもに記憶装置102のデータを読み出すのでは効率が良くない場合がある。例えば、ユーザが特定の文書ファイルを編集していることが把握できるときには、関連ファイルが秒単位で変更されることがあり得るのであり、そういったファイルに注目して重点的にデータを取り出すと効果的である。

【0055】

また、他のケースとしては、動作OS1が新たなアプリケーションを登録しようとしている際には、システム関連のファイル変更が重要なポイントとなり、それらのファイル（共通のライブラリファイルや、システム設定ファイルなど）に注目して重点的にデータを取り出すと効果的である。 30

【0056】

これらは、例えば、動作OS1の性質を考慮した上でデータ抽出部202のアルゴリズムを設計すると、より効果を発揮する場合もある。例えば、現在知られているいくつかのOSでは、ユーザが最近編集したファイルを簡単に取り出せる仕組みが用意されており、最近編集されたファイルの一覧あるいはそれらファイルへのポイントが一箇所にまとめて管理されている。動作OS1が、そのような性質のOSである場合には、当該動作OS1に対しては、その情報を利用することにより、ユーザが現在どのようなファイルを重点的に操作（変更）しているかを知り、データの取り出しの際の情報とすることができる。 40

【0057】

また、例えば、現在知られているいくつかのOSでは、新規のアプリケーションをインストール（登録）したりアンインストール（削除）したりするためのインタフェースが統一されており、アプリケーションのインストールやアンインストールを行う際には、必ず決められたソフトウェアが起動することになっている。動作OS1が、そのような性質のOSである場合には、当該動作OS1に対しては、その情報を利用することにより、現在、障害復旧を行うにあたっての重要な変更が行われようとしていることを、データ抽出部202が知ることにより、データの取り出しの際の情報とすることができる。

【0058】

なお、データ抽出部202が記憶装置102より抽出したデータを記憶退避装置203に 50

書き込むにあたっては、OS状態検出部201によって検出された動作OS1の動作状態を示す情報をも当該データに対応つけて記憶するようにしてもよい。

【0059】

以上のように、管理用OS2が動作OS1の動作状態を知り、例えば動作OS1が停止したことや、新規アプリケーションをインストールしようとしていることなどの情報を得て記憶装置102からデータを取り出すことができるようにすることによって、ユーザが動作OS1の状態を意識することなく、障害復旧に必要なデータを保存しておくことが可能となり、バックアップのし忘れの心配がなくなるため、種々多様な状況で障害が発生した場合においても、障害発生以前の任意の時点に復旧させるためのデータを記憶退避装置203に持つことが保証される（あるいは、それが期待される）。

10

【0060】

例えば、実際の障害発生から3日後に障害発生を認識した場合に、1日前や2日前の保存データを用いたときには障害発生前の状態が得られないので、障害を復旧させることはできないが、3日前の保存データを用いたときには障害が復旧されることを知ることができ、記憶装置102の状態を3日前の状態に戻すことによって、障害発生直前の状態に計算機を復旧させることが可能となる。

【0061】

なお、記憶退避装置203に退避したデータをユーザが直接利用できないようにしてもよい。このようにすれば、たとえコンピュータウイルスを含むファイルが記憶退避装置203に退避されていても、コンピュータウイルスが発動しない蓋然性が非常に高くなり、管理用OS2側がコンピュータウイルスにより被害を受けることを未然に防止することができる。

20

【0062】

ところで、仮に管理用OS2の安全性が動作OS1の安全性と同程度又はそれ以下であると、コンピュータウイルスのような悪意のソフトウェアによって動作OS1が被害を受けた場合、同じ原因により、管理用OS2も同時に被害を受け、結局、障害復旧することはできない危険性があるので、好ましくは、管理用OS2を動作OS1よりも安全性を高めたものにしておくのが望ましい。このようにすれば、管理用OS2が動作OS1と同時に障害が発生する確率を概ねゼロに（あるいは、非常に低く）することができる。

【0063】

一般に動作OS1はアプリケーション実行に必要なさまざまな機能を実現しなくてはならないので安全性をあまり高めることができないが、管理用OS2は機能を限定することができるため動作OS1に比べ安全性を高めることが比較的容易に可能である。

30

【0064】

例えば、動作OS1がWWWサーバである場合、WWWサーバのセキュリティーホールにより障害を受ける危険性があるが、管理用OS2にはWWWサーバをインストールしておく必要がないため、かかる原因により管理用OS2に障害が発生することはあり得ない。したがって、動作OS1よりも安全性を高めた管理用OS2に動作OS1の障害復旧機能を分離するという構成を採ることによって、従来よりも確実性の高い計算機システムを実現することが可能となる。

40

【0065】

上記のように、管理用OS2の安全性を動作OS1に比較してより高めておくことによって、仮に動作OS1がコンピュータウイルス等によって破壊されたとしても管理用OS2は破壊されないことが期待される。ここで、安全性を高めるとは一般にいくつかの方法がある。

【0066】

まず、第1に、（OSそれ自体に）セキュリティー・ホールが少ない（又は無い）ことが知られているOSを管理用OS2に採用する方法がある。この場合、管理用OS2は、必ずしも絶対的にセキュリティー・ホールが少ないもの（例えば、現に存在する最もセキュリティー・ホールが少ないOS）でなくても、動作OS1に比較して、よりセキュリティー・ホー

50

ルの少ないOSを用いて構わない（よりセキュリティ・ホールの少ないOSが、セキュリティ・ホールの無いOSであれば、理想的である）。動作OS 1においては、目的とするアプリケーションを動作させる必要があるため、必ずしも安全なOSを選択することはできないが、管理用OS 2においては、安全なOSを選択することが可能である。

【0067】

第2に、（動作OS 1と管理用OS 2と同じOSが用いられているか、あるいは動作OS 1と管理用OS 2とで異なるOSが用いられているが各OSの持つ安全性が同程度であるような場合であっても）、管理用OS 2の機能を制限することによって安全性を高める方法がある。管理用OS 2の機能は、動作OS 1に比較して、より制限されていればよく、管理用の機能のみ持つようにしてもよいし、管理用の機能以外の機能をも持っていたとしても構わない。動作OS 1においては、目的とするアプリケーションを稼働させるために、必要となる多くのサービスをインストールし、動作させなければならず、また必要なプログラム（コマンド）も多くインストールしておかなければならないが、管理用OS 2においては、動作OS 1を管理するだけの目的に使えばよいので、不要となるサービス（ほとんどのサービス）を停止させておけば安全性が高まる。また、プログラム（コマンド）も必要なものは限られるため、不要なプログラム（コマンド）を削除しておけば、セキュリティ・ホールのあるコマンドの動作によってシステムが障害を受けるといった危険性を下げることが可能である。

10

【0068】

第3に、管理用OS 2は動作OS 1とは異なる動作環境としておく方法がある。例えば、動作OS 1がWWWサーバであるとすれば、動作OS 1は必ずインターネットに接続させる必要があるが、管理用OS 2は動作OS 1を管理するのが目的であるからその必要はない。また、ファイアウォールを運用するなどにより、ネットワーク外からの動作OS 1へのアクセスに比較して、管理用OS 2へのアクセスをより厳しく制限しておくことも可能である。また、音声の入出力ドライバなど、本システムに必要な機能は管理用OS 2では動作しないように設定しておくことも可能である。

20

【0069】

以上例示した3つの手段は、少なくとも一つ採用すれば、安全性を高めたと言えることができるが、好ましくは複数を併用するとより安全性を高めることが期待される。

【0070】

なお、上記の他にも、例えば、特に重要なデータを管理する場合などで、管理用OS 2を多重化するという手法により、システムの信頼性若しくは安全性を高める方法も可能である。その際、多重化した管理用OS 2ごとに使用するOSを異ならせるようにしてもよい（多重化した管理用OS 2の全てが同時にコンピュータウィルスの被害を受けたために記憶装置102のデータの退避が不能になることを、概ねゼロもしくは非常に低い確率にすることができる）。

30

【0071】

以上説明したように、本実施形態では、動作OS 1とは別に管理用OS 2を設け、この管理用OS 2が動作OS 1の状態に応じてデータの退避等を行うことにより、動作OS 1の障害復旧に必要なデータをユーザが意識することなく自動的に管理用OS 2の記憶退避装置204に保存し続けることが可能になる。また、管理用OSを動作OSに比べてより安全性を高めるとより効果的である。ユーザは、待避された（保存された）データを用いて動作OS 1の障害復旧を行うことができる。あるいは、待避された障害発生前のデータをもとにして所望のアプリケーションの実行を再開することができる。

40

【0072】

（第2の実施形態）

次に、図8に、本発明の第2の実施形態に係るデータ管理システムを含む計算機システムの構成例を示す。

【0073】

この構成例は、管理用OS 2（データ管理システム）が、図1の構成例に加えて、データ

50

復旧部 204 を備えている。データの退避の処理については基本的には第 1 の実施形態と同様である。以下では、第 1 の実施形態と相違する部分を中心に説明する。

【0074】

図 9 に、管理用 OS 2 が記憶退避装置 203 から記憶装置 102 へ退避データを書き戻す手順の一例を示す。

【0075】

管理用 OS 2 は、ユーザから復旧時点を指定する情報を含む復旧の指示を受ける（ステップ S 21）。

【0076】

データ復旧部 204 は、指定された復旧時点を指定する情報に基づいて、記憶退避装置 203 から、記憶装置 102 へ書き戻すべきデータを抽出する（ステップ S 22）。 10

【0077】

記憶退避装置 203 は、抽出したデータを、記憶装置 102 へ書き戻す（ステップ S 23）。

【0078】

復旧時点の指定には種々の方法が考えられる。

【0079】

例えば、ユーザは、所望する日時（あるいは、現在から溯るべき時間等）を指定し、データ復旧部 204 は、記憶装置 102 の内容を復旧させることができる復旧可能時点（例えば、過去に実際にデータ退避が行われた時点）のうちから、指定された日時に最も近いもの（あるいは、指定された日時以前で最も近い当該指定された日時に最も近いもの）を選択し、記憶装置 102 の状態がその時点の状態になるように、記憶退避装置 203 から、記憶装置 102 へ書き戻すべきデータを抽出するようにしてもよい。 20

【0080】

例えば、“foo”という名前のファイルの退避・復旧を例にとって説明する。図 10 のように、“foo”という名前のファイルに関する情報が、時刻 t 3, t 5, t 9, t 14 でそれぞれ退避されたとする。ただし、時刻 t 3 以前には当該ファイルは存在しないものとする。また、時刻 t 9 では当該ファイルが削除された旨の情報が保存されたものとする。例えば、ユーザから復旧時刻として時刻 t 8 が指定された場合に、“foo”という名前のファイルについては、時刻 t 8 以前で最も近い時刻 t 5 が選択され、そのときに退避されたバージョン番号 2 のデータによる復旧が行われる（例えば、バージョン番号 2 のデータが記憶装置 102 へ書き戻される）。また、例えば、時刻 t 4 が指定された場合には、時刻 t 4 以前で最も近い時刻 t 3 が選択され、そのときに退避されたバージョン番号 1 のデータによる復旧が行われる。また、例えば、時刻 t 12 や t 2 が指定された場合には、“foo”という名前のファイルは存在しないので、そのための復旧がなされる（例えば、記憶装置 102 から“foo”という名前のファイルが削除される）。また、例えば、時刻 t 15 が指定された場合には、時刻 t 15 以前で最も近い時刻 t 14 が選択され、そのときに退避されたバージョン番号 1 のデータによる復旧が行われる。なお、foo(t 14, ver 1) は、foo(t 3, ver 1) とは、バージョン番号は同じであるが、時刻情報が異なるので、内容は異なる（foo(t 14, ver 1) は、“foo”という名前のファイルが一旦削除された後に再度作成されたものである）。 30 40

【0081】

また、例えば、ユーザは、所望する日時等と、データを退避するにあたって検出された動作 OS 1 の動作状態（例えば、シャットダウン、ブート、あるいはインストール等）とを指定し、データ復旧部 204 は、データを退避するにあたって検出された動作状態が指定された動作状態と一致する復旧可能時点のうちから、指定された日時に最も近いもの（あるいは、指定された日時以前で最も近い当該指定された日時に最も近いもの）を選択するようにしてもよい。

【0082】

また、例えば、管理用 OS 2 は、記憶退避装置 203 に記憶された情報をもとに、復旧可 50

能時点を示す日時等（あるいは、日時等と動作OS1の動作状態との組合せ）を一覧表として提示し、ユーザは、提示された日時等（あるいは、日時等ととの組合せ）のうちから所望のものを選択するようにしてもよい。

【0083】

また、ファイル名を指定して、当該ファイル名のファイルについてのみ、復旧処理できるようにしてもよい。

【0084】

その他、種々のバリエーションが可能である。

【0085】

以下、障害復旧を行う場合について詳しく説明する。

10

【0086】

まず、動作OS1の障害の発生は、例えば、ハードウェアの故障の他、あるソフトウェアの動作中にそのソフトウェアの全部又は一部の機能が利用不能に陥ったことや、計算機が予期しない動作をしたことなどがユーザによって認識されるに至ることによって、発見される。

【0087】

ユーザが動作OS1の障害発生を認識した場合、一旦、処理実行部101の動作を停止させ、ハードウェアの故障等があればその修理を行った後に、記憶装置102の内容を、障害発生前の状態に書き戻すことにより、障害を復旧することができる。

【0088】

20

前述したように、障害復旧に必要なデータは記憶退避装置203に格納されていることが保証（あるいは期待）されているが、記憶退避装置203に格納されているデータは過去に記録した全てのデータ（あるいは時系列上で複数の時点での状態に係るデータ）であり、障害復旧に必要なでないデータも含まれ得る。まず、データ復旧部204は、記憶退避装置203に格納されているデータから、記憶装置102に書き戻すべきデータを抽出する。

【0089】

書き戻すべきデータが何であるかはケースバイケースであり、それは例えばシステム管理者等が判断する。例えば、3日前の午後3時に動作OS1がコンピュータウィルスの被害を受けたと判定された場合、その直前の状態に戻すことを試みる。それは、3日前の午後2時59分頃の状態であるかもしれないし、またはそれ以前に動作OS1をシャットダウンした時刻、例えば3日前の午前5時であるかもしれない。あるいは、3日前の午前5時の状態に戻した上で、午後2時59分までに特定部分（ディレクトリ）に追加されたファイルを加えたものであるかもしれない。これは運用している動作OS1の性質によって変わるものであり、システム管理者等が判断する。この判断情報を、データ復旧部204に入力することによって、データ復旧部204は障害復旧処理を開始する。

30

【0090】

なお、上記ではシステム管理者等が判断情報を入力するとしたが、システム構成によっては入力しないものであっても、もちろんよい。例えば、常に、前回に動作OS1をシャットダウンした時点の状態に復旧するというものであってもよい。

40

【0091】

データ復旧部204は、入力された復旧の方法を決める判断情報、または予め決められている復旧方法に従い、記憶装置102に書き戻すためのデータを記憶退避装置203より取り出し、記憶装置102に書き込みを行う。記憶装置102への書き込みが終了したら、それはすなわち障害のない動作OS1の状態が再現できたということを意味するものであるから、再び処理実行部101が動作OS1の起動を行うことによって、障害の復旧を行うことができる。

【0092】

なお、記憶装置102の状態をある時点の状態に復旧した後は、記憶退避装置203に保存されているデータのうち、その時点以降の任意の状態に復旧させるのに必要なデータを

50

破棄するようにする構成も可能である。

【0093】

ところで、動作OS1の障害の発生は認識できたが、記憶装置102の状態をどの時点の状態に戻せばよいか把握できない場合が考えられる。このような場合のために、例えば、データ復旧部204に、記憶装置102の状態を、ユーザが指定した状態に仮に戻す機能を設けるようにしてもよい。この場合、ユーザはその仮に戻した記憶装置102の状態で作動OS1を起動して障害が復旧しているかどうかを判断するといった操作を、指定する記憶装置102の状態を少しずつ過去に溯らせるようにして繰り返し行うことによって、障害が復旧する状態を見出し、このときの記憶装置102の状態に確定させる指示をユーザが管理用OS2に与え、この指示によって、データ復旧部204は、記憶装置102の状態を確定させるようにしてもよい。 10

【0094】

(第3の実施形態)

次に、図11に、本発明の第3の実施形態に係るデータ管理システムを含む計算機システムの構成例を示す。この構成例は、データ管理システムが、図8の管理用OS2の他に、遠隔管理装置3をも含むものである。データの退避や障害からの復旧の処理については基本的には第2の実施形態と同様である。以下では、図8の構成例と相違する部分を中心に説明する。

【0095】

管理用OS2には、例えばネットワーク4等を介して、遠隔管理装置3が接続されている。遠隔管理装置3は、1つでも複数でもよい。ここでは、説明を簡略化するために、1つであるとする。なお、遠隔管理装置3は、例えば、典型的には、インターネットや電話回線等のネットワークを介してサーバセンタなどの離れた場所に設置する形態が可能であるが、これに限定されるものではなく、例えば、計算機の補助装置として遠隔管理装置3を採用し、動作OS1、管理用OS2、遠隔管理装置3を一体として設置する形態も可能である。 20

【0096】

管理用OS2のデータ抽出部202は、記憶退避装置203に障害復旧に必要なデータを保存するのみならず、(遠隔記憶退避装置302へ保存させるために)遠隔管理装置3のデータ受信部301へも障害復旧に必要なデータを送信する。 30

【0097】

記憶退避装置203と遠隔記憶退避装置302へのデータ保存方法には、記憶退避を2重化する第1の形態(記憶退避装置203と遠隔記憶退避装置302に同一のデータを保存する方法)、一部のデータのみについて2重化する第2の形態(遠隔記憶退避装置302には、記憶退避装置203に保存するデータの一部のデータのみを保存する方法)、データごとに保存方法を制御する第3の形態(データによって記憶退避装置203のみに保存するか遠隔記憶退避装置302のみに保存するか両方に保存するかを決定する方法)など種々の方法がある。

【0098】

第2の形態の具体例としては、例えば、管理用OS2のデータ抽出部202は、管理用OS2自身の持つ記憶退避装置203へは、障害復旧に必要な全てのデータを保存し、遠隔管理装置3のデータ受信部301へは、障害復旧に必要なデータのうち特に重要なデータ(例えば、システムのライブラリファイルに関するデータ等)のみに限定して送信するという方法も考えられる。 40

【0099】

データ受信部301は、受信したデータの全てを遠隔記憶退避装置302に保存する。なお、データ受信部301は、受信したデータのうち必要と判断される一部のデータを遠隔記憶退避装置302に保存するようにする構成も可能である。

【0100】

このように、障害復旧に必要なデータを記憶退避装置203だけでなく遠隔記憶退避装置 50

302にも保存しておくことにより、より安全性を高めることが可能である。

【0101】

例えば、動作OS1と管理用OS2の両者が同一の場所にある場合には、火災などによって物理的に同時に破壊される場合があり得るが、遠隔管理装置3をインターネット等で接続された他の場所にあるサーバセンタ等に設置することにより、障害復旧に必要なデータをより確実に保管することができ、たとえ動作OS1と管理用OS2の両者が物理的に同時に破壊されたとしても、遠隔記憶退避装置302に保存されたデータを用いて障害復旧を行うことができる。

【0102】

また、典型的な利用例としては、遠隔管理装置3をインターネットで接続された特に安全に管理されたサーバセンタ内に設置することにより、管理用OS2の破壊などのリスクを意識することなく、いつでも障害の復旧できる計算機をユーザに提供できるという利点が生じる。

【0103】

遠隔データ復旧部303は、先に説明したデータ復旧部204の機能と同様、記憶装置102に書き戻すべきデータを取り出す機能を持つ。遠隔データ復旧部303は、遠隔記憶退避装置302より必要なデータを取り出し、データ復旧部204に送信する。データ復旧部204は、遠隔データ復旧部303から受信したデータのみを利用し、もしくは記憶退避装置203から取り出したデータのみを利用し、または遠隔データ復旧部303からのデータおよび記憶退避装置203からのデータを利用して、障害復旧に必要なデータの記憶装置102への書き込みを行う。あるいは、遠隔データ復旧部303は、データ復旧部204にデータを送信することなく、直接に記憶装置102へデータを送り、記憶退避装置203のデータを利用しないということも考えられる。また、これらの処理はオフラインで行われることも考えられる。つまり、通常はネットワークを介して遠隔管理装置3をサーバセンタで運用するが、障害発生 of 報告を受けたサーバセンタ職員が遠隔記憶退避装置302（ハードディスクなど）を動作OS1の設置場所に持参し、記憶装置102を復旧させるような場合もあり得る。

【0104】

なお、図11の構成例は、図8の構成例に遠隔管理装置3を付加した場合であったが、図1に遠隔管理装置3（ただし、遠隔データ復旧部303は除く）を付加した構成も可能である。

【0105】

（第4の実施形態）

次に、図12に、本発明の第4の実施形態に係るデータ管理システムを含む計算機システムの構成例を示す。この構成例は、障害復旧作業をさらに迅速に行えるようにするため、図1の記憶装置102の部分に記憶変換機能を付加したものである。データの退避の処理については基本的には第1の実施形態と同様である。以下では、第1の実施形態と相違する部分を中心に説明する。

【0106】

ここで、障害発生検知時には、図12の動作OS1側の記憶退避装置203は、管理用OS2側に接続されており、第1の実施形態と同様に、障害からの復旧に必要なデータが保存されているものとする（図1参照）。

【0107】

また、障害発生が検知された後には、本実施形態では、動作用OS1の持っていた記憶退避装置（図示せず）が取り外されるとともに、管理用OS2の持っていた記憶退避装置203が取り外されて（実際に移送されて）動作OS1側へ接続される（図12は、この状態を示している）。

【0108】

本実施形態では、処理実行部101は、データの読み出しや書き込みを記憶変換装置401に対して行う。

【0109】

記憶変換装置401は、処理実行部101から受けた書き込みデータを、接続された記憶退避装置203へそのまま書き込む。ここで、そのまま書き込むとは、時系列を考慮して必要なデータを破壊しないように書き込むという意味であり、これについては既に説明した図1におけるデータ抽出部202が記憶退避装置203に書き込みを行う場合と同様である。

【0110】

他方、処理実行部101がデータの読み出しを行う際には、記憶変換装置401より行う。

【0111】

記憶変換装置401は、（障害発生検知後に管理用OS2側から取り外されて接続された）記憶退避装置203からデータを読み出すにあたって、障害発生時点以降のデータではなく、障害発生直前の時点のデータを読み出す。例えば、ユーザから3日前と指定された場合には、3日前のデータを読み出すような機能を持つ。

【0112】

記憶変換装置401は、例えば、データ抽出部202の機能とデータ復旧部204のような機能を持てばよい。

【0113】

このように、処理実行部101が記憶退避装置203へのデータの書き込みや読み出しを記憶変換装置401を介して行うことにより、あたかも通常の記憶装置へのデータの書き込みや読み出しを行っているかのように動作させることができる。すなわち、記憶変換装置401と記憶退避装置203の組み合わせでできた装置を、一つの記憶装置102'とすると、これが図1における通常の記憶装置102と同等の機能で動作することになる。

【0114】

また、記憶退避装置203が移動された後には、管理用OS2のデータ抽出部202は、記憶装置102'からのデータ（実際には、記憶変換装置401からのデータ）を受信し、記憶退避装置203の代わりに新たに接続した記憶退避装置203'への書き込みを、第1の実施形態と同様に行えばよい。

【0115】

このように、記憶変換装置401を追加した障害復旧を行うことにより、記憶装置102へのデータの書き戻しのような復旧作業を行うことなく、瞬時に動作OS1を障害発生前の状態で動作させることができる。この方法によれば、瞬時に障害を復旧することができ、例えばオンラインショッピングサイトのサービス等を行っている計算機システムのように、動作OS1を長期間停止できないあるいは無停止で運用させることが重要であるような算機システムには特に効果的である。

【0116】

なお、この構成を障害復旧中の一時的な特殊構成と考えず、最初から記憶装置及び記憶退避装置が、いずれも、記憶変換装置401と記憶退避装置203を包含する装置であるとすれば、記憶装置と記憶退避装置とを区別することなく、障害が発生するたびに両者を入れ換えるようにする構成も可能である。この場合、管理用OS2において、データ抽出部202の役割を記憶変換装置401が果たすようにすれば、データ抽出部202を省くことが可能である。

【0117】

また、図12では、管理用OS2の記憶退避装置203のデータをもとに障害復旧を行う例を示したが、図12の構成例に更に図11の遠隔管理装置3（ただし、遠隔データ復旧部303は除く）を付加し、上記と同様に、障害発生検知時に、管理用OS2の記憶退避装置203または遠隔管理装置3の遠隔記憶退避装置302を動作OS1側に接続して、記憶退避装置203または遠隔記憶退避装置302のデータをもとに障害復旧を行うようにする構成も可能である。

【0118】

10

20

30

40

50

また、この例は、障害発生時の一時的な復旧措置を迅速化させる目的のものであるから、一時的に管理用OS2を取り外し、記憶変換装置401を加えた動作OS1のみで動作させるという運用も可能である。

【0119】

(変形例)

本実施形態では、第1～第4の実施形態で説明した各構成に関する変形例について説明する。

【0120】

<1>各実施形態に係る計算機システムを高性能化する一つの手法として、保存データ量を削減することも可能である。管理用OS2のデータ抽出部202が、抽出したデータを記憶退避装置203にデータを保存する際、データを逐次そのまま記憶させていくのでは、保存すべきデータが膨大になるが、既に記憶されている過去の退避データを記憶退避装置より読み出し、再利用することにより、保存データを削減し、性能を向上させることが可能である。

10

【0121】

例えば、内容の変更されていない単位データ（ファイル）の保存を省略したり、ごく一部だけが変更されている単位データ（ファイル）に対してはその差分データのみを保存するなどにより、保存データを大幅に削減することができる。これは、遠隔管理装置3においてデータ受信部301が遠隔記憶退避装置302にデータを保存する際にも用いることができる。

20

【0122】

また、OS状態検出部201が、動作OS1の実行が終了したことを検出し、その情報を記憶装置102からデータを取り出す際に利用することができる。

【0123】

利用方法の一例としては、例えば、動作OS1の実行終了時には、通常のデータ保存時よりも詳細なデータを保存する方法がある。つまり、動作OSの起動途中とは異なる実行終了時特有のデータを追加して保存する。

【0124】

また、他の利用方法例としては、動作OS1の実行終了後には、記憶装置102のデータ内容全てを保存し、それ以外の場合には前回の動作OS1の実行終了時のデータに基づく差分データのみを保存するようにする方法がある。これによって、障害復旧に必要なデータの保存をより効率的に行うことができる。

30

【0125】

また、このような方法を取ることににより、障害復旧処理時にどのデータを採用すべきか判りやすくなる場合がある。例えば、「3日前の午後3時21分のデータ」と表示されているよりも「3日前の、一番最後にOSの実行を終了した時のデータ」と表示されている方が好都合である場合があると考えられる。

【0126】

このように、動作OSの実行が終了したことを検出し、その情報を記憶装置からデータを取り出す際に利用することによって、データの保存処理を効率化したり、障害復旧処理を簡便にしたりする効果が期待できる。

40

【0127】

また、OS状態検出部201が、動作OS1の実行が終了したことを検出することができるため、例えば、「障害復旧機能の起動」といった特別の作業を行う必要はなく、計算機のユーザからは、実際には動作OS1と管理用OS2とが動いているにもかかわらず、あたかも動作OS1だけが動いていて、その通常使っている動作OS1をそのまま使っている感覚で使えるという利点がある。

【0128】

<2>また、OS状態検出部201が、動作OS1の実行処理部101が記憶装置102のデータに変更を加えたことを検出し、その情報を記憶装置102からデータを取り出す

50

際に利用することができる。一般に、管理用OS 2が動作OS 1の記憶装置102の内容だけを見てその内容の変化をリアルタイムに知るためには記憶装置102の全体を繰り返し検索する必要があるが、動作OS 1が記憶装置102の内容に変更を加えたことを直接にデータ抽出部202に伝える手段があれば、記憶装置102の内容のうち変更されたと判っている部分のみを検索すれば済むため、データを取り出す効率が良くなる。例えば、動作OS 1の処理実行部101のデータ処理に関するシステムコールを改造し、データ抽出部202に記憶装置に関連する情報（ファイル名やノード番号等）を伝える仕組みにすることによりこの機能が実現できる。

【0129】

<3>また、本実施形態を一般の計算機に実装する一つの手法として、他のOSを実行するための仮想計算機ソフトウェアを用いる手法がある。 10

【0130】

仮想計算機ソフトウェアとしては例えばVMware (J. Sugerman, et al., Virtualizing I/O Devices on VMware Workstation ■s Hosted Virtual Machine Monitor, 2001 USENIX Conference, 2001/7/25) がある。

【0131】

この場合、一つの主たるOS（ホストOSと呼ぶ）上で動作する仮想計算機ソフトウェアを用意する。この仮想計算機は、CPUを直接実行し、さまざまな周辺デバイスを仮想化 20して組み込むことにより、あたかも実際の計算機が動いているかのごとく動作し、実際の計算機とほぼ同等の性能で動作するものである。仮想計算機ソフトウェアは、ホストOS上に1つまたは複数動作させることができ、それぞれの仮想計算機に副たるOS（ゲストOSと呼ぶ）を実装することができるため、結果として1台の計算機上に2つ以上所望する数だけOSを同時に動作させることができる。この仕組みを用いて、動作OS 1と管理用OS 2とを同時に1台の計算機上に実装することができる。

【0132】

これにより、特別な計算機装置を用意することなく、1台の計算機で容易に本計算機システムを実装することが可能となる。

【0133】

例えば、本実施形態を計算機に実装するにあたって、図13に例示するように、管理用OS 2は、他のOSを実行するための仮想計算機ソフトウェア22を持ち、動作OS 1は該ソフトウェア22上で動作するという構成が可能である。この様子を障害復旧を確実に 30行うためには、動作OSと管理用OSを同時に動かすことが望ましいが、仮想計算機ソフトウェアを用いることにより、特別な計算機装置を用意することなく、1台の計算機で容易に本計算機システムを実装することが可能となる。

【0134】

また、仮想計算機ソフトウェアを使う利点としては、動作OS 1と管理用OS 2とが同一のハードウェアを共有していることから、管理用OS 2のOS状態検出部が、動作OS 1の処理実行部101からの情報を受け取るための実装が容易になり、また、管理用OS 2 40のデータ抽出部202が動作OS 1の記憶装置102からのデータを受け取るための実装も容易になる。

【0135】

さらには、障害の復旧時においても、データ復旧部204が記憶退避装置203からのデータをもとに記憶装置102に復旧のためのデータを書き込む実装も容易になり、また、その性能も向上することが期待できるという効果がある。

【0136】

この方法の場合、ゲストOSはホストOSの上に実装される以上、安全性はゲストOSがホストOSを上回ることとは通常考えられない。なぜなら、ホストOSが破壊された場合、必然的にゲストOSも破壊される可能性が高いからである。したがって、動作OS 1は必 50

ずゲストOSとして実装され、管理用OS 2はゲストOSであってもホストOSであってもよい。

【0137】

また、1つの管理用OS 1に対して複数の動作OS 2がゲストOSとして実装されることも可能であるし、また、遠隔管理装置3も、このゲストOSまたはホストOSとして実装されることも可能である。

【0138】

また、この方法の場合、ホストOSを起動すると同時にゲストOSを起動させることができるため、計算機のユーザからは、実際には動作OS 1と管理用OS 2とが動いているにもかかわらず、あたかも動作OS 1だけが動いていて、その通常使っている動作OS 1をそのまま使っている感覚で使えるという環境を自然に実現することができるという効果もある。 10

【0139】

<4>また、以上では、1つの管理用OS 2が1つの動作OS 1を管理対象とするものであったが、図14に例示するように、1つの管理用OS 2が複数の動作OS 1を管理対象とすることも可能である。図14は、n台の計算機A1～An上でそれぞれ動作する動作OS 1を、それら計算機A1～AnとLANあるいはインターネット等のネットワーク8で接続された計算機B上で動作する管理用OS 2が管理対象とする例である。

【0140】

なお、以上の各機能は、ソフトウェアとして記述し適当な機構をもったコンピュータに処理させても実現可能である。 20

また、本実施形態は、コンピュータに所定の手段を実行させるための、あるいはコンピュータを所定の手段として機能させるための、あるいはコンピュータに所定の機能を実現させるためのプログラムとして実施することもできる。加えて該プログラムを記録したコンピュータ読取り可能な記録媒体として実施することもできる。

【0141】

なお、本発明は上記実施形態そのままに限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で構成要素を変形して具体化できる。また、上記実施形態に開示されている複数の構成要素の適宜な組み合わせにより、種々の発明を形成できる。例えば、実施形態に示される全構成要素から幾つかの構成要素を削除してもよい。さらに、異なる実施形態にわたる構成要素を適宜組み合わせてもよい。 30

【0142】

【発明の効果】

本発明によれば、計算機の障害復旧に必要なデータを従来よりも確実に保存することができる。

【図面の簡単な説明】

【図1】本発明の第1の実施形態に係るデータ管理システムを含む計算機システムの構成例を示す図

【図2】動作OS 1及び管理用OS 2の実現形態について説明するための図

【図3】動作OS 1及び管理用OS 2の実現形態について説明するための図 40

【図4】管理用OSの処理手順の一例を示すフローチャート

【図5】管理用OSの処理について説明するための図

【図6】管理用OSの処理について説明するための図

【図7】管理用OSの処理について説明するための図

【図8】本発明の第2の実施形態に係るデータ管理システムを含む計算機システムの構成例を示す図

【図9】管理用OSの処理手順の一例を示すフローチャート

【図10】管理用OSの処理について説明するための図

【図11】本発明の第3の実施形態に係るデータ管理システムを含む計算機システムの構成例を示す図 50

【図 1 2】 本発明の第 4 の実施形態に係るデータ管理システムを含む計算機システムの構成例を示す図

【図 1 3】 動作 OS 1 及び管理用 OS 2 の実現形態について説明するための図

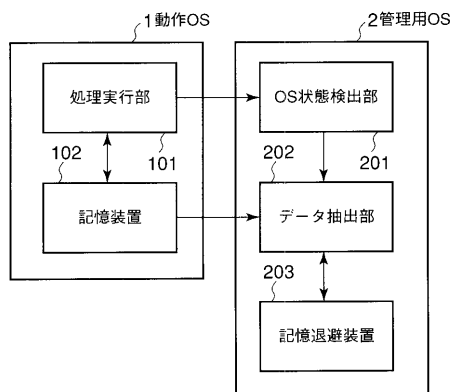
【図 1 4】 動作 OS 1 及び管理用 OS 2 の実現形態について説明するための図

【符号の説明】

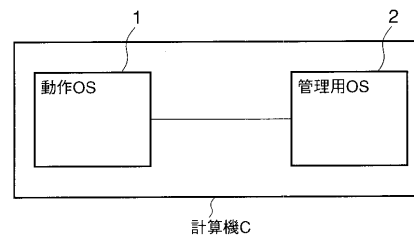
1 … 動作 OS、2 … 管理用 OS、3 … 遠隔管理装置、4, 8 … ネットワーク、22 … 仮想計算機ソフトウェア、101 … 処理実行部、102, 102' … 記憶装置、201 … OS 状態検出部、202 … データ抽出部、203, 203' … 記憶退避装置、204 … データ復旧部、301 … データ受信部、302 … 遠隔記憶退避装置、303 … 遠隔データ復旧部、401 … 記憶変換装置、A, A1 ~ An, B ~ D … 計算機

10

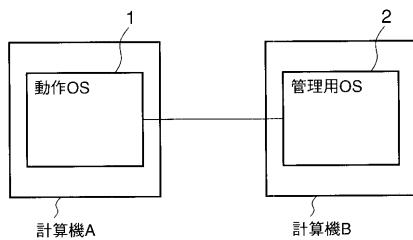
【図 1】



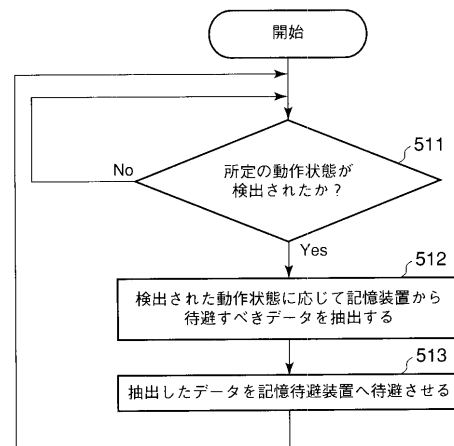
【図 3】



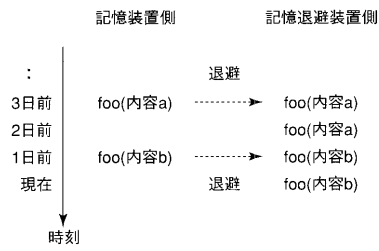
【図 2】



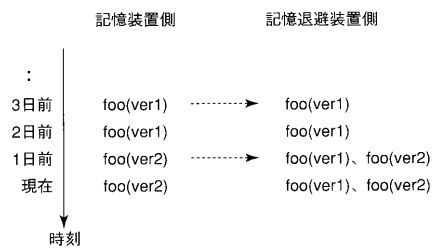
【図 4】



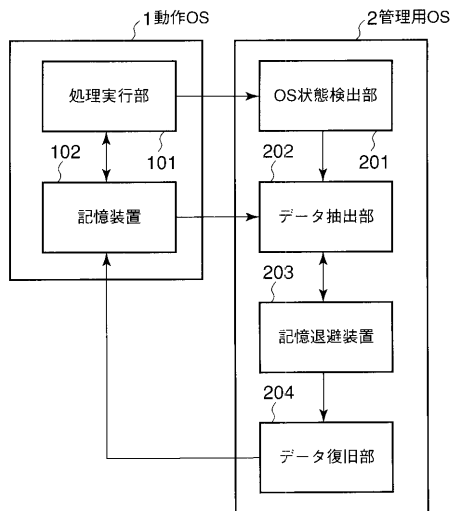
【図 5】



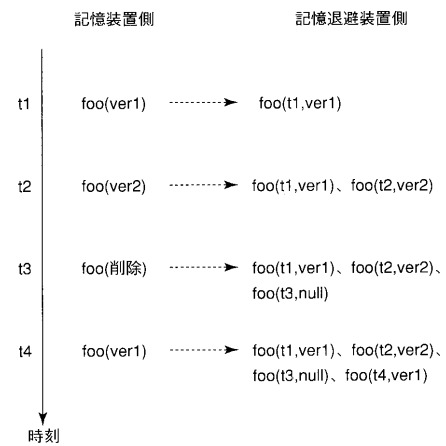
【図 6】



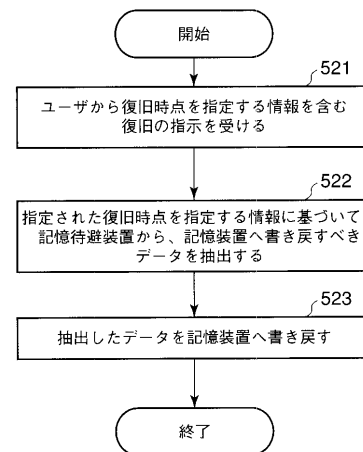
【図 8】



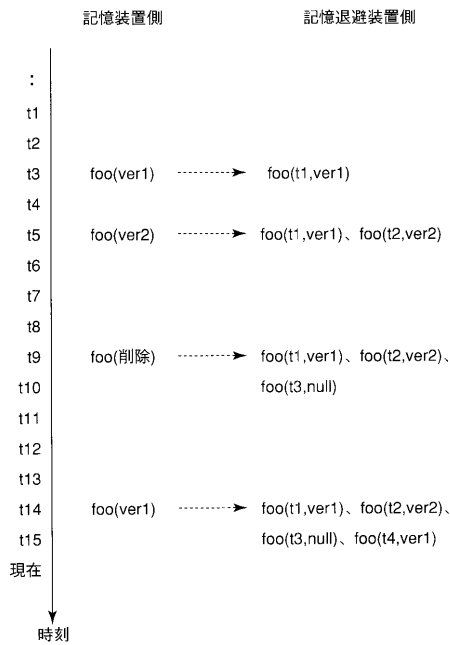
【図 7】



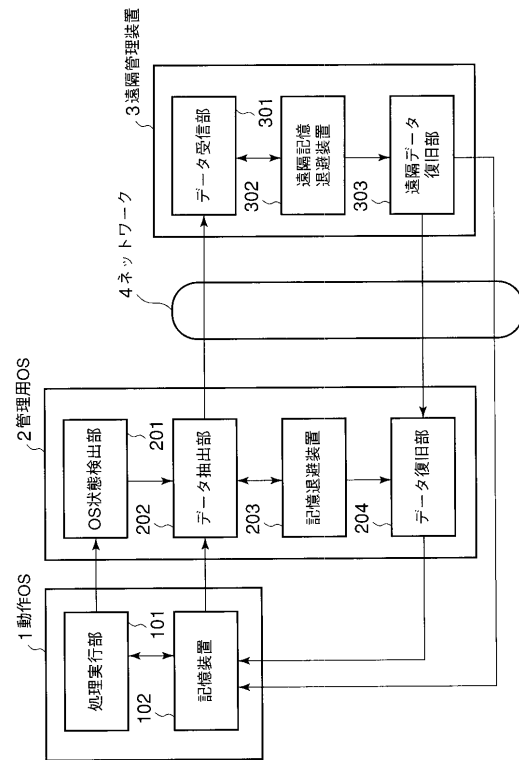
【図 9】



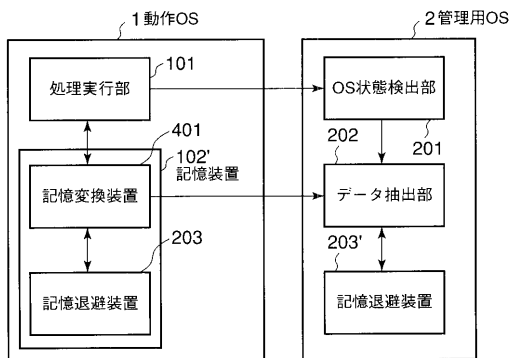
【図 10】



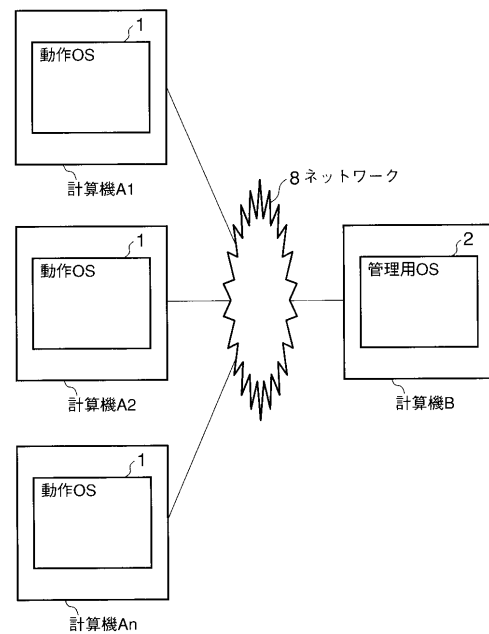
【図 11】



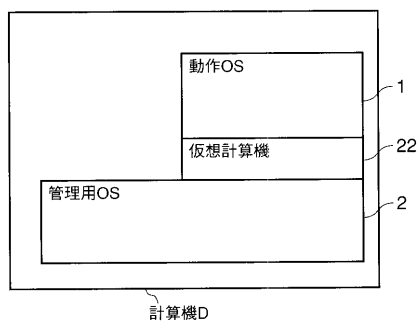
【図 12】



【図 14】



【図 13】



フロントページの続き

(72)発明者 高橋 俊成

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

Fターム(参考) 5B042 GA21 MA05 MC01

5B098 DD01 HH01