

(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2010-44759

(P2010-44759A)

(43) 公開日 平成22年2月25日(2010.2.25)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 12/00 (2006.01)	G06F 12/00 545F	5B065
G06F 3/06 (2006.01)	G06F 3/06 301A	5B077
G06F 13/38 (2006.01)	G06F 3/06 301Z	5B082
	G06F 13/38 310B	
	G06F 12/00 514K	

審査請求 有 請求項の数 21 O L 外国語出願 (全 48 頁)

(21) 出願番号 特願2009-180460 (P2009-180460)
 (22) 出願日 平成21年8月3日 (2009.8.3)
 (62) 分割の表示 特願2003-540836 (P2003-540836) の分割
 原出願日 平成14年10月7日 (2002.10.7)
 (31) 優先権主張番号 10/017, 923
 (32) 優先日 平成13年10月30日 (2001.10.30)
 (33) 優先権主張国 米国 (US)

(特許庁注：以下のものは登録商標)

1. リナックス

(71) 出願人 504166370
 ケイシー ケイ チャン
 アメリカ合衆国 カリフォルニア9050
 3 トランス ピー. オー. ボックス 1
 3906
 (74) 代理人 100101454
 弁理士 山田 卓二
 (74) 代理人 100081422
 弁理士 田中 光雄
 (74) 代理人 100091524
 弁理士 和田 充夫
 (74) 代理人 100125874
 弁理士 川端 純市
 (74) 代理人 100113170
 弁理士 稲葉 和久

最終頁に続く

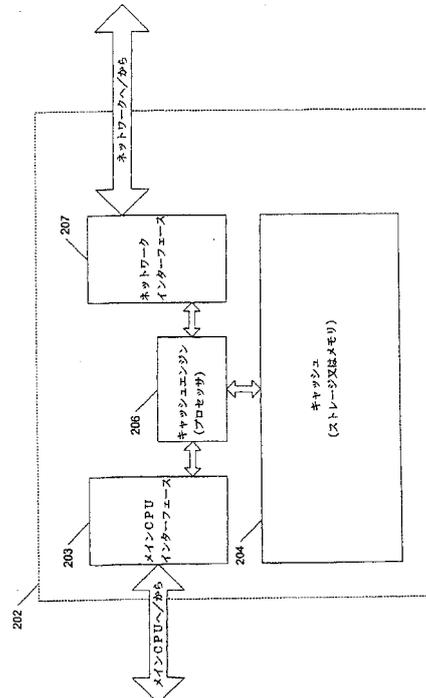
(54) 【発明の名称】 ネットワークを越えてファイルを配布するためのローカル化された読み込み専用記憶装置

(57) 【要約】 (修正有)

【課題】 コンピュータを管理し、ネットワークを介してファイルを分配する手段を簡素化する。

【解決手段】 読み込み専用記憶装置202は、コンピュータインターフェース203、プロセッサ206、ネットワークインターフェース207、及びデータ格納装置204を備えており、コンピュータ及びファイルサーバと交信できる。プロセッサ206は、コンピュータからファイル要求を受信すると、(1)データ格納装置204にファイルが保持されている場合、読み込み専用として、要求されたファイルをコンピュータに供給するか、(2)ファイルがファイルサーバから入手できる場合、そのファイルをファイルサーバから入手し、データ格納装置204に保持してから読み込み専用として、入手したファイルをコンピュータに供給するか、(3)コンピュータに、ファイルが入手不可能であるという通知を送信する、といういずれかの方法で、コンピュータに対応する。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

以下の四要素で構成された記憶装置で、

- (a) デバイスプロセッサと、
 - (b) 前記デバイスプロセッサと通信できるように接続されており、一つコンピュータと通信できるようにこのコンピュータの中央処理装置(CPU)と前記デバイスプロセッサとの間に限り通信を可能にするコンピュータインターフェースと、
 - (c) 前記デバイスプロセッサと通信できるように接続されており、ネットワークを介して通信でき、かつ前記デバイスプロセッサと他のファイルサーバとの通信は排除しながら一つ以上の特定のファイルサーバとの間の通信を可能にするネットワークインターフェースと、
 - (d) 前記デバイスプロセッサと通信できるように接続されており、前記デバイスプロセッサが保存されたファイルを管理し、前記CPUは保存されたファイルを管理できず、かつ前記CPUは保存されたファイルを読み込みだけ行うことができるデータ格納装置と、
- を備え、

前記記憶装置において、前記CPUから最初のファイルの要求があった場合、前記デバイスプロセッサは、(1)要求された最初のファイルがデータ格納装置に存在するか否かを判断し、(2)その最初のファイルがデータ格納装置には存在しない場合、その最初のファイルを特定のファイルサーバから回収し、回収した最初のファイルをデータ格納装置に保存し、(3)回収したファイルを読み込み専用として前記CPUに供給する、ということ

【請求項 2】

請求項 1 に記載の記憶装置で、データ格納装置に前記ファイルが存在せず、また特定のファイルサーバから回収できない場合、前記ファイルは入手不可能であるとの通知を前記デバイスプロセッサが前記CPUに通知できるように接続されている記憶装置。

【請求項 3】

請求項 1 に記載の記憶装置で、前記CPUが前記記憶装置のネットワークインターフェースを介して、ネットワーク資源と通信できるように接続されている記憶装置。

【請求項 4】

請求項 1 に記載の記憶装置で、読み書き自由な媒体から成るデータ格納装置を備えた記憶装置。

【請求項 5】

以下の四つで構成されたコンピュータネットワークで、

- (a) 一つ以上の特定のファイルサーバと、
 - (b) ネットワーク資源と、
 - (c) 前記特定のファイルサーバと前記ネットワーク資源から離れて配置されており、ネットワークを介して前記ネットワーク資源と通信できるように接続されているコンピュータと、
 - (d) ネットワークを介して前記コンピュータ及び前記の特定のファイルサーバと通信できるように接続されており、デバイスプロセッサ、コンピュータインターフェース、ネットワークインターフェース、データ格納装置を装備している記憶装置と、
- を備え、

前記デバイスプロセッサは、他のファイルサーバとの通信を排除しながら前記の特定のファイルサーバとの通信を可能にするようなネットワークインターフェースを備えている、また、前記デバイスプロセッサと前記コンピュータの中央処理装置(CPU)の間に限り通信を可能にするコンピュータインターフェースも備えていると共に、さらに、前記CPUはデータ格納装置に保存されたファイルを管理できず、かつ前記CPUはデータ格納装置に保存されたファイルを読み込みだけを行うという形で、前記デバイスプロセッサはデータ格納装置に保存されたファイルを管理できるように接続される。そして、前記CPUから最初のファイルの要求があった場合、前記デバイスプロセッサは、(1)要求された最初の

10

20

30

40

50

ファイルがデータ格納装置に存在するか否かを判断し、(2)その最初のファイルがデータ格納装置には存在しない場合、その最初のファイルを特定のファイルサーバから回収し、回収した最初のファイルをデータ格納装置に保存し、(3)回収したファイルを読み込み専用として前記CPUに供給する、ということを行う、コンピュータネットワーク。

【請求項6】

請求項5に記載のコンピュータネットワークで、データ格納装置に前記ファイルが存在せず、また特定のファイルサーバから回収できない場合、前記ファイルは入手不可能であるとの通知を前記デバイスプロセッサが前記CPUに通知できるように接続されているコンピュータネットワーク。

【請求項7】

請求項5に記載のコンピュータネットワークで、コンピュータが前記記憶装置のネットワークインターフェースを介して、ネットワーク資源と通信できるように接続されているコンピュータネットワーク。

【請求項8】

請求項5に記載のコンピュータネットワークで、読み書き自由な媒体から成るデータ格納装置を備えたコンピュータネットワーク。

【請求項9】

コンピュータへファイルを供給する手順で、

(a)記憶装置がコンピュータからファイルの要求を受信するステップで、この記憶装置は、デバイスプロセッサ、コンピュータインターフェース、ネットワークインターフェース、データ格納装置を装備している。デバイスプロセッサは、他のファイルサーバとの通信を排除しながら一つ以上の特定のファイルサーバとの通信を可能にするネットワークインターフェースを備えていると共に、また、デバイスプロセッサとコンピュータの中央処理装置(CPU)の間に限り通信を可能にするコンピュータインターフェースを備えており、さらに、前記CPUがデータ格納装置に保存されたファイルを管理できず、かつ前記CPUはデータ格納装置に保存されたファイルを読み込みだけを行うという形で、前記デバイスプロセッサはデータ格納装置に保存されたファイルを管理できるように接続される、ステップと、

(b)前記デバイスプロセッサは、前記ファイルが前記データ格納装置に存在するか否かを判断する。

(c)前記デバイスプロセッサは、前記ファイルが前記データ格納装置に存在しない場合、そのファイルを特定のファイルサーバから回収し、回収したファイルをデータ格納装置に保存するステップと、

(d)前記デバイスプロセッサは、回収したファイルを読み込み専用として前記CPUに供給するステップと、

を含む、コンピュータへファイルを供給する手順。

【請求項10】

請求項9に記載のコンピュータへファイルを供給する手順には、データ格納装置に前記ファイルが存在せず、また特定のファイルサーバから回収できない場合、前記ファイルは入手不可能であるとの通知を前記デバイスプロセッサが前記CPUに通知する手順も含まれることを追記する、コンピュータへファイルを供給する手順。

【請求項11】

請求項9に記載のコンピュータへファイルを供給する手順には、前記デバイスプロセッサが、一つ以上の特定のファイルサーバから前記記憶装置のデータ格納装置に保存されたファイルの削除命令を受信した場合、そのファイルを削除することも含まれることを追記する、コンピュータへファイルを供給する手順。

【請求項12】

コンピュータにアップデートされたファイルを供給する手順で、

(a)記憶装置のデバイスプロセッサがファイルサーバから回収しうる複数のファイルのリストをコンピュータの中央処理装置(CPU)に供給するステップと、

(b) 前記CPUから最初のファイルを要求された場合、前記デバイスプロセッサがファイルサーバ内の複数のファイルの中から最初のファイルを回収するステップと、

(c) 前記デバイスプロセッサが、前記最初のファイルを記憶装置内のデータ格納装置に保存す。前記CPUがデータ格納装置に保存されたファイルを管理できず、かつ前記CPUがデータ格納装置に保存されたファイルを読み込みだけ行うという形で、前記デバイスプロセッサがデータ格納装置に保存されたファイルを管理できるように接続されるステップと、

(d) 前記デバイスプロセッサが、前記ファイルを読み込み専用として前記CPUに供給するステップと、

(e) 前記デバイスプロセッサが、保存されたファイルを削除命令を特定のファイルサーバから受信した場合、保存された最初のファイルをデータ格納装置から削除するステップと、

を含む、コンピュータにアップデートされたファイルを供給する手順。

【請求項 13】

請求項 12 に記載のコンピュータにアップデートされたファイルを供給する手順には、前記デバイスプロセッサが、最初のファイルのアップデートバージョンが存在するという通知をファイルサーバから受信した場合、データ格納装置に保存された最初のファイルをデータ格納装置から削除することも含まれることを追記する、コンピュータにアップデートされたファイルを供給する手順。

【請求項 14】

以下の四要素で構成された記憶装置で、

(a) デバイスプロセッサ。

(b) 前記デバイスプロセッサと通信できるように接続されており、一つコンピュータと通信できるようにこのコンピュータの中央処理装置(CPU)と前記デバイスプロセッサとの間に限り通信を可能にするコンピュータインターフェースと、

(c) 前記デバイスプロセッサと通信できるように接続されており、ネットワークを介して通信でき、かつ前記デバイスプロセッサと他のファイルサーバとの通信は排除しながら一つ以上の特定のファイルサーバとの間の通信を可能にするネットワークインターフェースと、

(d) 前記デバイスプロセッサと通信できるように接続されており、前記デバイスプロセッサが保存されたファイルを管理し、前記CPUは保存されたファイルを管理できず、かつ前記CPUは保存されたファイルを読み込みだけ行うことができるデータ格納装置と、

を備え、

前記記憶装置において、コンピュータが起動する際に、起動ファイル(ブートストラップファイルかオペレーティングシステムファイルか)の要求があった場合、前記デバイスプロセッサは、(1)要求された起動ファイルがデータ格納装置に存在するか否かを判断し、(2)その起動ファイルがデータ格納装置に存在しない場合、その起動ファイルを特定のサーバから回収し、回収した起動ファイルをデータ格納装置に保存し、(3)回収した起動ファイルを読み込み専用として前記CPUに供給する、ということを行う、記憶装置

【請求項 15】

請求項 14 に記載の記憶装置で、データ格納装置に前記起動ファイルが存在せず、また特定のファイルサーバから回収できない場合、前記起動ファイルは入手不可能であるとの通知を前記デバイスプロセッサが前記CPUに通知できるように接続されている記憶装置。

【請求項 16】

請求項 14 に記載の記憶装置で、前記CPUが記憶装置のネットワークインターフェースを介して、ネットワーク資源と通信できるように接続されている記憶装置。

【請求項 17】

請求項 14 に記載の記憶装置で、読み書き自由な媒体から成るデータ格納装置を備えた記憶装置。

10

20

30

40

50

【請求項 18】

以下の四つで構成されたコンピュータネットワークで、

(a) 一つ以上の特定のファイルサーバと、

(b) ネットワーク資源と、

(c) 前記の特定のファイルサーバと前記ネットワーク資源から遠隔に配置されており、ネットワークを介して前記ネットワーク資源と通信できるように接続されたコンピュータと、

(d) ネットワークを介して前記コンピュータ及び前記の特定のファイルサーバと通信できるように接続されており、デバイスプロセッサ、コンピュータインターフェース、ネットワークインターフェース、データ格納装置を装備している記憶装置と、

を備え、

前記デバイスプロセッサは、他のファイルサーバとの通信を排除しながら前記の特定のファイルサーバとの通信を可能にするようなネットワークインターフェースを備えている、また、前記デバイスプロセッサと前記コンピュータの中央処理装置(CPU)の間に限り通信を可能にするコンピュータインターフェースも備えていると共に、さらに、前記CPUはデータ格納装置に保存されたファイルを管理できず、かつ前記CPUはデータ格納装置に保存されたファイルを読み込みだけを行うという形で、前記デバイスプロセッサはデータ格納装置に保存されたファイルを管理できるように接続される。そして、コンピュータが起動する際に、起動ファイル(ブートストラップファイルかオペレーティングシステムファイルか)の要求があった場合、前記デバイスプロセッサは、(1)要求された起動ファイルがデータ格納装置に存在するか否かを判断し、(2)その起動ファイルがデータ格納装置に存在しない場合、その起動ファイルを特定のサーバから回収し、回収した起動ファイルをデータ格納装置に保存し、(3)回収した起動ファイルを読み込み専用として前記CPUに供給する、ということを行う、コンピュータネットワーク。

【請求項 19】

請求項 18 に記載のコンピュータネットワークで、データ格納装置に前記起動ファイルが存在せず、また特定のファイルサーバから回収できない場合、前記起動ファイルは入手不可能であるとの通知を前記デバイスプロセッサが前記CPUに通知できるように接続されているコンピュータネットワーク。

【請求項 20】

請求項 18 に記載のコンピュータネットワークで、前記CPUが記憶装置のネットワークインターフェースを介して、ネットワーク資源と通信できるように接続されているコンピュータネットワーク。

【請求項 21】

請求項 18 に記載のコンピュータネットワークで、読み書き自由な媒体から成るデータ格納装置を備えたコンピュータネットワーク。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ネットワークと共に用いられるコンピュータ装置およびこのような装置を使用する方法に関する。

【背景技術】

【0002】

典型的なコンピュータは、市販品であろうと、個人用、ビジネス用、特殊用途、その他の目的に作られた特注品であろうと、多くの部品から構成される。コンピュータの一般的な構成部品は、プロセッサ、メモリ、記憶装置、インプットやアウトプット装置およびネットワークインターフェースなどである。また、オペレーティングシステム、例えばマクロソフトウィンドウズ(登録商標)、マックOS、ユニックス、リナックス等は、構成部品とそれらの機能の制御を扱う。それに加え、オペレーティングシステムは、アプリケーションやオペレーティングシステム自身からのファイルの要求を処理する。ファイルの要求

10

20

30

40

50

があると、オペレーティングシステムは、ローカルの記憶装置の中から、あるいはコンピュータがネットワークに接続されている場合、ネットワークサーバからファイルを供給することを試みる。

【0003】

本日のコンピュータを取り巻く環境上には、多くの異なった種類の記憶装置とネットワークインターフェースが存在するため、オペレーティングシステムは、一般的にコンピュータに接続可能なほとんど全ての記憶装置やネットワークインターフェースにアクセスできるようプログラムされている。従って、各記憶装置について、記憶装置の種類（例えば磁気や光学を利用したもの）や、インターフェースの種類（例えばIDEやSCSI）や、記憶装置における物理的パラメータ（例えばシリンダ、セクタ、ヘッド、トラックなどの数）を基に、オペレーティングシステムはプログラムされなければならない。また、コンピュータがネットワークに接続される場合、オペレーティングシステムは、さまざまな種類のネットワークインターフェース（例えばEthernet（登録商標）、Token Ring、ATM）や、ネットワークプロトコル（例えばTCP/IP、IPX、AppleTalk、SNA）や、ネットワーク資産（例えばサーバ）と通信するのに使用される特定の方法に対応できるよう、プログラムされなければならない。

10

【0004】

さらに、オペレーティングシステムは、ネットワークを介して記憶装置上のファイル进行管理することができなければならない。ローカルの記憶装置上、一般的にはオペレーティングシステムはファイル进行管理するために、通常はファイル割当表と呼ばれるルックアップテーブルが指数を使用する。オペレーティングシステムは、ネットワークを介して、ネットワークサーバと通信したりファイルを取得できるようにプログラムされなければならない。従って、ファイル取得プロセスを簡素化するために、オペレーティングシステムがネットワークから受け取ったファイル情報を、前記のファイル割当表に類似した形式に変換することは、しばしば有効である。

20

【0005】

オペレーティングシステムが、構成部品とそれらの機能を管理する既存の方法は、コンピュータを複雑化している。その複雑さは、コンピュータのセットアップのプロセスで容易に伺うことができる。通常、コンピュータが一般的に使用出来るように次のような処置を行う：（a）オペレーティングシステムのインストールを開始することのできる起動装置から起動すること、（b）ローカル記憶装置をパーティションしフォーマットすること、（c）オペレーティングシステム一式をインストールすること、（d）ディスプレイカードやネットワークインターフェースのようなハードウェア構成要素の設定を行うこと、そして（e）アプリケーションをインストールしたり、設定したりすること、などを行う。

30

【0006】

コンピュータの複雑さは、コンピュータが使用されている間でも明らかである。アプリケーションとオペレーティングシステムは、定期的な修正やアップデートを必要とすることがあり、修正ファイルやアップデートのインストールは、しばしばアプリケーションの旧式のバージョン、またはオペレーティングシステムの構成要素のアンインストールを伴う。さらにファイルは、プログラムエラー、ユーザーのエラー、またはコンピュータウイルスのため破損することがある。その場合、破損したファイルは、取り替えられるかまたは修復される必要があり、この取り替えあるいは修復のために、破損したファイルを使用するアプリケーションを再インストールしたり、場合によっては、オペレーティングシステム自体を再インストールしなければならないこともある。

40

【0007】

コンピュータの使用に伴う複雑さは、通常、高価なメンテナンスとサポートコストを招くこととなる。一般的な商業環境において、サポートコストは、1ユーザーあたりまたは容易に1コンピュータ装置あたり数千ドルにも達しうる。さらに、コンピュータ関連の問題が生じると、仕事の生産性がかなり減少するか、あるいはしばしばゼロになる傾向があ

50

ることから、コンピュータを維持するコストは増加する。

【0008】

記憶装置に関連する問題を除いて、ハードウェアの不良に関連する問題は、しばしば短時間で解決されうる。例えば、ビデオカードまたはネットワークインターフェースカードなどの壊れた部品は、同一の部品と迅速に取り替えられることが可能である。しかしながら、問題が記憶装置や記憶装置に保存されたファイルに関連するならば、コンピュータの修理は長いプロセスとなりうる。コンピュータのハードドライブが誤動作するか、または破損したファイルが問題を起こす場合、修理と回復プロセスには、ハードドライブの再初期化、オペレーティングシステムの再インストールおよび/またはアプリケーションの再インストールを伴うこともある。

10

【0009】

現在、コンピュータの複雑さを軽減するために、問題が起こった時の回復プロセスを短くしたり、回復プロセス自体の必要性を最小化したりするために、多数の手段が利用可能である。いくつかの一般的な手段としては、記憶装置の複製を作成すること、ネットワークからコンピュータを起動すること、専門的なコンピュータ管理ソフトウェアを利用すること、そして、ファイルレベルのセキュリティを適用することなどの手段がある。

【0010】

記憶装置の複製を作成することによって、インストールプロセスは短くなりうる。そのためには、最初にコンピュータにアプリケーション一式と共に完全にセットアップする。その後、記憶装置は、「イメージ」ファイルとして複製される。そして、そのイメージファイルは、コンピュータを初期状態にリセットしたり、または同様に装備されたコンピュータをセットアップするのに使用できる。多くの消費者向けコンピュータには、記憶装置を出荷時の状態に復元するのに使用できる、イメージを含む回復CD-ROMが付属されている。この記憶装置の複製を利用する上での短所は、記憶装置に保存されたオペレーティングシステム、アプリケーション、または他のファイルに変更があると、記憶装置の新しいイメージを常に作成しなければならないということである。もし、記憶装置が古いイメージから復元された場合、インストールされたソフトウェアに修正またはアップデートを適用する必要があり、この手段は複雑となりえる。

20

【0011】

ネットワークの起動手段は、しばしばネットワークサーバから必要なファイルをダウンロードするという単純な機能を果たすコンピュータと共に使用される。このようなコンピュータは通常、BOOTP、TFTP、またはPXEなどの周知なネットワークサービスを使用して、オペレーティングシステムの少量で基本的な部分をダウンロードおよび実行する。これを受けて、オペレーティングシステムは、残りのオペレーティングシステムの部分とアプリケーションのダウンロードを始めることができる。この手段の欠点は、コンピュータにローカル記憶装置がない場合、電源を入れるか、またはリセットする度に、必要なファイルをネットワークサーバからダウンロードするという同じ起動プロセスをたどらなければならないということである。コンピュータにローカル記憶装置がある場合、ダウンロードされたファイルがローカルに保存されるので、このプロセスは有益である。しかしその場合でも、ハードウェア構成要素や、ローカル記憶装置に保存されたファイルの管理等しばしば複雑なタスクを、ネットワークを通してダウンロードされたオペレーティングシステムが担うことになる。

30

40

【0012】

コンピュータ管理ソフトウェアを用いる手段は、エージェント、デーモン、またはサービスといった付加ソフトウェアコンポーネントを加えることによって、オペレーティングシステムの機能を強化させるために使用される。この手段を使った典型的な例は、コンピュータウィルスの感染を防ぐために、保存されたファイルを絶えずスキャンするウイルス対策ソフトウェアを使用することである。この手段の導入方法としては、ローカル記憶装置の重要なファイルを絶えず監視して、いかなるファイルの損傷や破損に対する自己修正を試みるソフトウェアコンポーネントを加えることが挙げられる。これらのソフトウェア

50

の追加は、コンピュータ管理ツールの一部として、サーバから出されたファイルのアップデートを処理するというソフトウェアの機能の追加にもなる。この手段の欠点は、エージェント、デーモン、サービスといったソフトウェアコンポーネントが、オペレーティングシステムに高く依存しているということである。オペレーティングシステムは、これらのソフトウェアコンポーネントが適切に稼働するよう、ローカル記憶装置やネットワークインターフェースを管理するなどの必要な機能を用意する必要がある。

【0013】

また、多くのオペレーティングシステムは、重要なファイルを破損しうるコンピュータウイルス、無権限のアクセス、ユーザーの誤操作、またはアプリケーションエラーに対する一定のレベルの保護を提供するために、ファイルレベルやディレクトリレベルセキュリティを適用することができる。この手段の欠点は、オペレーティングシステムへの依存性が高く、そして、スーパーユーザー、管理者、または完全なアクセス権と共に稼働しているプロセスにより、ローカル記憶装置上で重要なファイルが誤って変更されるか、削除されるか、または破損されることである。

10

【発明の開示】

【発明が解決しようとする課題】

【0014】

上記で述べた様々な手段は、単独であるいは他の手段と組み合わせることで、コンピュータの管理に伴う複雑さを減らすことに多少役に立つ。しかし、いずれの手段も、オペレーティングシステムがコンピュータの構成要素を管理する方法を根本的に変更するものではない。したがって、コンピュータを管理し、ネットワークを介してファイルを分配する手段を簡素化する新しいアプローチが要望されている。

20

【課題を解決するための手段】

【0015】

本発明は、読み取り専用記憶装置、その装置を結合するシステム、およびコンピュータネットワークを介してファイルを分配する方法を提供する。この記憶装置は、コンピュータインターフェースと通信できるように接続されたデバイスプロセッサ、ネットワークインターフェース、そして、データ格納装置を含むものである。記憶装置がコンピュータとファイルサーバに接続されている際、コンピュータインターフェースとネットワークインターフェースは、前記デバイスプロセッサとコンピュータ、そして前記デバイスプロセッサとファイルサーバとの通信を可能にする。コンピュータは、デバイスプロセッサからファイルを要求することができ、ファイルの要求があると、デバイスプロセッサは、まず、要求されたファイルがデータ格納装置に保持されているか確認する。もし、要求されたファイルがデータ格納装置に保持されていた場合、そのファイルは、読み込み専用としてコンピュータに供給される。

30

【0016】

要求されたファイルがデータ格納装置に保持されていない場合、前記デバイスプロセッサは、ネットワーク上のファイルサーバから要求されたファイルを手に入れることが可能である。要求されたファイルがファイルサーバから利用可能であるならば、デバイスプロセッサは、ファイルを手に入れて、データ格納装置にそのファイルを保持して、読み取り専用としてコンピュータに供給する。ファイルがデータ格納装置に一旦保持されると、そのファイルは、その後コンピュータに供給可能になる。もし、ファイルがファイルサーバから取得不可能な場合、デバイスプロセッサはその旨をコンピュータに送り返す。

40

【0017】

また、記憶装置は、データ格納装置に保持した可能性があるファイルを削除せよとの命令を、ファイルサーバから受信することもありうる。削除命令を受けると、デバイスプロセッサは、データ格納装置にファイルが保持されているか確認して、保持されている場合そのファイルを削除する。

【0018】

従って、本発明は、改良された読み取り専用記憶装置と、またこの記憶装置を含むシステ

50

ム全体と、コンピュータネットワークを介してファイルを分配する方法を開示するものである。本発明の特徴と利点は、添付の図面および後述本発明の好ましい実施形態の説明により明らかになるであろう。

【図面の簡単な説明】

【0019】

【図1】従来技術に関するネットワークコンピュータのブロック図である。

【図2】本発明の好ましい実施形態に従った読み取り専用記憶装置のブロック図である。

【図3】ネットワークコンピュータに結合された図2の記憶装置のブロックである図。

【図4】図3のネットワークコンピュータからのファイル要求に応じて記憶装置が使用する演算方式を示すフローチャートである。

10

【図5】ネットワークコンピュータに結合された図2の記憶装置のブロック図である。

【図6】本発明の第1の代替実施形態に従った読み取り専用記憶装置のブロック図である。

。

【図7】ネットワークコンピュータに結合された図6の記憶装置を示すブロック図である。

。

【図8】本発明の第2の代替実施形態に従った読み取り専用記憶装置のブロック図である。

。

【図9】ネットワークコンピュータに結合された図8の記憶装置を示すブロック図である。

。

【発明を実施するための形態】

20

【0020】

図面の細部に目を向けると、図1は、従来の技術に見られる典型的なネットワークコンピュータ101を示している。この図に示されているように、ネットワークコンピュータ101は、市販品、特注品、個人用、またはビジネス用のコンピュータであり、また、個人用のデジタルアシスタント、電話、あるいはネットワーク化された電気製品のような専用装置でもある。図1を見ると、コンピュータ101は次のような様々な構成部品と共に示されている。例えば、中央演算処理装置(CPU)104、および様々な構成部品間を連結し通信を可能にする内部インターフェースバス105。また、ファイル・プリントの共有サーバ115および/またはネットワークアプリケーションサーバ116のようなネットワーク資産を有するネットワーク114に、コンピュータ101に有線か無線で連結するネットワークインターフェース107。その他、ハードドライブ、非揮発性ランダムアクセスメモリのような、通常、オペレーティングシステム103やアプリケーションファイル102が保存されている補助記憶装置108といった、一般的な構成部品と共に示されている。これらの構成部品は、ネットワーク化されたコンピュータに一般的であり、広く知られている。

30

【0021】

オペレーティングシステム103は、機能的なコンピュータを形成すべく、構成部品間のやりとりを管理する。そして、オペレーティングシステム103は、アプリケーション102またはオペレーティングシステム103自身からのファイル要求を処理する。このようなファイル要求は、補助記憶装置108、ファイル・プリントの共有サーバ115、またはネットワークアプリケーションサーバ116のいずれかに存在するファイルに、アプリケーション102またはオペレーティングシステム103がアクセスを要求するとき

に生じる。このファイル要求を処理するために、オペレーティングシステム103は、コンピュータと共用されうる広範で多様な記憶装置とネットワークプロトコルを扱うために、記憶装置108とネットワークインターフェース107の様々な特性をもとにプログラムされなければならない。さらに、このオペレーティングシステム103は、補助記憶装置108に保存されたファイルを整理するのに必要な情報と、ネットワークサーバ115および116から利用できる多様な資産をコンピュータの動作に統合する方法についての情報をもとにプログラムされなければならない。従って、一般的な動作環境で、オペレーティングシステム103は、多くの異なった複雑なタスクを行わなければならない。

40

50

【0022】

図2は、本発明の好ましい実施形態に従った、キャッシュベースの記憶装置(CBD)202のブロック図である。CBDは、様々なコンピュータの構成部品や、複数のネットワークサーバとのやりとりを管理するときにオペレーティングシステムに要求されるタスクを簡素化するために、図1で示されたようなコンピュータと共に使用されることを目的とする。CBD202は、次のような4つの基本的な要素から成り立つ。1) CBD202をコンピュータに接続するメインコンピュータインターフェース203、2) CBD202をネットワークに接続するネットワークインターフェース207、3) CBD202内の構成要素を管理し、コンピュータのオペレーティングシステムからのファイル要求を処理するキャッシュエンジン(またはデバイスプロセッサ)206、4) キャッシュエンジン206がローカルにファイルを保存するのに使用するキャッシュ(またはデータ格納装置)204である。

10

【0023】

キャッシュエンジン206は、本明細書に説明されているようにCBD202の機能を制御するのに適切であれば、どのような種類の中央演算処理装置であってもよい。設計により、どのような種類の中央演算処理装置を使用するか決定する。キャッシュエンジン206用の中央演算処理装置は、ARM、ヒューレット・パカード、IBM、インテル、アドバンストマイクロデバイセスなどのメーカーの集積回路部門から入手できる。

【0024】

キャッシュエンジン206は、キャッシュ204に保存されたファイルを管理し、そして必要に応じてコンピュータとネットワークと通信できるようプログラムされている。キャッシュエンジン206は、C++、Java(登録商標)等のようなプログラミング言語でもプログラム可能である。そして、コンパイルされたプログラムは、キャッシュエンジン206自体に組み込むか、または、プログラム組み込み専用のメモリに保存することも可能。別の方法としては、キャッシュエンジン206がアクセスできるよう、プログラミングをキャッシュ204の中に保存することも可能である。プログラミングにより、コンピュータはキャッシュエンジン206と通信でき、読み込みを原則としてキャッシュ204で保存されたファイルへのアクセスが可能になる。また、プログラミングにより、キャッシュエンジン206は、ネットワークインターフェース207を介してネットワークと通信できる。従って、コンピュータが、キャッシュ204に保存されていないファイルをCBD202から要求するとき、キャッシュエンジン206は、CBD202が接続されたネットワーク上のサーバにそのファイルを要求することも可能である。要求されたファイルがネットワーク上で見つけられると、キャッシュエンジン206は、キャッシュ204にファイルを保存し、読み込みを原則として、コンピュータがそのファイルにアクセスできるようにする。要求されたファイルがキャッシュ204上でも、ネットワーク上でも利用可能でない場合、キャッシュエンジン206は、要求されたファイルが利用不可能であるとの通知をコンピュータに供給しうる。

20

30

【0025】

コンピュータインターフェース203とネットワークインターフェース207は、キャッシュエンジン206が、コンピュータとネットワークのそれぞれに通信が可能なインターフェースであれば、どのような種類であってもよい。コンピュータとネットワークとをCBD202に相互接続する特定のインターフェースの種類を選択は、設計に委ねられる。コンピュータインターフェース203の種類として、例えばIDE、ATA、SCSI、IEEE1394等、一般的に知られている形式が適している。ネットワークインターフェース207のネットワークインターフェースの種類として、例えばEthernet(登録商標)、Token Ring、ATM等、その他一般的に知られている形式が、そしてネットワークプロトコルの種類としてはTCP/IP、IPX、AppleTalk、SNA等、その他一般的に知られている形式が利用可能である。以上に挙げたIDEやTCP/IPなどの構成要素は、さまざまなコンピュータハードウェアメーカーから入手することができる。

40

【0026】

50

従って、記憶装置であるC B Dは、特定のプロトコルまたはファイルの構造に適合する必要がないので、C B Dは、オペレーティングシステムから独立した装置である。C B Dに保存されたファイルの管理は、オペレーティングシステムではなくデバイスプロセッサであるキャッシュエンジンが行うので、オペレーティングシステムのプログラムはC B Dに保存されたファイルを要求するのに必要とされるプロトコルだけである。従って、オペレーティングシステムとC B Dとの通信は少ない手順で実行される。これらの手順のみが、オペレーティングシステムとシステム構造設計に依存するものであり、どのようなハードウェアプラットフォーム上でも、これらの手順は標準化が可能である。

【0027】

また、データ格納装置であるキャッシュ204に使用される特定の記憶形式の選択は設計に委ねられる。キャッシュ204には、当業者によく知られている媒体として、例えばハードドライブ、DRAM、SDRAM、フラッシュメモリ、あらゆる形式の取り外し可能で読み書き自由な媒体を使用できる。キャッシュエンジン206に使用されるプログラミングは、キャッシュ204用に選ばれた特定の形式の読み書き自由な媒体の機能に適合しなければならない。

【0028】

図3は、本発明の最適実施形態に従ってC B D 202が組み込まれ、ネットワークに接続されたコンピュータ301を示す。この実施形態では、C B D 202は、コンピュータ301の内部に組み込まれている。しかし、また、本明細書で説明された機能性のいずれをも失わずに、C B D 202は、コンピュータ301の外部に配置されることも可能である。従来技術において、コンピュータは、中央演算処理装置(CPU)304、インターフェースバス305、補助記憶装置308およびネットワークインターフェース307を備えている。オペレーティングシステム303は、様々な構成部品の動作を統合し、アプリケーション302は、オペレーティングシステム303により作り出された環境の中で実行される。コンピュータ301は、ネットワークインターフェース307を介してファイル・プリント共有サーバ315およびアプリケーションサーバ316に接続されている。C B D 202は、コンピュータインターフェース(図3に示されず)をインターフェースバス305に接続することによって、コンピュータと通信できるように接続されている。

【0029】

図3のネットワークでつながれたコンピュータ301は、2つの別々のネットワークに接続されている。第1のネットワーク309は、C B D 202のみに利用されるC B Dファイルサーバ314に、C B D 202を直接接続する。第2のネットワーク311は、コンピュータのネットワークインターフェース307をファイル・プリント共有サーバ315とネットワークアプリケーションサーバ316とに接続する。これらのネットワークは、銅線、光ファイバー、ワイヤレス通信、または当業者に知られている他の種類のネットワークで形成される。さらに、第1のネットワーク309と、第2のネットワーク311が同じ形式である必要はない。例えば、第1のネットワーク309がワイヤレスで、第2のネットワークでも構わない。

【0030】

好ましくは、上記2つのネットワークが相互作用しない独立したネットワークであること。しかしながら、2つのネットワークは、図5で示されるように、ネットワーク集信装置513を通して、C B D 202およびネットワークインターフェース307に共有される単一のネットワーク接続からなる、単一のネットワークに併合されてもよい。ネットワーク集信装置513は、ネットワーク情報をC B D 202およびネットワークインターフェース307に伝達するように、そしてネットワーク509と通信できるように接続されている。このような環境下では、C B D 202をネットワーク上でC B Dファイルサーバ314と交信する唯一の装置とすることは、必須ではないが好ましいことである。言い換えれば、C B Dファイルサーバ314は、C B D 202以外のいかなる装置とも交信せず、そして、C B D 202は、C B Dファイルサーバ314以外のいかなるサーバとも交信しないということである。

10

20

30

40

50

【0031】

図6で示された本発明の第1の代替実施形態において、ネットワーク集信装置608は、上記の513と同様の集信装置であり、CBD602に組み入れられている。このCBD602は、図7で示されているようにコンピュータ701内に配置されている。図7のコンピュータ701は、図3のコンピュータ301と同じ構成部品を備えている。しかし、コンピュータ301が、外部に配置された集信装置513を通してネットワークに接続されるのと違い、コンピュータ701は、CBD602を通してネットワーク509に接続されている。CBDファイルサーバ314、ファイル・プリント共有サーバ315およびネットワークアプリケーションサーバ316は、全て単一のネットワーク509に接続されている。CBD602は、ネットワーク集信装置608を介してネットワーク509に接続されている。ネットワーク集信装置608は、ネットワークインターフェース307に接続されており、コンピュータ701とネットワーク通信を行う。この接続は、図示されているようにコンピュータの外部で形成されても、あるいはコンピュータの内部で形成されてもよい。

10

【0032】

図8では本発明の第2の代替実施形態が示されている。この実施形態においては、CBD802により、ネットワーク集信装置を使用することなく、コンピュータがネットワークに接続される。CBD802のネットワークインターフェース807は、コンピュータに向けて発信されたネットワーク通信を、直接コンピュータインターフェース803に伝達する。したがって、図9に示すように、CBD802がコンピュータ901の内部に組み入れられているとき、ネットワーク通信は、CBD802を介して、コンピュータ901内部のインターフェースバス305に伝達される。この構成では、CBDファイルサーバ314、ファイル・プリント共有サーバ315およびネットワークアプリケーションサーバ316は、全て、CBD802とコンピュータ901の両方にネットワーク通信を供給する単一のネットワーク509に接続されている。加えて、CBD802とCBDファイルサーバ614間の通信のために、現在普及している暗号化技術を導入することによって、上記の構成を図3が示すような好ましい実施形態にすることが可能である。

20

【0033】

図4は、デバイスプロセッサであるキャッシュエンジンが、オペレーティングシステムからのファイル要求に回答する際に従う際、好ましい論理を示すフローチャートである。キャッシュエンジンがオペレーティングシステムからファイル要求402を受け取ると、その要求がファイルの読み込み403であるのか、それとも、ファイルの書き込みあるいはファイルの消去の要求であるのかをチェックする。もし、ファイル要求がファイルの読み込みでない場合、キャッシュエンジンは、「要求無効」の通知をオペレーティングシステムに返信することにより、その要求404を拒絶する。もし、その要求がファイルの読み込みの際にキャッシュエンジンは、そのファイルがキャッシュから読み取り可能かどうかを知るために、データ格納装置であるキャッシュ406をチェックする。もしファイルがキャッシュ406から読み取り可能な場合、キャッシュエンジンはファイル407をオペレーティングシステムに供給する。もし要求されたファイルがキャッシュから利用できない場合、キャッシュエンジンはファイル要求408をCBDファイルサーバに送信する。このようなファイル要求は、ネットワーク上のあらゆるCBDファイルサーバに送信され、ネットワーク上の特定のCBDファイルサーバに送信されてもよい。CBDが接続されている特定のネットワークにより、ファイル要求が送信される手段が決まる。もしファイルがCBDファイルサーバ409から利用可能であるならば、キャッシュエンジンは、CBDサーバからローカルキャッシュにファイル410をコピーし、そのコピーしたファイル407をオペレーティングシステムに供給する。もし、ファイルがCBDファイルサーバから読み取り不可能な場合、キャッシュエンジンは、「ファイル読み取り不可能」の通知をオペレーティングシステム411に返信する。

30

40

【0034】

また、オペレーティングシステムの要求に応じて、前記CBDサーバからファイルを読

50

み取ることに加えて、加えて、前記C B Dは、必要に応じて、もしくはC B Dファイルサーバから指示があった場合、ファイルを読み取り保持するように予めプログラムされることもできる。C B D上に保存されたファイルは、一つ以上の基準に基づいて削除されてもよく、その結果、そのファイルが必要とされる次の段階で、C B Dは、C B Dサーバからファイルを再読み取りしなければならない。ファイル削除は、C B Dサーバの指示によって強制的に行っても、また、特定のファイルがファイル有効期限タグを備えるようにしてもよい。従って、C B Dは、ファイル有効期限タグによって指示されている日時に、そのファイルを削除することになる。ファイル有効期限は、指定した日時、あるいは予想された出来事に基づく日時に設定されることができる。

【0035】

次に、本発明のいくつかの利点の実施例を示す。

【0036】

1. パーソナルコンピュータ上へのオペレーティングシステムのインストール

【0037】

今日のパーソナルコンピュータ用のオペレーティングシステムは、多くの機能を提供することから、しばしば、オペレーティングシステムを、ローカルハードドライブにインストールすることが必要である。起動プロセスにおいて、ROMに基づいたBIOSは、ハードドライブに保存されているブートストラップを最初に読み込むことにより、オペレーティングシステムをロードする。ブートストラップは、オペレーティングシステムをロードするのに必要とされる、最初に実行可能なコードについての情報を提供する。ブートストラップから提供された情報に基づいて、BIOSは、その最初のファイルをメモリーに取り込んで、ファイルにコントロールを移管する。その結果、このファイルは、全体のオペレーティングシステムをロードするのに必要な他のファイルの取り込みを開始する。

【0038】

従って、C B Dを備えたコンピュータにおいて、オペレーティングシステムは、起動プロセスの前にローカルハードドライブにインストールされる必要がない。前記BIOSは、オペレーティングシステムがロードを開始するのに必要な最初のファイルを、C B Dから要求してもよい。新しいコンピュータにおいては、C B D上のキャッシュは、この段階ではおそらく空である。デバイスプロセッサであるキャッシュエンジンが、コンピュータから最初のファイルの要求を受け取ると、その最初のファイルはC B Dファイルサーバから読み取られる。次に、そのファイルはキャッシュに保存され、BIOSに提供される。そして、そのファイルはBIOSからの起動プロセスを制御する。ファイル要求はC B Dへ継続して送られ、要求されたファイルは、読み取られ、キャッシュに保存され、ファイル要求者に提供されることができる。従って、初めにオペレーティングシステムをロードするのに必要である全てのファイルは、C B DによってC B Dファイルサーバから入手されうる。

【0039】

前記のプロセスは、BOOTP、TFTPまたはPXEなどの周知のプロセスによって、ネットワークからコンピュータを起動することと幾分似かよっているが、C B Dを利用する主な利点は、コンピュータが再起動されるとき明らかになる。例えばBOOTPを利用してコンピュータを再起動する場合、ネットワークからファイルをダウンロードするプロセスを繰り返すことが必要となる。一方、C B Dを備えたコンピュータを起動する場合、全ての必要なファイルは、C B D内のキャッシュに保存され、そしてそのキャッシュから供給されるので、再びネットワークから全てのファイルをダウンロードすることは必要ではない。また、前記のプロセスは、ローカルハードドライブ上にオペレーティングシステムを保存することと異なる。なぜならその場合、オペレーティングシステムがハードドライブ上のファイルを管理しなければならないからである。一方、C B Dを備えたコンピュータ上で作動するオペレーティングシステムは、C B Dのキャッシュ内のファイルを管理しない。

【0040】

また、前記のプロセスは、例えば、CD-ROMなどのローカルに装着された読み取り専用装置からコンピュータを起動することと幾分似通っている。しかし、C B Dを使用する利点

10

20

30

40

50

は、C B Dの柔軟性によって明らかとなる。ファイルを交換する必要があるとき、あるいはアップデートする必要があるとき、CD-ROMを新しいものに取り替えるか、または米国特許番号5832263および5790848に開示されたものと同様の技術を使用しなければならない。一方、C B Dによって保持されたファイルは、管理者によって動的に取り替えられるか、あるいはアップデートされる。管理者は、特定のファイルを削除するために、単に「削除」コマンドをC B Dへ送る操作をC B Dファイルサーバに対して行えばよい。ファイルを削除した後で、次回そのファイルが要求されるか必要になった時には、最新版のファイルがC B Dファイルサーバから読み取られる必要がある。その結果、C B Dは、最新のファイルをキャッシュ内にいつでも保有していることが可能となる。

【0041】

2. パーソナルコンピュータへのアプリケーションのインストール

【0042】

C B Dを備えたコンピュータ上でアプリケーションをインストールするプロセスは、前記の、オペレーティングシステムをインストールするプロセスと類似している。オペレーティングシステムは、ユーザーに利用可能なアプリケーションのリストを提示する。ユーザーがアプリケーションを選択すると、オペレーティングシステムは必要なファイルをC B Dから要求する。アプリケーションが以前使用されたことがない場合、C B Dは、C B Dファイルサーバから必要なファイルを読み取り、キャッシュ内にそれらのファイルを保存する。そしてオペレーティングシステムがアプリケーションをロードできるように、そのファイルをオペレーティングシステムに供給する。

【0043】

上記の方法に従って初めてアプリケーションにアクセスする速度は、サーバベースのアプリケーション(例えば、ネットワーク上の複数のユーザーが使用できるようネットワークサーバにインストールされたマイクロソフトワード)を起ち上げる速度に匹敵しているといえる。C B Dを使用する利点は、2回目以降にアプリケーションにアクセスするとき明らかとなる。アクセスする度に、ネットワークサーバから取り込まなければならないサーバベースのアプリケーションと違って、C B Dを使用する場合、アプリケーションはC B Dに保持されている。従って、2回目以降に必要な度に、ファイルに素早くアクセスすることが可能である。さらに、アプリケーションの構成要素またはアプリケーションそれ自体は、管理者が「削除」コマンドをC B Dに送る操作をC B Dファイルサーバに行わせることにより、動的に取り替えられるか、あるいはアップデートされることができる。

【0044】

3. オペレーティングシステムとアプリケーションに重要なファイルの保護

【0045】

従来コンピュータにおいて、ローカルハードドライブに保存されたファイルは、通常、偶然の削除、変更、あるいはユーザーエラーやコンピュータウイルスによって引き起こされる破損の対象になる。削除、変更、または破損されたファイルは、アプリケーションエラー、オペレーティングシステムエラー、またはコンピュータ自体の誤作動につながる可能性がある。したがって、オペレーティングシステムとアプリケーションにとって重要なファイルは、最高水準の保護を受けるべきである。

【0046】

ファイルを保護するための従来方法は、数多く存在する。会社環境において、管理者は、自分が完全なアクセス権(読み込み、書き込み、変更、追加、削除など)を持つ唯一のユーザーとなる一方で、一般ユーザーが制限されたアクセス権(読み込み、実行など)を与えるようにコンピュータをセットアップすることで、ファイルレベルのセキュリティを適用することができる。一旦このようなファイルレベルのセキュリティが適用されると、通常それを維持することは、オペレーティングシステムの重要なタスクとなる。オペレーティングシステムは、ファイルとセキュリティを管理するための複雑なリストあるいはデータベースを管理しなければならない。オペレーティングシステムは、破損したかもしれないファイルを自動的に修復しようとする自己修復機能を備えることもできる。加えて、

10

20

30

40

50

ウイルスからパーソナルコンピュータを保護するためにウイルス対策ソフトウェアがインストールされる可能性がある。しかし、もしファイルレベルのセキュリティシステムが何らかの手段で侵害されると、ハードディスク上のファイルは削除、変更、あるいは破損といったリスクにさらされる。さらに、ウイルス対策ソフトウェアが最新のものでないと、そのソフトウェアは、ウイルスに対する適切な防御を提供できない事態に陥ることもある。

【 0 0 4 7 】

対照的に、C B Dを備えたコンピュータは、人為ミスやウイルスに対して、重要なファイルを完全に保護できる。コンピュータは、前記C B Dに保持されたファイルに対して読み込み専用アクセス権だけを与られているので、ユーザーやコンピュータからのウイルスによってそれらのファイルが削除されたり、変更されたりまたは破損されたりする可能性が無い。さらにC B Dは、C B Dファイルサーバからファイルまたはアップデートを受け入れるだけなので、管理者もオペレーティングシステムもC B Dのキャッシュ内に保存されたファイルを変更することはできない。したがって、C B Dは、必然的にオペレーティングシステムやアプリケーションにとって重要なファイルを保護することができる。

10

【 0 0 4 8 】

さらに、オペレーティングシステムは、上記C B D以外のどのような装置からのいかなるプログラムも、あるいは実行可能なコードもロードしないように構成できるため、コンピュータウイルス、ワームまたはスクリプトなどのような、権限のないプログラムもしくはコードに対して、オペレーティングシステムやアプリケーションにとって重要なファイルに別の保護が追加される。この付加的な保護は、上記C B Dを利用したシステムに固有のものである。なぜなら、C B Dを通してコンピュータに提供される全てのファイルは、C B Dファイルサーバ上で使用するに安全であると確認されてからコンピュータに提供されるからである。

20

【 0 0 4 9 】

4. 簡易なソフトウェアライセンス追跡と管理

【 0 0 5 0 】

会社環境において、ソフトウェアライセンスとその使用状況を管理することはきわめて困難である。ユーザーは、合法的ライセンス付きのアプリケーションがインストールされたコンピュータを最初に与えられる。しかし、ユーザーは、自分で他のアプリケーションをインストールし始めるかもしれないし、また、ライセンス料が支払われたインストール済みのアプリケーションのいくつかを一度も使用しないかもしれない。したがって、どのソフトウェアがユーザーによって使用されているか、どのソフトウェアがライセンス付きなのか、どのソフトウェアがライセンス付でないのかを管理するのは、難しい作業となる。

30

【 0 0 5 1 】

C B Dを備えたコンピュータは、上記のようなソフトウェアとソフトウェアライセンスを管理する作業の煩雑さを低減するために、会社全体に導入される。まず第1に、C B Dは読み込み専用装置であるので、ユーザーはアプリケーションをC B Dにインストールすることができない。従って、コンピュータのローカルハードドライブにインストールされたソフトウェアは、ユーザーがインストールしたソフトウェアであることが容易に確認される。第2に、C B Dを備えたコンピュータは、アプリケーションをC B Dファイルサーバからダウンロードするため、管理者は、C B Dファイルサーバからアプリケーションがダウンロードされた回数を監視することによって、使用されているアプリケーションの正確な数を容易に把握することができる。管理者は、C B Dからアプリケーションの主要な構成要素を削除する「削除」コマンドを発行し、そして、次にアプリケーションが使用される時に、その構成要素がC B Dファイルサーバから要求されるようにすることで、いつでも使用中のアプリケーションの正確な数を把握することができる。第3に、C B Dが使用される場合、アプリケーションの使用を監視しやすいために、アプリケーション販売業者は、使用ごとにライセンスを付与するなどといった以前に無かったライセンス付与のモデ

40

50

ルを導入できる。アプリケーション販売業者は、インターネットを通してC B Dファイルサーバにアクセスすべく構成されたC B Dから、さらに大きな利益を得ることも可能になる。

【 0 0 5 2 】

5 . データベースアプリケーション

【 0 0 5 3 】

データベースアプリケーションにとって、C B Dの使用は有効であるといえる。アドレス帳アプリケーションまたは地図アプリケーションは、データベースにまとめられた記録の集合体を利用するアプリケーションの例である。このようなアプリケーションには、比較的変更が少ないものの、周期的なアップデートを必要とすることがある記録がしばしば含まれている。

10

【 0 0 5 4 】

データベースアプリケーションと共にC B Dを利用するために、ルックアップテーブルを含むファイルをC B Dに含めることができる。この場合、ルックアップテーブル上の記入事項が、データベース上の個々の記録を示している。初めに、データベースアプリケーションは、C B Dから、または必要な場合C B Dファイルサーバから、ルックアップテーブルのファイル入手する。ユーザーが必要としているルックアップテーブルファイルからの各記録は、C B Dファイルサーバから読み取られ、ユーザーがアクセスできるようC B Dに保持されることができる。従って、データベースアプリケーションが使用された後、キャッシュはユーザーがアクセスした全ての記録を含むことになる。ユーザーによって保持されたいずれかの記録がアップデートされる必要がある場合、C B Dファイルサーバは「削除」命令を発行することができ、その結果、キャッシュから特定の記録が削除される。また、C B Dファイルサーバは、適切な「削除」コマンドを発行することによって、ルックアップテーブルファイル全体をアップデートすることもできる。記録またはルックアップテーブルのファイルが削除された後でユーザーがデータベースにアクセスする時に、C B Dは、必ずアップデートされた記録またはファイルをC B Dファイルサーバから読み取らなければならない。従って、ユーザーは、データベースアプリケーションを使用する際、常に最新の記録にアクセスすることになる。

20

【 0 0 5 5 】

携帯電話などのワイヤレス機器において、上記に述べたデータベース管理方法を有効的に導入することが可能。携帯電話は、しばしば多くの連絡先リストを保存しているが、その連絡先リスト上の多くの記入事項が全く使用されないことがある。しかし、これらの多くの記入事項のために、携帯電話のデータベースを会社のサーバのマスターデータベースと同期させるのに時間がかかることもある。キャッシュとしてフラッシュメモリを装備しているC B Dを備えている携帯電話は、携帯電話上で最新のデータベースを維持するために必要な時間を短縮することが可能である。まず、携帯電話のC B Dは、連絡先リストの記録のルックアップテーブルファイルをダウンロードする。ユーザーが、アクセスしようとしてルックアップテーブルの特定の記入事項を選択すると、C B Dは、C B Dファイルサーバにワイヤレス接続を開始して、キャッシュにその記入事項を読み取ることで、必要な情報をユーザに提供する。この様にして読み取られた全ての記入事項は、ユーザーが後でC B Dファイルサーバにアクセスせずに記入事項にアクセスできるよう、C B D内に保持される。さらに、C B Dファイルサーバは、C B Dとのワイヤレス接続を開始して、ルックアップテーブルのファイルまたは特定の記録を消すために「削除」命令を送信することも可能である。その結果、携帯電話は、ユーザーが定期的に必要とする記録をC B Dに持っているだけで、記録のアップデートは、ユーザーにとって煩わしい事でなくなる。

30

40

【 0 0 5 6 】

6 . インターネットサービスプロバイダーまたはアプリケーションサービスプロバイダーのためのC B D

【 0 0 5 7 】

インターネットサービスプロバイダーまたはアプリケーションサービスプロバイダーは

50

、サービスを提供するためにC B Dをサービス加入者に提供することができる。C B Dは、インターネットを介して、ソフトウェアと他のサービスとを顧客へ提供するために、プロバイダーのC B Dファイルサーバにアクセスするように構成されることができる。

【0058】

例えば、サービスプロバイダーは、C B Dを備えたパーソナルコンピュータを顧客に提供することがとできる。この場合、C B Dは、インターネット上でサービスプロバイダーのC B Dファイルサーバを使用するように、あらかじめ設定される。サービスプロバイダーは、しばしば、インターネット接続がLAN接続よりはるかに遅くなるので、初期のダウンロード時間を短くするために、オペレーティングシステムといくつかの基本的なアプリケーションを、C B Dのキャッシュに前もってインストールすることも可能である。

10

【0059】

コンピュータがインターネットに接続されている場合、C B Dはファイルをキャッシュにダウンロードしたり、必要に応じて「削除」コマンドを受け取り、全てインターネットを介して通常通り作動する。もし、ユーザーが別のアプリケーションかオペレーティングシステムを必要とする場合、その要求がサービスプロバイダーに届けられ、そして、本明細書に説明された方法によって、ユーザーは新しいアプリケーションまたはオペレーティングシステムにアクセスすることができるようになる。このようにC B Dを応用することで、ユーザーはC B Dに対して読み取り専用のアクセスしかえられないので、サービスプロバイダーは、C B Dの内容に対し完全な掌握が可能になる。従って、サービスプロバイダーは、大きな障害なくユーザーのシステムを管理し、アップデートでき、ユーザーのコンピュータに広告を配置することで、付加的利益を作り出す事が可能になる。また、コンピュータの販売御者は、コンピュータ上に広告の配置を許容することと引き換えに、コンピュータの価格を下げることもできる。このようにコンピュータ購入の費用が下がれば、ユーザーの利益にもなる。

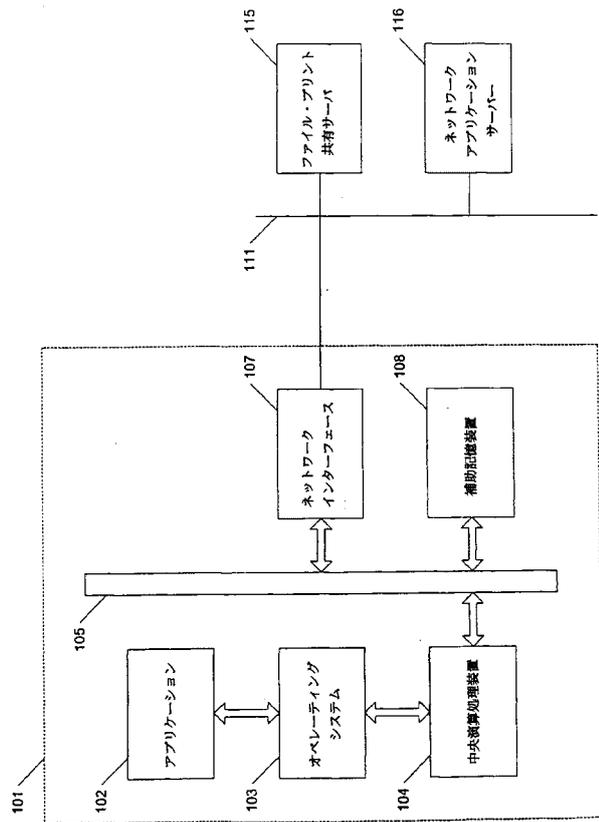
20

【0060】

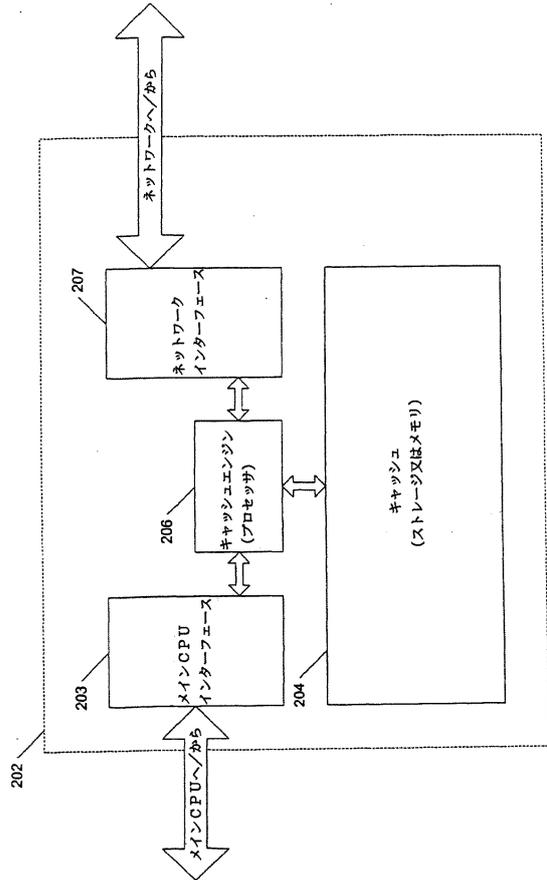
本発明は、ネットワークインターフェースを備えた読み取り専用記憶装置、その装置を含むシステム、およびネットワーク上にファイルを分配する方法を開示している。本発明の実施形態が図示され説明されたが、本明細書に説明された発明概念の範囲から外れない限り、より多くの設計変更が可能であることは当業者にとって明らかである。従って、次に挙げる請求項の意図する範囲である限り、本発明はなんら限定されるものではない。

30

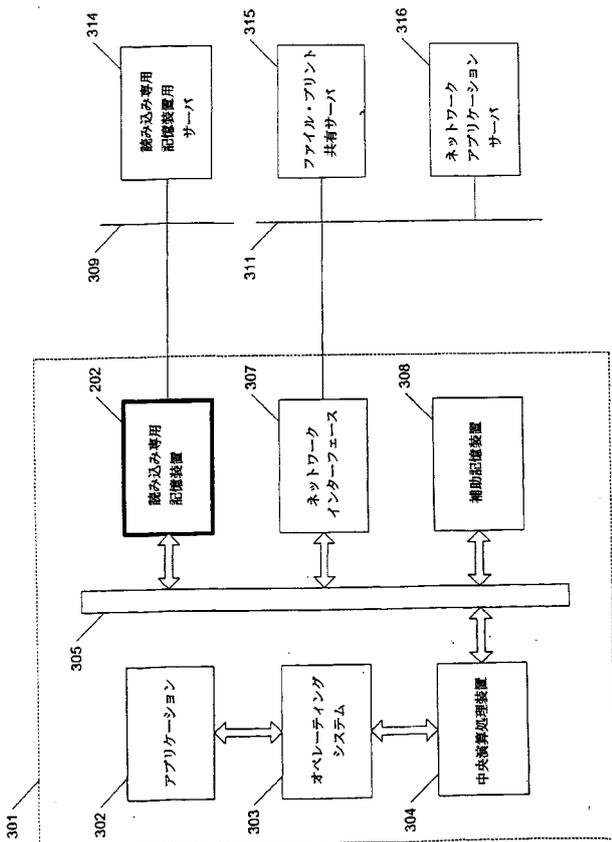
【図 1】



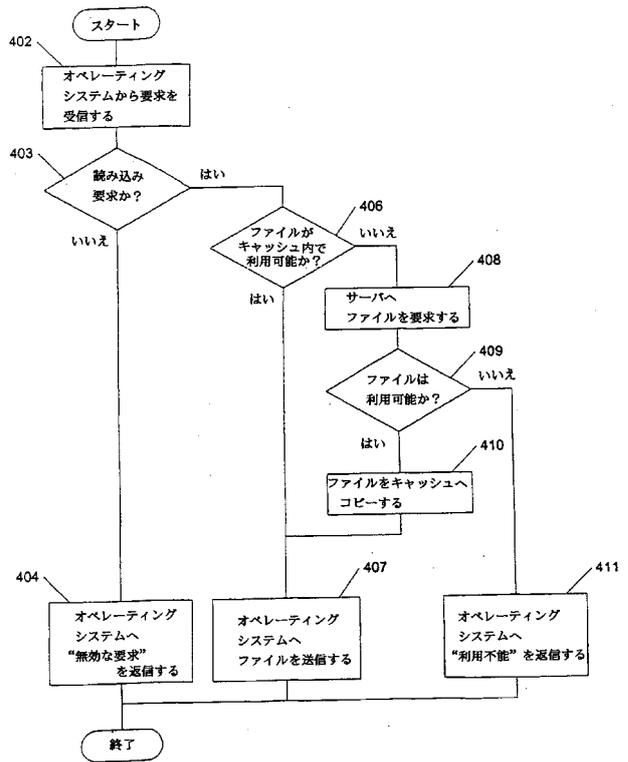
【図 2】



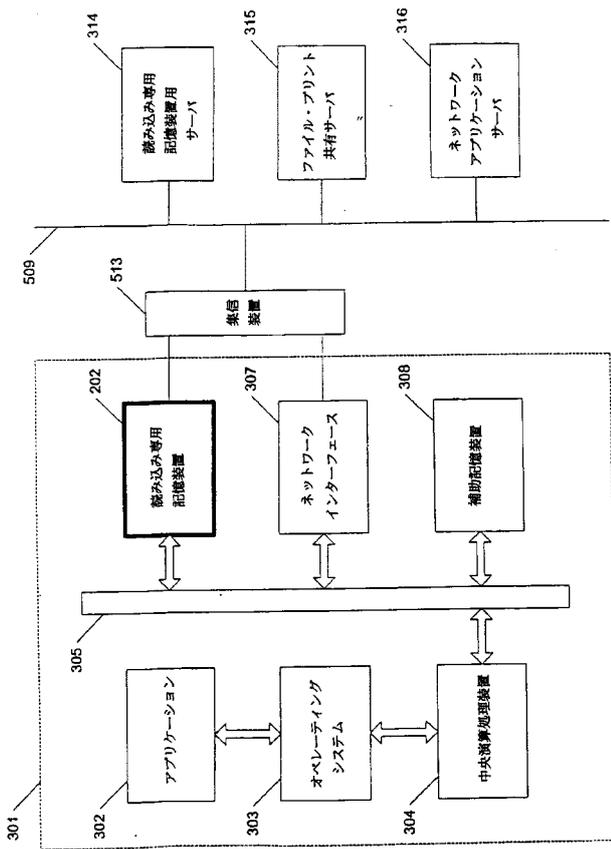
【図 3】



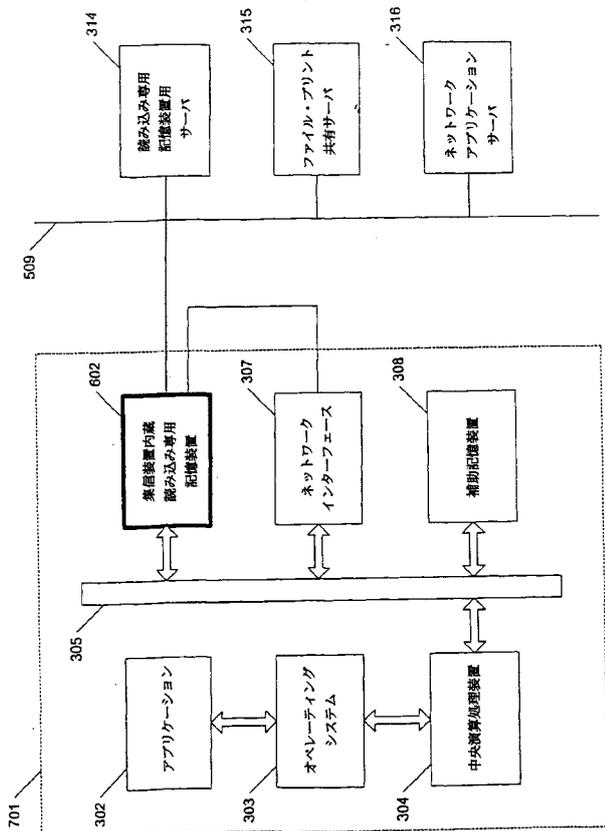
【図 4】



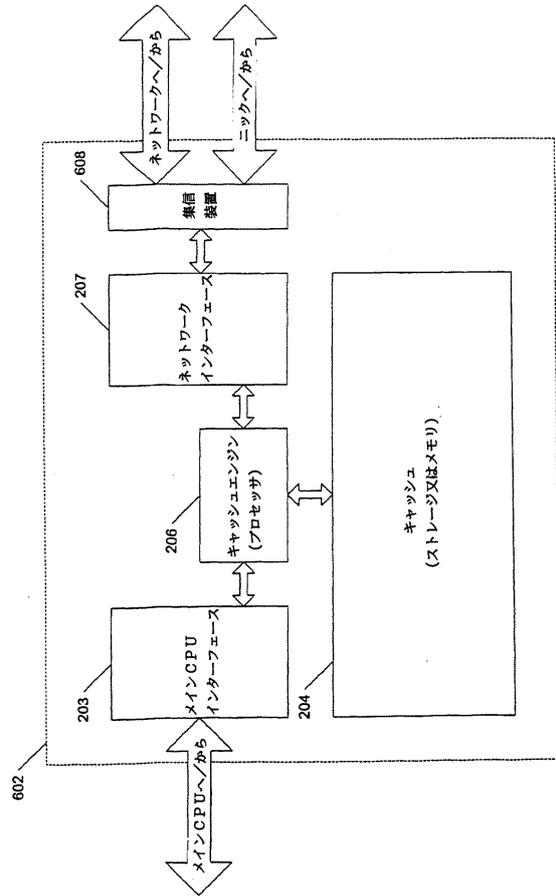
【図5】



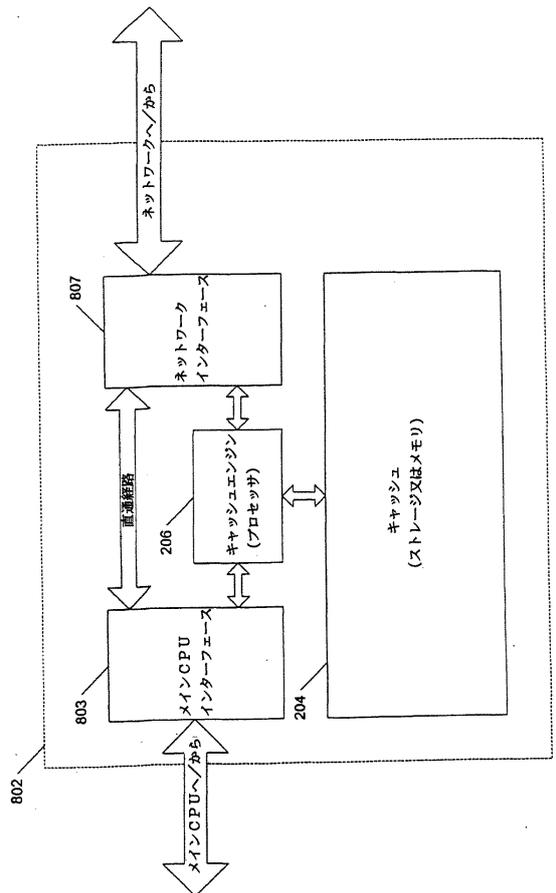
【図7】



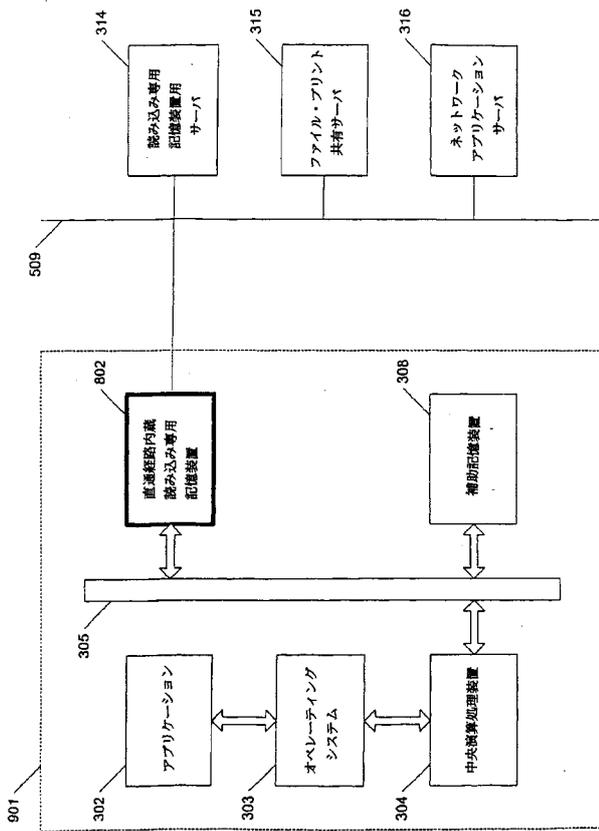
【図6】



【図8】



【図 9】



フロントページの続き

(72)発明者 ケイシー ケイ チャン

アメリカ合衆国 カリフォルニア 90503 トランス ピー . オー . ボックス 13906

Fターム(参考) 5B065 CC10 ZA15

5B077 DD01

5B082 HA02 HA05 HA08

【 外国語明細書 】

WO 03/038643

PCT/US02/31991

S P E C I F I C A T I O N

LOCALIZED READ-ONLY STORAGE DEVICE FOR DISTRIBUTING FILES OVER
A NETWORKBACKGROUND OF THE INVENTION

5 1. Field of the Invention

[0001] The field of the present invention is computing devices for use in combination with a network and methods of using such devices.

2. Background

10 [0002] A typical computer, whether an off-the-shelf or a customized computer for personal, business, specialty, or other use, has many components. Some of the common components are processors, memories, storage devices, input and output devices, and network interfaces. The operating system, such as Microsoft Windows®, Mac OS®, Unix, Linux, etc., is responsible for controlling the components and their functions. In particular, the operating system handles file requests from
15 applications or from the operating system itself. When a file request is received, the operating system attempts to supply the file from one of the local storage devices or from a network server if the computer is connected to a network.

[0003] Because there are many different options available for storage devices and network interfaces, the operating system is typically programmed with the
20 characteristics necessary to access almost every storage device and network interface that could be connected to the computer. Thus, for each storage device, the operating system has to be programmed with the storage device type (i.e. magnetic, optical, etc.), the interface type (i.e. IDE, SCSI, etc.), and the physical parameters of the storage device (i.e. number of cylinders, sectors, heads, tracks, etc.).
25 If the computer is connected to a network, the operating system also has to be programmed with the various network interface types (i.e. Ethernet, Token Ring, ATM, etc.), the network protocols (i.e. TCP/IP, IPX, AppleTalk, SNA, etc.), and any particular methods used to communicate with network resources (i.e. servers).

WO 03/038643

PCT/US02/31991

[0004] In addition, the operating system also has to be able to manage files on storage devices and over a network. On local storage devices, the operating system typically uses lookup tables or indices, usually referred as file allocation tables, to manage the files. Over the network, the operating system has to be programmed to communicate with a network server and retrieve files. It is often advantageous for the operating system to translate the file information received from the network into a format resembling the file allocation tables to simplify the file retrieval process.

[0005] The way an operating system manages components and their functions add complexity to a computer. The complexity is easily seen during the setup process of a computer. Typically, a computer goes through a setup process that would involve (a) booting from a bootable device that can initiate the installation of an operating system, (b) partitioning and formatting local storage devices, (c) installing the full operating system, (d) configuring hardware components such as display cards and network interface, (e) installing and configuring applications, etc., until the computer is prepared for everyday general use.

[0006] The complexity is also evident while the computer is in use. Applications and the operating system may require periodic patches or updates, the installation of which frequently involves uninstalling the older versions of the applications or the components of the operating system. Additionally, files may be corrupted due to program errors, user errors, or computer viruses. When this happens, the corrupted files need to be replaced or repaired, a process that may involve reinstalling the applications that use the corrupted files or even possibly reinstalling the operating system itself in some instances.

[0007] The complexity involved in using a computer usually results in high maintenance and support costs being incurred. In a business environment, the support costs can easily reach thousands of dollars per user or per computing device. Additionally, the cost of maintaining computers increases because work productivity tends to decrease significantly, often to zero, when computer related problems arise.

[0008] Problems related to hardware malfunction, except problems related to storage devices, may often be resolved within a short amount of time. For instance, a broken component, such as a video card or a network interface card, may be quickly replaced with an identical component. However, computer repair may

WO 03/038643

PCT/US02/31991

become a lengthy process if the problems are related to storage devices or the files stored on the storage devices. When a hard drive in a computer malfunctions or corrupted files cause problems, the repair and recovery process might involve reinitializing the hard drive, reinstalling the operating system, and/or reinstalling applications.

5

[0009] Numerous methods are presently available to reduce the complexity of computers, shorten the recovery process when problems occur, or to minimize the need for a recovery process altogether. Some of the common methods are cloning the storage device, booting the computer from a network, utilizing specialized computer management software, and applying file level security.

10

[0010] By cloning the storage device, the installation process may be shortened. A computer is first completely set up with a full set of applications. Then the storage device is cloned or duplicated as an "image" file. The image file may then be used to reset the computer to its original condition or to set up identically equipped computers. Many consumer-oriented computers come with recovery CD ROM's containing the factory default image that can be used to restore the storage device to its factory default condition. The drawback of this method is that a new image of the storage device has to be created whenever there is a change in the operating system, applications, or any other files stored on the storage device. Complications may arise using this method in instances when it is necessary to apply patches or updates to the installed software after the storage device is restored from an old image.

15

20

[0011] The network boot method is often used in conjunction with simple computers that download necessary files from a network server. The computer usually uses a well-know network service, such as BOOTP, TFTP, or PXE, to download and execute a small basic portion of an operating system, which in turn can start downloading the rest of the operating system and any applications. The drawback of this method is that if the computer does not have a local storage device, it has to go through the same boot process of downloading needed files whenever it is powered on or reset. If the computer has a local storage device, this process can benefit by storing downloaded files locally. But then the operating system downloaded over the network is, once again, responsible for the often complex task of managing hardware components and files stored on the local storage device.

25

30

WO 03/038643

PCT/US02/31991

[0012] The computer management software method is used to enhance the operating system by adding additional software components as agents, daemons, or services. One typical way of using this method is to use anti-virus software that constantly scans stored files for any computer virus infection. This method may also
5 be implemented by adding a software component that constantly monitors important files on the local storage device and attempts to self-heal any damaged or corrupted files. An additional implementation adds a software component that handles file updates pushed out from a server as a part of a computer management tool. The drawback of this method is that the software components acting as agents,
10 daemons, or services are highly dependent on the operating system. The operating system has to provide necessary functions, such as managing local storage devices or network interfaces, for these software components to work properly.

[0013] Many operating systems can also apply file level or directory level security to provide certain level of protection against computer viruses, unauthorized access,
15 user errors, or application errors that can corrupt important files. The drawback of this method is that it is operating system dependant, and a super user, an administrator, or a process running with full access privileges can accidentally modify, delete, or corrupt important files in the local storage.

[0014] The above methods, by themselves or in combination with other methods,
20 provide some help in reducing the complexities involved with computers. However, none of the methods fundamentally changes how the operating system manages the components of a computer. Thus, a new approach is needed for managing computers and simplifying the manner in which files are distributed over a network.

SUMMARY OF THE INVENTION

[0015] The present invention is directed to a read-only storage device, a system
25 incorporating the device, and a method of distributing files over a computer network. The storage device comprises a processor communicably connected to a computer interface, a network interface, and a storage means. The computer interface and the network interface allow the processor to communicate with a computer and a file
30 server, respectively, when the storage device is connected to a computer and a file server. The computer may request a file from the processor. When such a request is made, the processor first checks the storage means to determine whether the

WO 03/038643

PCT/US02/31991

requested file is cached on the storage means. If the requested file is cached on the storage means, the file is provided to the computer on a read-only basis.

5 [0016] If the requested file is not cached on the storage means, the processor may obtain the requested file from a file server on the network. Provided the requested file is available from the file server, the processor obtains the file, caches the file on the storage means, and provides to the computer on a read-only basis. Once a file is cached on the storage means, the file is available to the computer upon future requests. If the file is not available from the file server, then the processor returns a file unavailable notice to the computer.

10 [0017] The storage device may also receive commands from the file server to delete a file that may have been cached on the storage means. Upon receiving a delete command, the processor will check the storage means for the presence of the file and, if the file is present on the storage means, delete the cached copy of the file.

15 [0018] Accordingly, it is an object of the present invention to provide a storage device having a network interface which may obtain files from a networked file server and make the obtained files available to a computer on a read-only basis. An additional object of the invention is to provide a networked computer system including the storage device, thereby providing a system and method to simplify aspects of file management over a network. Other objects and advantages will appear hereinafter.

20

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] In the drawings, wherein like reference numerals refer to similar components:

25 Figure 1 illustrates a block diagram of a networked computer according to the prior art;

Figure 2 illustrates a block diagram of a read-only storage device in accordance with a preferred embodiment of the present invention;

Figure 3 illustrates a block diagram of the device of Figure 2 incorporated into a networked computer;

30 Figure 4 is a flow chart illustrating an algorithm the device may use in response to a file request from the networked computer of Figure 3;

WO 03/038643

PCT/US02/31991

Figure 5 illustrates a block diagram of the device of Figure 2 incorporated into a networked computer;

Figure 6 illustrates a block diagram of a read-only storage device in accordance with a first alternative embodiment of the present invention;

5 Figure 7 illustrates a block diagram of the device of Figure 6 incorporated into a networked computer;

Figure 8 illustrates a block diagram of a read-only storage device in accordance with a second alternative embodiment of the present invention; and

10 Figure 9 illustrates a block diagram of the device of Figure 8 incorporated into a networked computer.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0020] Turning in detail to the drawings, Figure 1 is representative of a typical networked computer 101 as exhibited in the prior art. As used herein, the networked computer 101 may be a personal or business computer, whether an off-the-shelf or a custom built computer, or it may be a specialty device such as a personal digital assistant, a telephone, or a networked appliance. The computer 101 is shown with common components such as the central processing unit 104, the internal interface bus 105 that connects and allows communication between the various components, the network interface 107 that may be wireless or wired to connect the computer 101 to a network 114 having network resources such as a file and print sharing server 115 and/or a network applications server 116, and a secondary storage device 108, such as a hard drive or non-volatile random access memory, where the operating system 103 and application files 102 are typically stored. These components are common to networked computers and are therefore well known to those skilled in the art.

[0021] The operating system 103 is responsible managing the interaction between the components to form a functional computer. In particular, the operating system 103 handles file requests that may originate from applications 102 or from the operating system 103 itself. These file requests arise when the applications 102 or operating system 103 require access to a file which resides on either the secondary storage device 108, the file and print sharing server 115, or the network applications server 116. To handle the file requests, the operating system 103 has to be

WO 03/038643

PCT/US02/31991

programmed with the characteristics of numerous storage devices 108 and network interfaces 107 to account for the wide variety of storage devices and network protocols that may be used in conjunction with the computer. The operating system 103 must also be programmed with information that enables it to keep track of files stored on the secondary storage device 108 and how to integrate into the computer's operation the various resources which are available from the various network servers 115, 116. Therefore, under normal operating circumstances, the operating system 103 of the computer 101 must address many different complex tasks.

[0022] Figure 2 is a block representation of a cache based device (CBD) 202 in accordance with a preferred embodiment of the present invention. The CBD is intended to be used in conjunction with a computer, such as the computer illustrated in Figure 1, to simplify the tasks required of the operating system when managing the various computer components and the interaction with one or more network servers. The CBD 202 has four basic components: the main computer interface 203 that connects the CBD 202 to a computer, the network interface 207 that connects the CBD 202 to a network, the cache engine (or processor) 206 that manages the components within the CBD 202 and processes file requests from the operating system of the computer, and the cache (or storage means) 204 that the cache engine 206 uses to store files locally.

[0023] The cache engine 206 may be any type of central processing unit appropriate to control the functions of the CBD 202 as described herein. The particular type of central processing unit used is a matter of design choice. Acceptable central processing units are available from the integrated circuit divisions of vendors such as ARM, Hewlett-Packard (HP), International Business Machines (IBM), Intel, and Advanced Micro Devices (AMD).

[0024] The cache engine 206 is programmed to manage the files stored on the cache 204 and interact with the computer and network as needed. The programming may be in any programming language, such as C++, Java, etc., and the compiled program may be embedded into the cache engine 206 itself, or it may be stored in a separate memory specifically provided for such a purpose. Alternatively, the programming may be stored within the cache 204 for access by the cache engine 206. The programming permits the cache engine 206 to communicate with a computer, giving the computer access to files stored in the cache 204 on a

WO 03/038643

PCT/US02/31991

read-only basis. The programming also permits the cache engine 206 to communicate with a network through the network interface 207. Thus, when a computer requests a file from the CBD 202 that is not stored in the cache 204, the cache engine 206 may request the file from a server on the network the CBD 202 is connected to in an attempt to obtain the requested file. If the requested file is found on the network, the cache engine 206 will store the file in the cache 204 and give the computer access to the file on a read-only basis. If the requested file is not available, either within the cache 204 or on the network, the cache engine 206 may provide the computer with a notice that the requested file is unavailable.

5
10 [0025] The computer interface 203 and the network interface 207 may also be any type of appropriate interface that permits the cache engine 206 to communicate with a computer and a network, respectively. The particular method of interconnecting the computer and the network to the CBD 202 is a matter of design choice. Methods utilizing commonly known formats such as IDE, ATA, SCSI, and IEEE 1394, among others, are acceptable for the computer interface 203. Methods utilizing commonly known network interface types, such as Ethernet, Token Ring, and ATM, among others, and network protocols such as TCP/IP, IPX, AppleTalk, and SNA, among others, are acceptable for the network interface 207. Such components are generally widely available from a variety of computer hardware vendors.

15
20 [0026] A CBD may thus be operating system independent because the CBD does not have to conform to any particular protocols or file structures. An operating system needs to be programmed only with the protocol needed to request files stored on the CBD because the cache engine, and not the operating system, manages the files stored on a CBD. Thus, communication between the operating system and the CBD may be performed using a small number of procedures. These procedures would be the only ones dependent on the operating system and system architecture, and they can be standardized on any hardware platform.

25
30 [0027] The particular type of storage used for the cache 204 is also a matter of design choice. The cache 204 may be any type of random access media that is well known to those skilled in the art, such as a hard drive, DRAM, SDRAM, flash memory, and any type of removable random access media. The programming used by the cache engine 206 must be adapted to function with the particular type of random access media chosen.

WO 03/038643

PCT/US02/31991

[0028] Figure 3 illustrates a networked computer 301 incorporating a CBD 202 in accordance with a preferred embodiment of the present invention. In the embodiment shown, the CBD 202 is incorporated internally to the computer 301. However, the CBD 202 may also be external to the computer 301 without losing any of the functionality described herein. As in the prior art, the computer includes a CPU 304, an interface bus 305, a secondary storage device 308, and a network interface 307. The operating system 303 integrates the operation of the various components and the applications 302 are executed within the environment created by the operating system 303. The computer 301 is connected to a file and print server 315 and an application server 316 via the network interface 307. The CBD 202 is communicably connected to the computer by connecting the computer interface (not shown in Figure 3) to the interface bus 305.

[0029] The networked computer 301 of Figure 3 is connected to two separate networks. The first network 309 connects the CBD 202 directly to a CBD file server 314 that serves only the CBD 202. The second network 311 connects the network interface 307 of the computer to a file and print sharing server 315 and a server for network applications 316. The networks may be formed of copper-based wiring, optical fibers, wireless communications, or any other type of networking known to those skilled in the art. Additionally, the first and second networks 309, 311 do not have to be the same type. For example, the first network 309 may be wireless and the second network 311 may be formed of copper-based wiring.

[0030] The two networks are preferably independent networks that do not interact. However, the two networks may be merged into a single network, as illustrated in Figure 5, with a single network connection being shared by the CBD 202 and the network interface 307 through a network concentrator 513. The network concentrator 513 is communicably connected to the network 509 to pass network communications through to the CBD 202 and the network interface 307. Under such circumstances, it is desirable, but not necessary, to have the CBD 202 be the only device type on the network that communicates with the CBD file server 314. In other words, the CBD file server 314 will not communicate with any device other than the CBD 202 and the CBD 202 will not communicate with any server other than the CBD file server 314.

WO 03/038643

PCT/US02/31991

[0031] In a first alternative embodiment illustrated in Figure 6, a network concentrator 608, such as the one previously described, is incorporated into the CBD 602. Such a CBD 602 may be placed into a computer 701 as is illustrated in Figure 7. The computer 701 in Figure 7 has the same components as the computer of Figure 3. However, the computer 701 is connected to the network 509 through the CBD 602. The CBD file server 314, the file and print sharing server 315, and the network application server 316 are all connected to the single network 509. The CBD 602 is connected to the network 509 by way of the network concentrator 608. The network concentrator 608 is additionally connected to the network interface 307 to pass network communications through to the computer 701. This connection may be formed external to the computer, as is shown, or internal to the computer.

[0032] A second alternative embodiment is illustrated in Figure 8. In this embodiment, the CBD 802 may provide a computer with a network connection without the use of a network concentrator. The network interface 807 of the CBD 802 passes network communications intended for the computer directly to the computer interface 803. Thus, when the CBD 802 is incorporated into a computer 901, as illustrated in Figure 9, the network communications are passed through the CBD 802 to the internal interface bus 305 of the computer 901. In this configuration, the CBD file server 314, the file and print sharing server 315, and the network application server 316 may all be connected to a single network 509 which provides network communications to both the CBD 802 and the computer 901. Additionally, with the implementation of known encryption techniques for the communications between the CBD 802 and the CBD file server 614, this configuration can be made as desirable as the preferred configuration depicted in Figure 3.

[0033] Figure 4 illustrates a flowchart of the preferred logic the cache engine follows in responding to file requests from the operating system. When the cache engine receives a file request 402 from the operating system, it checks whether the request is for a file read 403, as opposed to, for example a file write or file delete. If the request is not for a file read, the cache engine rejects the request 404 by returning an "Invalid Request" notice to the operating system. If the request is for a file read, the cache engine checks the cache 406 to see if the file is available from the cache. If the file is available from the cache, the cache engine returns the file 407 to the operating system. If the file is not available from the cache, the cache

WO 03/038643

PCT/US02/31991

engine sends a request 408 for the file to a CBD file server. The request may be in the form of a broadcast to any and all CBD file servers on the network, or the request may be made to a specific CBD file server on the network. The method used depends on the particular network the CBD is connected to. If the file is available from a CBD file server 409, the cache engine copies the file 410 from the CBD server to its local cache and returns the file 407 to the operating system. If the file is not available from a CBD file server, the cache engine notifies the operating system 411 by returning a "File Not Available" notice to the operating system.

[0034] In addition to retrieving files from the CBD server at the request of the operating system, the CBD may also be pre-programmed to retrieve and cache files as needed or as instructed by the CBD file server. The files cached on the CBD may be deleted based on one or more criteria, thus forcing the CBD to re-retrieve the file from the CBD server at the next instance the file is needed. File deletions may be forced at the prompting of the CBD server, or certain files may have a file expiration tag associated with the file, thus causing the CBD to delete the file at the time indicated by the file expiration tag. The file expiration tag may indicate an absolute date or time, or a date or time based on an anticipated event.

[0035] The following are examples that demonstrate some of the benefits of the present invention:

[0036] 1. Installation of an Operating System on a Personal Computer

[0037] Since today's operating systems for personal computers provide many functions, it is frequently necessary to install an operating system on a local hard drive. During the boot process, a ROM-based BIOS loads the operating system by first reading the bootstrap stored on the hard drive. The bootstrap provides information about the first executable code needed to load the operating system. Based on the information provided by the bootstrap, the BIOS loads the first file into memory and passes control over to the file, which starts loading other files necessary for loading the entire operating system.

[0038] On a computer that includes a CBD, the operating system would not need to be installed on a local hard drive prior to the boot process. The BIOS may request from the CBD the first file needed to start loading the operating system. For a new computer, the cache on the CBD would most likely be empty at this point. When the cache engine receives the request for the first file from the computer, the first file is

WO 03/038643

PCT/US02/31991

retrieved from the CBD file server. The first file may then be saved to the cache and supplied to the BIOS. The file may then take control of the boot process from the BIOS. File requests may continue to the CBD, with the requested files being retrieved, saved to the cache, and supplied to the requestor. Thus, all the files
5 needed to initially load the operating system may be obtained by the CBD from the CBD file server on an as needed basis.

[0039] While this process is somewhat similar to a computer booting from the network by well-known processes, such as BOOTP, TFTP, or PXE, the primary benefit of utilizing a CBD becomes apparent when the computer is rebooted.

10 Rebooting a computer that utilizes BOOTP, for example, requires a repeat of the process of downloading files from the network. In contrast, rebooting a computer that includes a CBD does not require downloading all the files from the network again because all the needed files are stored in and supplied from the cache within the CBD. This process also differs from storing the operating system on a local hard
15 drive because, in doing so, the operating system must manage the files on the hard drive. In contrast, the operating system running on a computer including a CBD does not manage the files in the CBD cache.

[0040] The process is also somewhat similar to a computer booting from locally attached read-only devices, such as a CD-ROM. But the benefit of utilizing a CBD
20 becomes apparent because of the flexibility of a CBD. When a file is needed to be replaced or updated on a CD, either the CD has to be replaced with a new one or techniques similar to those disclosed in U.S. Patent Nos. 5,832,263 and 5,790,848 must be employed. In contrast, files cached by a CBD may be replaced or updated dynamically by an administrator. The administrator may simply have the CBD file
25 server send a "delete" command to the CBD to delete a particular file. After deleting the file, the next time that file is requested or needed, the file or its updated replacement must be retrieved from the CBD file server. Thus, it is possible for a CBD to always have up-to-date files in its cache.

[0041] 2. Installation of Applications on a Personal Computer

30 **[0042]** The process of installing applications is similar to the process of installing the operating system. The operating system may present a list of available applications to the user. When the user selects an application, the operating system requests the necessary files from CBD. Where the application has never been used

WO 03/038643

PCT/US02/31991

before, the CBD will retrieve the necessary files from the CBD file server, store the files within the cache, and supply the files to the operating system so that the operating system may load the application.

5 [0043] The speed of accessing an application for the first time in this manner would be comparable to the speed of launching a server-based application (i.e., Microsoft Word[®] installed on a network server for use by multiple users on the network). The advantage of using a CBD becomes evident when the application is accessed after the first time. Unlike a server-based application that has to be retrieved from a network sever every time it is accessed, the application is cached to the CBD thus allowing the files to be quickly accessed for each subsequent desired use. Additionally, components of an application or an application itself may be dynamically replaced or updated by an administrator by having the CBD file server send a "delete" command to the CBD.

10 [0044] 3. Protection of Files Important to the Operating System and Applications

15 [0045] On a typical computer, files stored in a local hard drive are generally subject to accidental deletion, modification, or corruption caused by user errors or computer viruses. Deleted, modified, or corrupted files may lead to application errors, operating system errors, or the computer itself becoming non-functional. Therefore, files important to the operating system and applications should be subject to the highest level of protection.

20 [0046] There are many conventional ways to protect files. In a corporate environment, an administrator may set up a computer and apply file-level securities so that the administrator is the only user having full access (read, write, execute, modify, add, delete, etc.), while a regular user may have only limited access (read and execute, for example). Once applied, maintaining this type of file-level security usually becomes a substantial task of the operating system. The operating system has to maintain a complex list or database to manage files and security. The operating system may also include a self-healing function that attempts to automatically repair files that might be damaged. In addition, antivirus software may be installed to protect the PC from viruses. But, if the file-level security system is somehow compromised, then files on the hard disk are exposed to the risk of being deleted, modified, or corrupted. Additionally, if the antivirus software is not up-to-date, it may not be able to provide proper defense against viruses.

WO 03/038643

PCT/US02/31991

[0047] In contrast, a computer with a CBD may fully protect important files against human errors or viruses. Files cached in the CBD can not be deleted, modified, or corrupted by the user or a virus from the computer since the computer has read-only access to the files. In addition, neither an administrator nor the operating system
5 can change files stored in CBD's cache as the CBD will only accept files or updates from the CBD file server. Therefore, a CBD can naturally protect files important to the operating system or applications.

[0048] In addition, if the operating system is configured not to load any programs or executable codes from any device other than the CBD, additional protection is
10 provided against unauthorized programs or codes, such as computer viruses, worms, or scripts. The additional protection is inherent with such a system because all files supplied to a computer through a CBD may be verified at the CBD file server as being safe for use.

[0049] 4. Simplified Software License Inventory and Management

[0050] In a corporate environment, managing software licenses and usage is quite
15 complex. Users are initially given PC's with a set of pre-loaded applications that have legal licenses. But users may start installing applications on their own, or they may never use some of the pre-installed applications for which license fees have been paid. Thus managing which software is used by a user, which software is
20 licensed, and which software is not licensed may easily turn into a difficult task.

[0051] CBD's may be implemented on a company-wide basis to make the task of managing software and software licenses less difficult. First, a user cannot install an application on a CBD because the CBD is a read-only device. Therefore, software installed on a local hard drive can be easily identified as user installed software.
25 Second, an administrator may easily obtain an accurate count of the applications in use by monitoring the number of times an application is downloaded from the CBD file server. An administrator may at any time obtain an accurate count of applications in use by issuing a "delete" command to delete key components of applications from the CBD's, thus requiring that component to be requested from the
30 CBD file server the next time the application or applications are used. Third, application vendors may implement different licensing models, such as per-use licensing, if CBD's are used because of the ease with which application use may be

WO 03/038643

PCT/US02/31991

monitored. A vendor might even gain greater benefits from CBD's configured to access the CBD file server through the internet.

[0052] 5. Database Applications

[0053] Database applications may benefit from use of a CBD. Address book applications or map applications are examples of applications that utilize collections of records organized into databases. These applications often include records that are relatively static but may require periodic updates.

[0054] To utilize a CBD in conjunction with a database application, a file containing a lookup table may be provided to the CBD, with each entry in the table representing a record in the database. Initially, the application would obtain the lookup table file from the CBD, the file being requested from the CBD file server if necessary. Each record from the lookup table file that is desired by the user may be retrieved from the CBD file server and cached in the CBD for access by the user. After a period of use, the cache will contain all the records accessed by the user. If any of the records cached by the user need to be updated, the CBD file server may issue a "delete" command, thereby deleting particular records from the cache. The CBD file server may also update the entire lookup table file by issuing an appropriate "delete" command. After a record or the lookup table file has been deleted, the next time a user accesses the database, the CBD will necessarily have to retrieve the updated version of the record or file from the CBD file server.

[0055] This method of database management may be advantageously implemented with wireless devices such as cellular phones. Cellular phones frequently store a full list of contact information, but many of the entries may not be used at all. However, because of all the entries, synchronizing the database with a master database on a company server may be time consuming. A cell phone equipped with a CBD, the CBD having a flash memory as its cache, may reduce the time needed to maintain an up-to-date database in the cell phone. Initially, the CBD in the cell phone would download the record lookup table file. As the user selects a particular entry in the lookup table for access, the CBD may initiate a wireless connection to the CBD file server and retrieve the entry to its cache, thus providing the user with the desired information. All entries retrieved in this manner are cached within the CBD for later access by the user without the need to further access the CBD file server. Additionally, the CBD file server may initiate a wireless connection

WO 03/038643

PCT/US02/31991

with the CBD and send "delete" commands to delete the lookup table file or particular record. A cell phone would therefore only have in its CBD those records that the user regularly needs and updates may be performed without great inconvenience to the user.

5 **[0056]** 6) CBD's for Internet Service Providers or Application Service Providers

[0057] Internet Service Providers or Application Service Providers may provide CBD's to subscribers to provide services. A CBD may be configured to access the provider's CBD file server to provide software and other services to the customers over the Internet.

10 **[0058]** For example, a service provider might provide a customer a personal computer equipped with a CBD. The CBD would be pre-configured to use the service provider's CBD file server over the Internet. The service provider might pre-install an operating system and some basic applications in the cache of a CBD to shorten the initial download time since internet connections are often much slower
15 than LAN connections.

[0059] When the computer is connected to the Internet, the CBD would operate normally, downloading files to its cache and receiving "delete" command as needed, all via the internet. If the user requires a different application or operating system, a request could be made to the service provider and the new application or operating
20 system could be accessed by the user in the manner described herein. Such an application gives the service provider complete control over the content of the CBD because the user has read-only access to the CBD. Thus, the service provider may manage and update a user's system without much difficulty and may create additional revenue by placement of advertisements on the user's computer.
25 Advertisements may also benefit the user by reducing the cost of computer ownership.

[0060] Thus, a read-only storage device having a network interface, systems including the device, and methods of distributing files over a network are disclosed. While embodiments of this invention have been shown and described, it would be
30 apparent to those skilled in the art that many more modifications are possible without departing from the inventive concepts herein. The invention, therefore, is not to be restricted except in the spirit of the following claims.

CLAIMS

1. A storage device comprising:

a device processor;

a computer interface communicably connected to the device processor, wherein the computer interface is adapted to communicably connect to a single computer and to enable communications exclusively between the device processor and a central processing unit (CPU) of the single computer;

a network interface communicably connected to the device processor, wherein the network interface is adapted to communicate over a network and to enable communications exclusively between the device processor and one or more select file servers to the exclusion of other file servers; and

a storage means communicably connected to the device processor, wherein:

the device processor is adapted to manage files stored on the storage means and to exclude the CPU from managing the files stored on the storage means such that the CPU has read-only access to the files stored on the storage means; and

upon receiving a request for a first file from the CPU, the device processor is adapted to (1) determine if the first file exists on the storage means, (2) if the first file does not exist on the storage means, obtain the first file from the select file servers and save the obtained first file on the storage means, and (3) provide the first file to the CPU on a read-only basis.

2. The storage device of claim 0, wherein the device processor is adapted to provide to the CPU a response indicating that the file is not available if the file does not exist on the storage means and is not obtainable from the select file servers.

3. The storage device of claim 0, wherein the CPU is communicably connected to a network resource through the network interface of the storage device.

4. The storage device of claim 0, wherein the storage means comprises random access media.

5. A computer network comprising:

one or more select file servers;

a network resource;

a computer communicably connected to the network resource over a network, the computer being remotely disposed over a network from the select file servers and the network resource;

a storage device communicably connected to the computer and to the select file servers over a network and comprising a device processor, a computer interface, a network interface, and a storage means, wherein

the device processor is adapted to employ the network interface for communications with the select file servers to the exclusion of other file servers;

the device processor is adapted to employ the computer interface for communications exclusively between the device processor and a central processing unit (CPU) of the computer;

the device processor is adapted to manage files stored on the storage means and to exclude the CPU from managing the files stored on the storage means such that the CPU has read-only access to the files stored on the storage means; and

upon receiving a request for a first file from the CPU, the device processor is adapted to (1) determine if the first file exists on the storage means, (2) if the first file does not exist on the storage means, obtain the first file from the select

file servers and save the obtained first file on the storage means, and (3) provide the first file to the CPU on a read-only basis.

6. The computer network of claim 5, wherein the device processor is adapted to provide to the CPU a response indicating that the file is not available if the file does not exist on the storage means and is not obtainable from the select file servers.

7. The computer network of claim 5, wherein the computer is communicably connected to the network resource through the network interface of the storage device.

8. The computer network of claim 5, wherein the storage means comprises random access media.

9. A method of providing a file to a computer comprising:

receiving, with a storage device, a request from the computer for the file, wherein the storage device comprises a device processor, a computer interface, a network interface, and a storage means, the device processor being adapted to employ the network interface for communications with one or more select file servers to the exclusion of other file servers, the device processor being adapted to employ the computer interface for communications exclusively between the device processor and a central processing unit (CPU) of the computer, and the device processor being adapted to manage files stored on the storage means and to exclude the CPU from managing the files stored on the storage means such that the CPU has read-only access to the files stored on the storage means;

determining, with the device processor, if the file exists on the storage means;

obtaining, with the device processor, the file from the select file servers if the file does not exist on the storage means, and saving the obtained file on the storage means; and

providing, with the device processor, the file to the CPU on a read-only basis.

10. The method of claim 9, further comprising providing, with the device processor, to the CPU a response indicating that the file is not available if the file does not exist on the storage means and is not obtainable from the select file servers.

11. The method of claim 9, further comprising deleting, with the device processor, a saved file from the storage means upon receiving a command from one or more select file servers to delete the saved file.

12. A method for providing updated files to a computer comprising:
providing, with a device processor of a storage device, to a central processing unit (CPU) of the computer a list identifying a plurality of files which the device processor may retrieve from a file server;
obtaining, with the device processor, from the file server a first file of the plurality of files when the device processor receives a request for the first file from the CPU;
saving, with the device processor, the first file on a storage means of the storage device, wherein the device processor is adapted to manage files stored on the storage means and to exclude the CPU from managing the files stored on the storage means such that the CPU has read-only access to the files stored on the storage means;
providing, with the device processor, the file to the CPU on a read-only basis;
deleting, with the device processor, the saved first file from the storage means upon receiving a command from to delete the saved file.

13. The method of claim 12, further comprising deleting, with the device processor, the saved first file from the storage means upon receiving a notice from the file server that an updated version of the first file exists on the file server.

14. A storage device comprising:
a device processor;

a computer interface communicably connected to the device processor, wherein the computer interface is adapted to communicably connect to a single computer and to enable communications exclusively between the device processor and a central processing unit (CPU) of the single computer;

a network interface communicably connected to the device processor, wherein the network interface is adapted to communicate over a network and to enable communications exclusively between the device processor and one or more select file servers to the exclusion of other file servers; and

a storage means communicably connected to the device processor, wherein:

the device processor is adapted to manage files stored on the storage means and to exclude the CPU from managing the files stored on the storage means such that the CPU has read-only access to the files stored on the storage means; and

upon receiving a request for a boot file when the computer is booted, the boot file being a bootstrap file or an operating system file, the device processor is adapted to (1) determine if the boot file exists on the storage means, (2) if the boot file does not exist on the storage means, obtain the boot file from the select file servers and save the obtained boot file on the storage means, and (3) provide the boot file to the CPU on a read-only basis.

15. The storage device of claim 14, wherein the device processor is adapted to provide to the CPU a response indicating that the boot file is not available if the boot file does not exist on the storage means and is not obtainable from the select file servers.

16. The storage device of claim 14, wherein the CPU is communicably connected to a network resource through the network interface of the storage device.

17. The storage device of claim 14, wherein the storage means comprises random access media.

18. A computer network comprising:

one or more select file servers;

a network resource;

a computer communicably connected to the network resource over a network, the computer being remotely disposed over a network from the select file servers and the network resource;

a storage device communicably connected to the computer and to the select file servers over a network, and comprising a processor, a computer interface, a network interface, and a storage means, wherein

the device processor is adapted to employ the network interface for communications with the select file servers to the exclusion of other file servers;

the device processor is adapted to employ the computer interface for communications exclusively between the device processor and a central processing unit (CPU) of the computer;

the device processor is adapted to manage files stored on the storage means and to exclude the CPU from managing the files stored on the storage means such that the CPU has read-only access to the files stored on the storage means; and

upon receiving a request for a boot file when the computer is booted, the boot file being a bootstrap file or an operating system file, the device processor is adapted to (1) determine if the boot file exists on the storage means, (2) if the boot file does not exist on the storage means, obtain the boot file from the select file servers and save the obtained boot file on the storage means, and (3) provide the boot file to the CPU on a read-only basis.

19. The computer network of claim 18, wherein the device processor is adapted to provide to the CPU a response indicating that the boot file is not available if the boot file does not exist on the storage means and is not obtainable from the select file servers.

20. The computer network of claim 18, wherein the CPU is communicably connected to the network resource through the network interface of the storage device.

21. The storage device of claim 18, wherein the storage means comprises random access media.

ABSTRACT

A Read-only storage device having network interface, a system including the device, and a method of distributing files over a network. The storage device comprises a device processor, a computer interface, a network interface, and a storage means. The device processor communicates with a computer and a file server through the computer interface and the network interface, respectively. The computer may request a file from the device processor and the device processor responds by either (1) providing the requested file to the computer on a read-only basis if the file is cached on the storage means, (2) obtaining the file from the file server if the file is available from the file server, caching the obtained file on the storage means, and providing the obtained file to the computer on a read-only basis, or (3) returning a file unavailable notice to the computer.

WO 03/038643 PCT/US02/31991

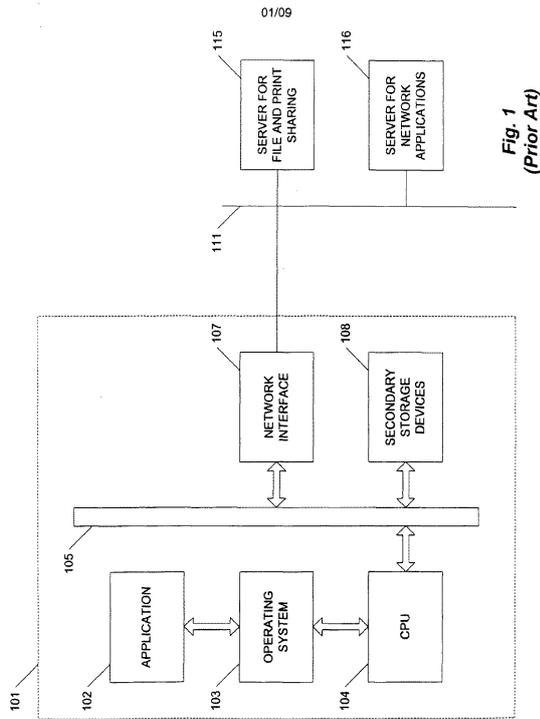


Fig. 1 (Prior Art)

WO 03/038643 PCT/US02/31991

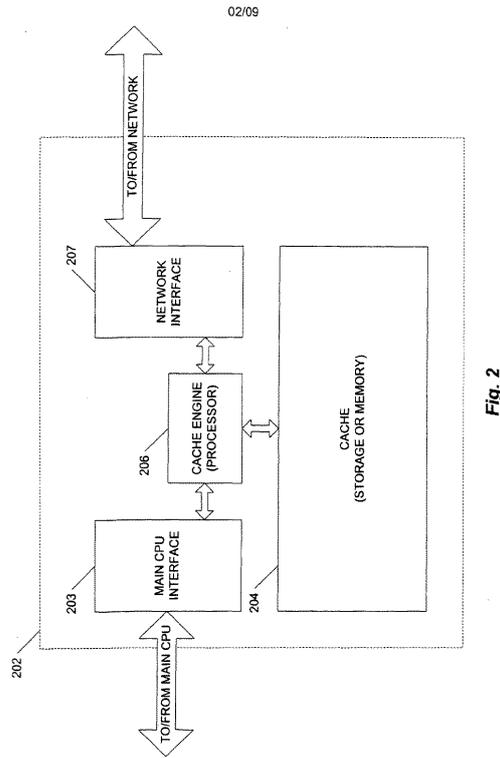


Fig. 2

WO 03/038643 PCT/US02/31991

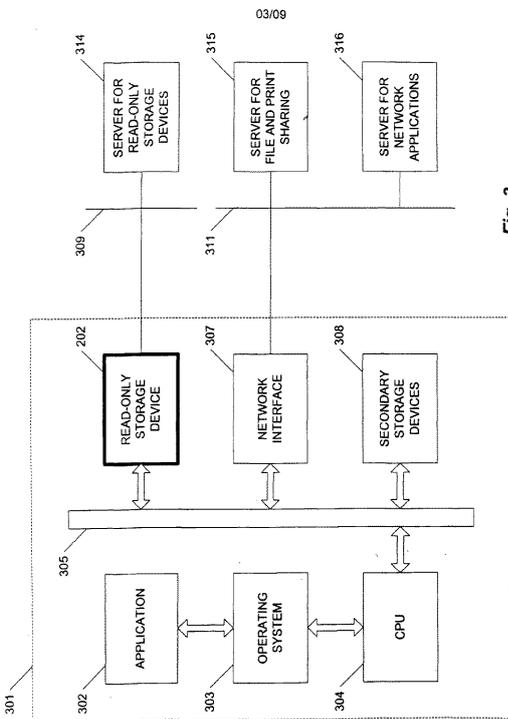


Fig. 3

WO 03/038643 PCT/US02/31991

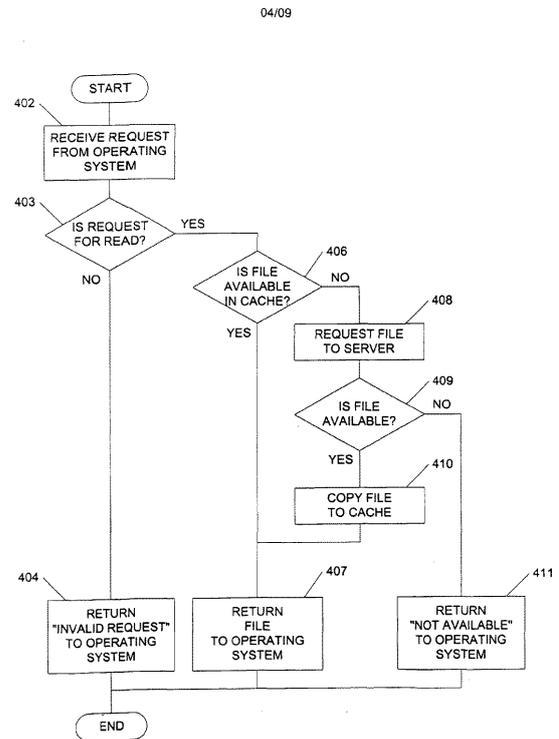
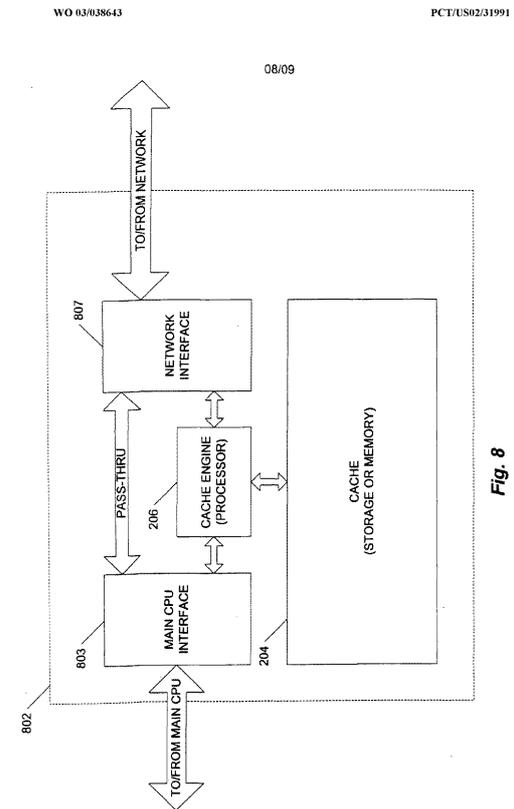
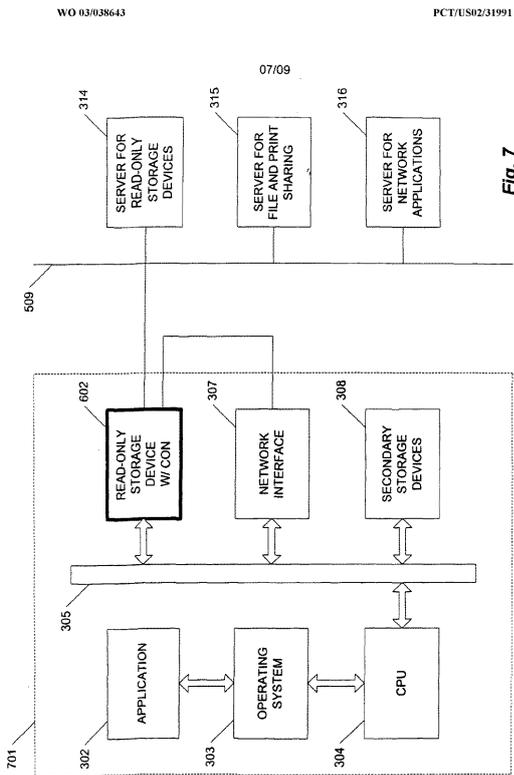
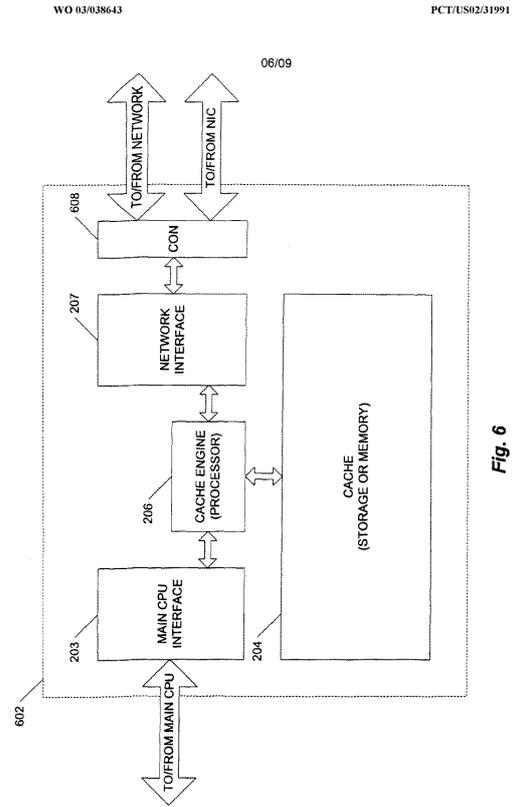
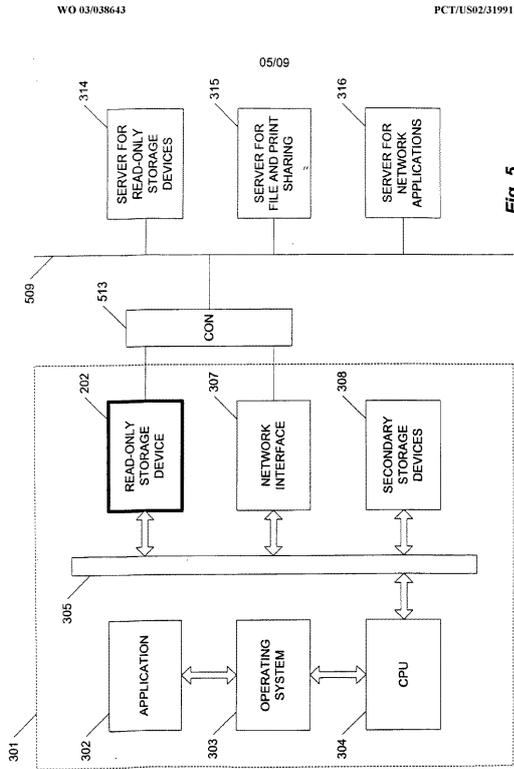


Fig. 4



WO 03/038643

PCT/US02/31991

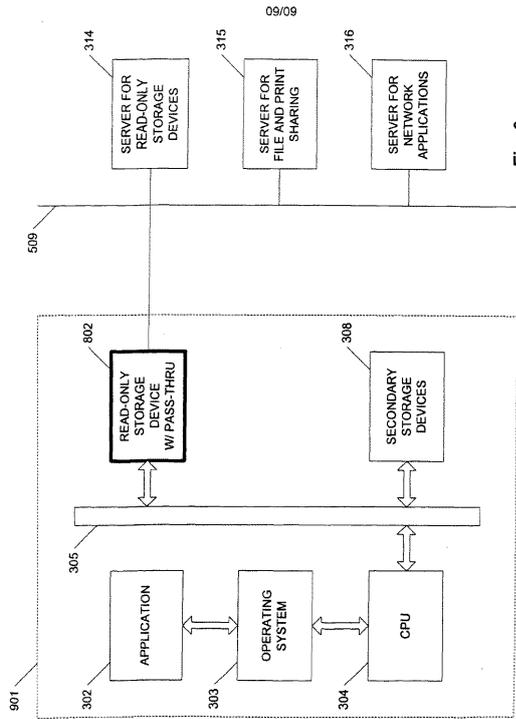


Fig. 9