(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4496093号
(P4496093)

(45)発行日　平成22年7月7日(2010.7.7)　(24)登録日　平成22年4月16日(2010.4.16)

(51)Int.Cl.		F I		
G06F 13/00	(2006.01)	G06F	13/00	351N
G06F 11/16	(2006.01)	G06F	11/16	310C
G06F 11/30	(2006.01)	G06F	11/30	D
		G06F	11/30	E
		G06F	11/30	K

請求項の数 19　(全 25 頁)

(21)出願番号　特願2005-6858(P2005-6858)
(22)出願日　平成17年1月13日(2005.1.13)
(65)公開番号　特開2005-209191(P2005-209191A)
(43)公開日　平成17年8月4日(2005.8.4)
審査請求日　平成17年1月13日(2005.1.13)
審判番号　不服2008-7416(P2008-7416/J1)
審判請求日　平成20年3月26日(2008.3.26)
(31)優先権主張番号　10/761164
(32)優先日　平成16年1月20日(2004.1.20)
(33)優先権主張国　米国(US)

早期審査対象出願

(73)特許権者　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100108501
弁理士　上野　剛史
(74)代理人　100112690
弁理士　太佐　種一
(74)代理人　100091568
弁理士　市位　嘉宏

最終頁に続く

(54)【発明の名称】高可用性システムの遠隔エンタープライズ管理

(57)【特許請求の範囲】

【請求項１】

高可用性システムの遠隔エンタープライズ管理を可能にするためのシステムであって、

ネットワークを介して遠隔エンタープライズ・サーバに通信接続された複数の高可用性システムのうちの特定の高可用性システムを有し、

前記特定の高可用性システムが、

ウェブ・アプリケーションをサポートするためのミドルウェア・スタックを走らせる一次ノードであって、第１のＩＰアドレスと、該第１のＩＰアドレスとは異なり、要求を向けるための仮想ＩＰアドレスを割り当てられる前記１次ノードと、

前記１次ノードの前記ミドルウェア・スタックの複数の層をミラーリングするための冗長ミドルウェア・スタックを走らせる２次ノードであって、前記第１のＩＰアドレス及び前記仮想ＩＰアドレスとは異なる第２のＩＰアドレスを割り当てられ、前記冗長ミドルウェア・スタックの前記複数の層のうちの第１の選択されたものがアクティブであって、前記冗長ミドルウェア・スタックの前記複数の層のうちの第２の選択されたものが待機状態である、前記２次ノードと、

前記アクティブなミドルウェア・スタック、すなわち前記複数の層の前記第１の選択されたものに対してアクセス可能なデータをもち、前記１次ノードと２次ノードの間で共有されるデータ複製区画と、

前記１次ノードのみが前記データ複製区画にアクセスできるように前記データ複製区画をマウントし、前記高可用性システムの前記１次ノードの状況を監視し、該状況がエラー

を示すことに応答して、前記仮想ＩＰアドレスを前記第１のノードから前記第２のノードに転送し、前記１次ノードに対して電源を切り、前記２次ノードのみがアクセスできるように前記データ複製区画を再マウントし、前記データ複製区画内の前記データに対するアクセスを要求する前記冗長ミドルウェア・スタックの前記複数の層のうちの第２の選択されたものを活動化させる、クラスタ管理コントローラと、

前記クラスタ管理コントローラが、前記１次ノードにおいて前記エラーに反応した時を検出し、前記エラーの時点で、前記高可用性システムの複数のコンポーネントの状態を検出するための監視コントローラであって、前記エラーと、前記複数のコンポーネントの状態を、前記複数の高可用性システムの各々から受け取った別のレポートに基づき前記複数の高可用性システムを管理するようにイネーブルされた前記遠隔エンタープライズ・サーバに報告する、前記監視コントローラとを有し、

前記仮想ＩＰアドレスは、前記１次ノードと前記２次ノードのうち、一度に１つのノードのみが利用可能である、

システム。

【請求項２】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項１に記載のシステムであって、前記特定の高可用性システムが、各々が前記クラスタ管理コントローラによって監視されるＪ２ＥＥ準拠ミドルウェア・スタックを実装する複数のサーバをもつ前記１次ノード及び前記２次ノードをさらに備えるシステム。

【請求項３】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項１に記載のシステムであって、前記クラスタ管理コントローラが、前記高可用性システムの１次ノードのステータスを検出するためのハートビート・モニタをさらに備えるシステム。

【請求項４】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項１に記載のシステムであって、前記クラスタ管理コントローラが、
前記高可用性システムの前記１次ノードのミドルウェア層によって提供されるサービスのステータスを検出するためのサービス監視デーモンをさらに備えるシステム。

【請求項５】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項１に記載のシステムであって、前記監視コントローラが、構成要求を前記遠隔エンタープライズ・サーバから受信し、更なるエラーに応答して前記高可用性システムを調整するように前記クラスタ管理コントローラが対応する手法の構成を調整するシステム。

【請求項６】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項１に記載のシステムであって、前記監視コントローラが、構成要求を前記遠隔エンタープライズ・サーバから受信し、前記要求に従って前記高可用性システムのハードウェア構成を調整するシステム。

【請求項７】

高可用性システムの遠隔エンタープライズ管理を可能にするための方法であって、

ネットワークを介して遠隔エンタープライズ・サーバに通信接続された、複数の高可用性システムうちの特定の高可用性システムの少なくとも１つのコンポーネントのステータスを監視する段階と、

前記特定の高可用性システム内の１次ノードで、ウェブ・アプリケーションをサポートするためのミドルウェア・スタックを走らせる段階であって、前記１次ノードが第１のＩＰアドレスを割り当てられ、前記１次ノードはさらに、該第１のＩＰアドレスとは異なり、要求を向けるための仮想ＩＰアドレスを割り当てられる、段階と、

前記特定の高可用性システム内の２次ノードで、前記１次ノードの前記ミドルウェア・スタックの複数の層をミラーリングするための冗長ミドルウェア・スタックを走らせる段階であって、前記２次ノードが前記第１のＩＰアドレスとは異なる第１のＩＰアドレスを

割り当てられ、前記冗長ミドルウェア・スタックの前記複数の層のうちの第１の選択されたものがアクティブであって、前記冗長ミドルウェア・スタックの前記複数の層のうちの第２の選択されたものが待機状態である、段階と、

前記１次ノードと２次ノードの間で、前記アクティブなミドルウェア・スタックである、前記複数の層の第１の選択されたものに対してアクセス可能なデータをもつデータ複製区画を共有する段階と、

前記１次ノードのみが前記データ複製区画にアクセスできるように前記データ複製区画をマウントし、エラーを示す状況に応答して、前記１次ノードから前記２次ノードに前記仮想ＩＰアドレスを転送することによって前記特定の高可用性システムを調整するように対応し、前記１次ノードに対して電源を切り、前記２次ノードのみアクセスできるように前記データ複製区画を再マウントし、前記データ複製区画内のデータに対するアクセスを要求する前記冗長ミドルウェア・スタックの前記複数の層のうちの第２の選択されたものを活動化させる段階と、

、前記エラーと、前記複数のコンポーネントの状態を、前記複数の高可用性システムの各々から受け取った別のレポートに基づき前記複数の高可用性システムを管理するようにイネーブルされた前記遠隔エンタープライズ・サーバに報告する段階を有し、

前記仮想ＩＰアドレスは、前記１次ノードと前記２次ノードのうち、一度に１つのノードのみが利用可能である、

方法。

【請求項８】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項７に記載の方法であって、各々が、Ｊ２ＥＥ準拠ミドルウェア・スタックを実装する複数のサーバを有する、前記１次ノード及び前記２次ノードを監視することによって、前記コンポーネントの状況を監視する段階をさらに含む方法。

【請求項９】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項７に記載の方法であって、前記高可用性システムの１次ノードのステータスをハートビート・モニタによって監視する段階をさらに含む方法。

【請求項１０】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項７に記載の方法であって、前記高可用性システムの前記１次ノードの前記ミドルウェア層によって提供されるサービスのステータスを、サービス監視デーモンによって検出する段階をさらに含む方法。

【請求項１１】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項７に記載の方法であって、構成要求を前記遠隔エンタープライズ・サーバから受信し、

将来のエラーに応答して前記高可用性システムを調整するために前記クラスタ管理コントローラが対応する手法の構成を調整する、ことをさらに含む方法。

【請求項１２】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項７に記載の方法であって、

構成要求を前記遠隔エンタープライズ・サーバから受信し、

前記要求に従って前記高可用性システムのハードウェア構成を調整する、

ことをさらに含む方法。

【請求項１３】

コンピュータ読み取り可能媒体に常駐し、高可用性システムの遠隔エンタープライズ管理を可能にするためのコンピュータ・プログラムであって、

ネットワークを介して遠隔エンタープライズ・サーバに通信接続された、複数の高可用性システムうちの特定の高可用性システムの少なくとも１つのコンポーネントのステータスを監視する手段と、

前記特定の高可用性システム内の１次ノードで、ウェブ・アプリケーションをサポートするためのミドルウェア・スタックを走らせる手段であって、前記１次ノードが第１のＩＰアドレスを割り当てられ、前記１次ノードはさらに、該第１のＩＰアドレスとは異なり、要求を向けるための仮想ＩＰアドレスを割り当てられる、手段と、

前記特定の高可用性システム内の２次ノードで、前記１次ノードの前記ミドルウェア・スタックの複数の層をミラーリングするための冗長ミドルウェア・スタックを走らせる手段であって、前記２次ノードが前記第１のＩＰアドレス及び前記仮想ＩＰアドレスとは異なる第２のＩＰアドレスを割り当てられ、前記冗長ミドルウェア・スタックの前記複数の層のうちの第１の選択されたものがアクティブであって、前記冗長ミドルウェア・スタックの前記複数の層のうちの第２の選択されたものが待機状態である、手段と、

前記１次ノードと２次ノードの間で、前記アクティブなミドルウェア・スタックである、前記複数の層の第１の選択されたものに対してアクセス可能なデータをもつデータ複製区画を共有する手段と、

前記１次ノードのみが前記データ複製区画にアクセスできるように前記データ複製区画をマウントし、エラーを示す状況に応答して、前記１次ノードから前記２次ノードに前記仮想ＩＰアドレスを転送することによって前記特定の高可用性システムを調整するように対応し、前記１次ノードに対して電源を切り、前記２次ノードによるアクセスのため前記データ複製区画を再マウントし、前記データ複製区画内のデータに対するアクセスを要求する前記冗長ミドルウェア・スタックの前記複数の層のうちの第２の選択されたものを活動化させる手段と、

前記エラーの時点で前記高可用性システムの複数のコンポーネントの状態を検出する手段と、

、前記エラーと、前記複数のコンポーネントの状態を、前記複数の高可用性システムの各々から受け取った別のレポートに基づき前記複数の高可用性システムを管理するようにイネーブルされた前記遠隔エンタープライズ・サーバに報告する手段を有し、

前記仮想ＩＰアドレスは、前記１次ノードと前記２次ノードのうち、一度に１つのノードのみが利用可能である、

コンピュータ・プログラム。

【請求項１４】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項１３に記載のコンピュータ・プログラムであって、各々がＪ２ＥＥ準拠ミドルウェア・スタックを実装する複数のサーバをもつ１次ノード及び２次ノードを監視することによって、前記コンポーネントのステータスを監視するための手段、

をさらに有するコンピュータ・プログラム。

【請求項１５】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項１３に記載のコンピュータ・プログラムであって、

前記高可用性システムの１次ノードのステータスをハートビート・モニタによって監視するための手段、

をさらに有するコンピュータ・プログラム。

【請求項１６】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項１３に記載のコンピュータ・プログラムであって、

前記特定の高可用性システムの前記１次ノードの前記ミドルウェア層によって提供されるサービスのステータスを、サービス監視デーモンによって検出するための手段、

をさらに有するコンピュータ・プログラム。

【請求項１７】

高可用性システムの遠隔エンタープライズ管理を可能にするための請求項１３に記載のコンピュータ・プログラムであって、

構成要求を前記遠隔エンタープライズ・サーバから受信するための手段と、

前記高可用性システムを調整するために前記クラスタ管理コントローラが対応する手法の構成を調整するための手段と、
をさらに有するコンピュータ・プログラム。
【請求項１８】
高可用性システムの遠隔エンタープライズ管理を可能にするための請求項１３に記載のコンピュータ・プログラムであって、
構成要求を前記遠隔エンタープライズ・サーバから受信するための手段と、
前記要求に従って前記特定の高可用性システムのハードウェア構成を調整するための手段と、
をさらに有するコンピュータ・プログラム。
【請求項１９】
複数の高可用性システムを遠隔的に構成するためのシステムであって、
ネットワークを介して遠隔エンタープライズ・サーバに通信接続された複数の高可用性システムのうちの特定の高可用性システムを有し、
前記高可用性システムの各々が、更に、
ウェブ・アプリケーションをサポートするためのミドルウェア・スタックを走らせる１次ノードであって、該ミドルウェア・スタックの複数の層が活動的であり、前記１次ノードが第１のＩＰアドレスを割り当てられ、さらに、該第１のＩＰアドレスとは異なり、要求を向けるための仮想ＩＰアドレスを割り当てられる前記１次ノードと、
前記１次ノードの前記ミドルウェア・スタックの複数の層をミラーリングするための冗長ミドルウェア・スタックを走らせる２次ノードであって、前記冗長ミドルウェア・スタックの前記複数の層のうちの第１の選択されたものがアクティブであって、前記第１のＩＰアドレス及び前記仮想ＩＰアドレスとは異なる第２のＩＰアドレスを割り当てられ、前記冗長ミドルウェア・スタックの前記複数の層のうちの第２の選択されたものが待機状態である、前記２次ノードと、
前記１次ノードと、前記アクティブ・ミドルウェア・スタックである、前記複数の層の第１の選択されたものに対してアクセス可能なデータをもつ２次ノードの間で、共有されるデータ複製区画と、
前記１次ノードのみが前記データ複製区画にアクセスできるように前記データ複製区画をマウントし、前記高可用性システムの前記１次ノードの状況を監視し、該状況がエラーを示すことに応答して、前記仮想ＩＰアドレスを前記第１のノードから前記第２のノードに転送し、前記１次ノードに対して電源を切り、前記２次ノードのみがアクセスできるように前記データ複製区画を再マウントし、前記データ複製区画内の前記データに対するアクセスを要求する前記冗長ミドルウェア・スタックの前記複数の層のうちの第２の選択されたものを活動化させる、クラスタ管理コントローラと、
前記ステータスがエラーを示すのと同時に、前記複数の高可用性システムの各々の個別のものの複数のコンポーネントについての監視された情報を検出するための監視コントローラと、
前記ネットワークに通信接続された、遠隔エンタープライズ・サーバであって、前記個別の監視コントローラから前記複数の高可用性システムの各々についての前記監視された情報を受け取り、前記監視された情報を解析し、再構成によって調整可能なエラーを示す監視された情報を提出する、前記高可用性システムの各々に対して再構成の要求を送る前記遠隔エンタープライズ・サーバとを有し、
前記仮想ＩＰアドレスは、前記１次ノードと前記２次ノードのうち、一度に１つのノードのみが利用可能である、
システム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般に、改善された高可用性（ｈｉｇｈ　ａｖａｉｌａｂｉｌｉｔｙ）クラ

スタ管理に関し、特に、高可用性システムの遠隔クラスタ管理に関する。より具体的には、本発明は、エンタープライズ・ネットワークにおける多数の高可用性システムの改善された遠隔監視及び管理に関する。

【背景技術】

【０００２】

本発明は、以下の同時継続出願に関連する。

（１）２００４年１月２０日出願の米国特許出願第１０／７６１１６３号

負荷及び需要が常に変動し、各々の顧客の要求を処理することが最重要事項である小売業、銀行、及び他のオンライン・サービスのために、ミッションクリティカルな作動を取扱う高可用性（ＨＡ）システムが開発されてきた。一般に、ＨＡシステムは、ネットワーク・システムのコンポーネントの計画停止又は計画外停止のいずれかを原因とするサービスの損失をなくすか又は最小限にするように設計されたシステムである。ＨＡシステムを提供する主要な方法は、サーバのクラスタにグループ化された冗長化ハードウェア及びソフトウェア・コンポーネントによるものである。

【０００３】

ＨＡシステムにおいては、クラスタの１つのノードに故障が発生したとき、システムが１つのノードによって行われる処理を別のノードに移すようになっているため、冗長性が重要である。例えば、２ノードのＨＡクラスタにおいては、一方のノードは、典型的には１次ノードとして指定され、他方のノードは、典型的にはバックアップ・ノードとして指定される。一般に、１次ノードは、クラスタが起動されたときに最初にアプリケーションを稼動させる。さらに、一般に、バックアップ・ノードが指定され、１次ノードが故障した場合にそのアプリケーションを稼動させる。ＨＡクラスタ・システムは、典型的には、１次ノードのポーリング（すなわち、ハートビートの検査）を定期的に行って該ノードが引き続きアクティブ状態にあるかどうかを判断するクラスタ管理処理を実施するものである。「ハートビート」が検出されない場合には、クラスタ・マネージャが、ソフトウェア処理をクラスタ内の別のサーバに移動させる。

【０００４】

ＨＡシステムの重要な特徴は、回復時間にある。一般に、ＨＡシステムの回復時間は、バックアップ・ノードが故障した１次ノードからアプリケーションを引き継ぐのにかかる時間である。顧客が迅速に取引を完了できない場合には、小売業者は重要なビジネスを失うことがあるため、回復時間は、ＨＡシステムを基盤とする販売では特に重要である。回復時間の３０秒の遅れでさえ、小売業者の商取引を減少させるものとなる。

【０００５】

ＨＡシステムの別の重要な特徴は、フェイルオーバーの際にデータをほとんど又はまったく喪失しないことにある。特に、受託データをほとんど又はまったく喪失しないことが重要である。例えば、フェイルオーバーの際に、顧客の注文又は顧客の情報に関する貴重な情報を喪失することは不都合なことである。

【０００６】

回復時間を短くし、フェイルオーバーの際にデータをほとんど又はまったく喪失しないようにするためには、まず、ＨＡシステムの構築といった方法でハードウェア及びソフトウェアを組み合わせることが重要である。しかしながら、ＨＡシステムを起動した後は、該ＨＡシステムの構成を監視して調整し、フェイルオーバー効率及び他のエラーの修正効率の改善を試みることが重要である。

【０００７】

ハードウェア及びソフトウェアをＨＡシステムとして構成するときは、多くの開発者は、新たなハードウェアを必要とすることが多い顧客環境でアプリケーションを制御するために、カスタマイズされたＨＡソフトウェア・サービスを開発してきた。これらの解決策は、高価なものとなることが多く、多数のプラットフォーム間でのアプリケーションの移植を可能にするオープン・ソース技術の利点を生かすものではない。さらに、サーバ・システムで利用可能な能力がフェイルオーバー効率を自動的に向上させることを期待して、

多くの場合、高価なサーバ・システムが選択される。

【０００８】

別の方法として、オープン・ソース開発者は、ＨＡシステムを実装する際に構成できる機能を用いて、オープン・ソース技術を拡張し続ける。例えば、Ｌｉｎｕｘは、プラットフォームに依存しない安価なオペレーティング・システムを提供する。Ｌｉｎｕｘの開発者は、他の開発者がオープン・ソース方式で実装することができるオペレーティング・システムに、機能を追加し続ける。「ハートビート」及びディストリビューテッド・リプリケイテッド・ブロック・デバイス（ｄｒｂｄ）といったこれらの機能の幾つかは、Ｌｉｎｕｘオペレーティング・システムと共に実装されて、ＨＡシステムの構成を支援する。

【発明の開示】

【発明が解決しようとする課題】

【０００９】

Ｌｉｎｕｘツールは、故障を監視し、ＨＡシステムで用いられるハードウェアを構成するためのフレームワークを提供するが、付加的な監視及び構成機能についての必要性がある。具体的には、ＨＡシステムのハードウェア及びソフトウェアの両方における故障、エラー、及び他の非理想的な状態を監視し、オープン・ソースのＨＡツールが故障及びエラーを検出したことを監視する方法についての必要性がある。さらに、監視されたシステム・ステータスを遠隔的に収集し、ＨＡシステムの遠隔的な再構成を容易にする必要性がある。

【００１０】

さらに、典型的には、多数のＨＡシステムをネットワーク内で組み合わせて、エンタープライズ・システムを形成する。各々のＨＡシステムは、例えば、エンタープライズ・システム内部の異なる記憶装置についてのトランザクション要求を処理することができる。遠隔的に、エンタープライズ・システム内部の多数のＨＡシステムの監視されたシステム・ステータスを収集し、該システム・ステータスを性能要件と比較し、該エンタープライズ・システム内部の各ＨＡシステムのハードウェア及びソフトウェアのニーズを突き止めるための方法、システム、及びプログラムについての必要性がある。

【００１１】

さらに、オープン・ソースのオペレーティング・システム・フレームワークを用いてＨＡシステムを実装するときは、オープン・ソース対応のミドルウェア層を実装してトランザクション要求を処理することが有利になる。具体的には、（１）遠隔エンタープライズ・コンソールとインターフェース接続するオープン・ソースに基づくクラスタ管理によって制御され、（２）エンタープライズ・ネットワーク内の多数のＨＡシステムを監視し、構成することができる、Ｊａｖａ（商標）２プラットフォーム、エンタープライズ・エディション（Ｊ２ＥＥ）準拠ミドルウェア・スタックを実装することが有利になる。

【課題を解決するための手段】

【００１２】

本発明は、改善された高可用性クラスタ管理を提供するものであり、具体的には、オープン・ソース・フレームワークに従って実装される高可用性システムの遠隔クラスタ管理を提供するものである。さらにより具体的には、本発明は、エンタープライズ・ネットワークにおける多数の高可用性システムの改善された遠隔監視及び管理に関するものである。

【００１３】

本発明の一態様によると、多数の高可用性システムがエンタープライズ内でネットワーク化され、遠隔エンタープライズ・サーバによって全体的に管理される。各々の高可用性システム内部では、クラスタ管理コントローラが、高可用性システムの特定のコンポーネントのステータスを監視し、そのステータスがエラーを示したときには該高可用性システムを調整するように対応する。さらに、各々の高可用性システムでは、監視コントローラが、クラスタ管理コントローラが特定のコンポーネントのステータスに対応した時を検出し、該高可用性システムの多数のコンポーネントの状態を検出する。次いで、監視コント

ローラは、エラー及びコンポーネントの状態を遠隔エンタープライズ・サーバに報告する。遠隔エンタープライズ・サーバは、その報告に基づいて、高可用性システムを管理することができる。

【００１４】

具体的には、高可用性サーバは、ハートビート・モニタ及びサービス監視デーモンなどのオープン・ソース機能によって監視されるＪ２ＥＥ準拠ミドルウェア・スタックを実装する。ハートビート・モニタは、具体的には、ミドルウェア・スタックが常駐する特定のサーバのステータスを検出する。サービス監視デーモンは、具体的には、ミドルウェア・スタックによって提供されるサービスの特定のインスタンスのステータスを検出する。

【００１５】

遠隔エンタープライズ・サーバは、構成変更が行われるべきであることを報告から判断し、構成要求を高可用性システムに送信する。次に、監視コントローラは、ハートビート・モニタ又はサービス監視デーモンがエラーを検出し、エラーに対して対応する方法を調整するために、高可用性システムの構成を調整する。さらに、高可用性システム内部の他のハードウェア及びソフトウェア・コンポーネントは、監視コントローラによって再構成することができる。

【００１６】

遠隔エンタープライズ・サーバは、各々の高可用性システムに関する監視情報をデータベースに格納することが好ましい。さらに、エンタープライズ・サーバは、監視情報を分析し、どの高可用性システムが性能要件を満足していないかを判断することが好ましい。エンタープライズ・サーバは、ハードウェア及びソフトウェアの変更並びに構成の変更を推奨することができる。さらに、エンタープライズ・サーバは、比較性能を表示し、高可用性システムの実時間表示と、エラーが各々のシステムで検出された時とを提供する。

【００１７】

発明の特性と考えられる新規な特徴が、添付の特許請求の範囲に記載される。しかしながら、発明自体並びに好ましい使用モード、そのさらなる目的及び利点は、以下の例示的な実施形態の詳細な説明を添付の図面と併せて読んだときに、最もよく理解されることになるであろう。

【発明を実施するための最良の形態】

【００１８】

ここで図面、具体的には図１を参照すると、本方法、システム、及びプログラムを実装することができるシステムの一実施形態が示されている。本発明は、各種のコンピュータ・システム、サーバ・システム、及びエンタープライズ・システムを含む様々なシステムで実施することができる。

【００１９】

コンピュータ・システム１００は、該コンピュータ・システム１００内部の情報を伝達するためのバス１２２又は他の通信装置と、情報を処理するために該バス１２２に結合された多数のプロセッサ１１２ａ～１１２ｎを含む。バス１２２は、ブリッジ及びアダプタによって接続され、多数のバス・コントローラによってコンピュータ・システム１００内部で制御される低待ち時間のパス及びより高待ち時間のパスを含むことが好ましい。

【００２０】

プロセッサ１１２ａ～１１２ｎは、通常作動の際に、ランダム・アクセス・メモリ（ＲＡＭ）１１４などの動的記憶装置、及びリード・オンリー・メモリ（ＲＯＭ）１１６などの静的記憶装置からアクセス可能なオペレーティング・システム及びアプリケーション・ソフトウェアの制御下でデータを処理する、ＩＢＭのＰｏｗｅｒＰＣ（商標）プロセッサなどの汎用プロセッサとすることができる。好ましい実施形態においては、多数のソフトウェア層は、プロセッサ１１２ａ～１１２ｎで実行されるときに、本明細書で説明される図７、図８、図９、図１１、図１２、図１３などのフローチャートに示される動作を行う機械実行可能命令を含む。代替的には、本発明のステップは、該ステップを実施するための配線論理回路を含む特定のハードウェア・コンポーネントか、又は、プログラムされた

コンピュータ・コンポーネントとカスタム・ハードウェア・コンポーネントとの任意の組み合わせによって実施することができる。

【００２１】

本発明は、本発明に係る処理を行うようにコンピュータ・システム１００をプログラミングするのに用いられる機械実行可能命令を格納した機械読み取り可能媒体に含ませて、コンピュータ・プログラムとして提供することができる。ここで用いられる「機械読み取り可能媒体」という用語は、実行命令をプロセッサ１１２ａ～１１２ｎ又はコンピュータ・システム１００の他のコンポーネントに与えることに関係する何らかの媒体を含む。こうした媒体は、不揮発性媒体、揮発性媒体、及び伝送媒体を含む多くの形態をとることができるが、これらに限定されるものではない。不揮発性媒体の一般的な形態として、例えば、フロッピー（登録商標）・ディスク、フレキシブル・ディスク、ハード・ディスク、磁気テープ又は他の何らかの磁気媒体、コンパクト・ディスクＲＯＭ（ＣＤ－ＲＯＭ）又は他の何らかの光媒体、パンチ・カード又は孔のパターンを備えた他の何らかの物理媒体、プログラマブルＲＯＭ（ＰＲＯＭ）、消去可能ＰＲＯＭ（ＥＰＲＯＭ）、電気的ＥＰＲＯＭ（ＥＥＰＲＯＭ）、フラッシュメモリ、他の何らかのメモリ・チップ又はカートリッジ、又は、コンピュータ・システム１００が読み取り可能で、命令を格納するのに適した他の何らかの媒体が挙げられる。本発明の実施形態においては、不揮発性媒体の例は、図示されるようなコンピュータ・システム１００の内部コンポーネントである大容量記憶装置１１８であるが、外部装置として構成できることも理解されるであろう。揮発性媒体として、ＲＡＭ１１４などの動的メモリが挙げられる。伝送媒体として、バス１２２を構成するワイヤを含む、同軸ケーブル、銅線、又は光ファイバが挙げられる。伝送媒体は、無線データ通信又は赤外線データ通信の際に生成される波などの音波又は光波の形態をとることもできる。

【００２２】

さらに、本発明は、コンピュータ・プログラムとしてダウンロードすることも可能であり、この場合、プログラム命令は、搬送波又は他の伝搬媒体に統合されたデータ信号として、サーバ１４０などの遠隔コンピュータから、バス１２２に結合された通信インターフェース１３２へのネットワーク・リンク１３４ａ～１３４ｎの１つを経由して、必要なコンピュータ・システム１００に転送することができる。通信インターフェース１３２は、例えばローカル・エリア・ネットワーク（ＬＡＮ）、広域ネットワーク（ＷＡＮ）に接続することができる多数のネットワーク・リンク１３４ａ～１３４ｎにつながる双方向データ通信を提供する。サーバ・システムとして実装されるときは、コンピュータ・システム１００は、通常、入力／出力コントローラに接続された多数のペリフェラル・コンポーネント・インターコネクト（ＰＣＩ）バス・ブリッジを経由してアクセス可能な多数の通信インターフェースを含む。このようにして、コンピュータ・システム１００は、多数のネットワーク・コンピュータへの接続が可能になる。

【００２３】

ネットワーク環境においては、コンピュータ・システム１００は、ネットワーク１０２を通して他のシステムと通信する。ネットワーク１０２は、伝送制御プロトコル（ＴＣＰ）及びインターネット・プロトコル（ＩＴ）などの特定のプロトコルを用いて相互に通信する世界中のネットワーク及びゲートウェイの集まりを参照することができる。ネットワーク１０２は、デジタル・データ・ストリームを搬送する電気信号、電磁信号、又は光信号を用いる。様々なネットワークを通り、ネットワーク・リンク１３４ａ～１３４ｎと通信インターフェース１３２とを通って、デジタル・データをコンピュータ・システム１００に又はそこから搬送する信号は、情報を運ぶ搬送波の例示的な形態である。図示されてはいないが、コンピュータ・システム１００には、通信を容易にする多数の周辺コンポーネントを含むこともできる。

【００２４】

コンピュータ・システム１００がサーバ・システムとしてＨＡクラスタに実装されるときは、他のサーバ・システムとの局所的な接続を支援するために、付加的なネットワーク

・アダプタを含むことができる。さらに、サーバ・システムとしてＨＡクラスタに実装されるときは、コンピュータ・システム１００は、ＩＢＭ社のｘＳｅｒｉｅｓ（商標）サーバなどの商用ハードウェア・サーバとして設計することができる。

【００２５】

当業者であれば、図１に示されるハードウェアは変更可能であることを認識するであろう。さらに、当業者であれば、図示された例は、本発明に関してアーキテクチャの限定を意味することを意図するものではないことを認識するであろう。

【００２６】

ここで図２を参照すると、フェイルオーバーの際にミドルウェアを効率的に移行させるための高可用性クラスタのハードウェア構成のブロック図が示されている。図示されるように、クライアント・システム２０２及び２０４が、サービス要求を転送するためにネットワークに接続される。この実施形態においては、クライアント・システム２０２及び２０４は、フェイルオーバーの際の回復時間を短くして受託データの損失を最小限にするように構成された高可用性（ＨＡ）システム２０８から、サービスを要求する。

【００２７】

図示されるように、ＨＡシステム２０８は、１次ノード２１０及び２次ノード２２０を含む。以下に説明するように、１次ノード２１０及び２次ノード２２０は、実行時には高可用性システムとなる冗長化ハードウェア及びソフトウェアを実装することが好ましい。具体的には、以下に説明するように、１次ノード２１０及び２次ノード２２０は、好ましい実施形態においてはＪ２ＥＥアプリケーションに対応する冗長化ミドルウェアを実装する。ミドルウェアとは、ウェブ・アプリケーション及びシステムを開発し、統合し、管理するソフトウェアである。以下に説明するように、ミドルウェアは、通信、処理、データの統合と、トランザクション能力管理及びシステム管理の自動化とを可能にする。

【００２８】

具体的には、Ｊａｖａ（商標）２プラットフォーム、エンタープライズ・エディション（Ｊ２ＥＥ）は、ウェブ・アプリケーションの作成に用いるための再利用可能なコンポーネント・モデルを提供する。Ｊ２ＥＥは、標準アプリケーション・モデルと、アプリケーションのホストとして動作するための標準プラットフォームと、互換性必要条件と、Ｊ２ＥＥプラットフォームのオペレーション定義とを定める。このオープン・ソース・モデルの利点は、多数の開発者が付加的なコンポーネント及び構成と共にＪ２ＥＥモデルを実装することが可能で、さらに、すべてのＪ２ＥＥアプリケーションがＪ２ＥＥベースのシステム上で稼動することである。

【００２９】

インターナショナル・ビジネス・マシーンズ・コーポレーション（ＩＢＭ（商標））の開発者は、Ｊ２ＥＥモデルを実装するソフトウェアを開発した。このソフトウェアは、多くの場合、Ｊ２ＥＥフレームワークでは規定されない隙間の部分を埋めるものである。例えば、具体的には、ＩＢＭ（商標）は、サーバのクラスタに実装されるとＪ２ＥＥアプリケーションに対応するＪ２ＥＥ準拠ソフトウェアであるミドルウェア・スタックを開発した。一般に、ミドルウェア・スタックとして、ウェブ・サーバ、データベース・サーバ、及びユニバーサル・インターネット・アプリケーション・サーバが挙げられる。具体的には、このスタックとして、ＩＢＭ　ＤＢ２（商標）ＵＤＢ　Ｅｎｔｅｒｐｒｉｓｅ　Ｅｄｉｔｉｏｎ、ＩＢＭ　ＨＴＴＰ　Ｓｅｒｖｅｒ、及び、ＩＢＭＷｅｂＳｐｈｅｒｅ（商標）Ａｐｐｌｉｃａｔｉｏｎ　Ｓｅｒｖｅｒなどの製品を挙げることができる。

【００３０】

さらに、１次ノード２１０及び２次ノード２２０は、ＨＡクラスタのＪ２ＥＥ準拠ミドルウェア・スタック及びハードウェアの故障及びエラーを監視する監視・構成コントローラを実装する。監視・構成コントローラの例として、Ｊ２ＥＥフレームワークで稼動するソフトウェアを監視するための隙間の部分を埋め、Ｊ２ＥＥフレームワークが稼動するシステムの構成を容易にする、Ｔｉｖｏｌｉ（商標）Ｍｏｎｉｔｏｒｉｎｇコントローラを実装することができる。

【００３１】
１次ノード２１０及び２次ノード２２０は、各々のノードが他のノードのハートビートを迅速に検査することを可能にする信頼性の高い簡単な方法で、接続される。実施形態においては、この接続は、各々のノードにおけるネットワーク・アダプタ間に接続されるクロスケーブル２１８によって可能になる。具体的には、クロスケーブル２１８は、ハートビート・データを転送するためのイーサネット（登録商標）接続を可能にすることが好ましい。代替的には、ハートビート・データは、クロスケーブル２１８が故障した場合に、ネットワーク１０２を経由して公衆ＩＰ接続間で転送することもできる。ハートビートの通信チャネルを１次ノード２１０と２次ノード２２０との間に提供するために他のハードウェアを実装しても良いこと、及び、ネットワークを基盤とする接続に加えてシリアル接続を実装しても良いことが理解されるであろう。

【００３２】
具体的には、ハートビート信号が１次ノード２１０と２次ノード２２０との間でクロスケーブル２１８を通して送られるとき、該ハートビートが機能しなくなった場合には、２次ノード２２０が、故障の前に１次ノード２１０によって提供されたサービスを引き継ぐことになる。しかしながら、以下に説明するように、本発明の利点によるとミドルウェア・コンポーネントは、ハートビート故障をさらに分析し、２次ノード２２０が１次ノード２１０によって提供されるサービスを引き継ぐ前に、該故障に関する付加的な情報を提供することができる。そのうえ、以下に説明するように、Ｌｉｎｕｘによるハートビート及びＬｉｎｕｘによらないハートビートは共に、クロスケーブル２１８を介して監視することができる。

【００３３】
１次ノード２１０及び２次ノード２２０は、データ記憶システム２１４及び２２４にアクセスする。有利なことに、ここではｄｒｂｄパーティション２３０として示されるデータ・リプリケータが、１次ノード２１０と２次ノード２２０との間で実際に物理的に共有される記憶装置を必要とせずに１次ノード及び２次ノードによってアクセス可能なデータを複製するために、データ記憶装置２１４及び２２４の各々のパーティションを含む。本発明の利点によると、ｄｒｂｄは、フェイルオーバーの際の１次ノード２１０から２次ノード２２０へのデータ転送を容易にするために、パーティション上で稼働するように構成される。本発明は、ｄｒｂｄスクリプトによって管理されるｄｒｂｄパーティションに関して説明されるが、他の分散型データ複製システムを実装しても良いことが理解されるであろう。

【００３４】
無停電電源装置（ＵＰＳ）２１２及びＵＰＳ２２２の各々は、それぞれ１次ノード２１０及び２次ノード２２０に独立電源を供給する。好ましくは、ＵＰＳ２１２と２次ノードとの間、及び、ＵＰＳ２２２と１次ノード２１０との間にも、接続が確立される。一実施形態においては、シリアルケーブル２１６が１次ノード２１０とＵＰＳ２２２との間に設けられ、シリアルケーブル２２６が２次ノード２２０とＵＰＳ２１２との間に設けられる。しかしながら、他の形式の接続ハードウェアを実装しても良いことが理解されるであろう。

【００３５】
本発明の利点によると、１次ノード２１０に故障が検出されたとき、２次ノード２２０は、事前に１次ノード２１０に向けられた要求をフェイルオーバー後に受信し始める。１次ノード２１０上で稼働しているハードウェア、ソフトウェア、又はネットワークの一部のみが故障することがあるため、該１次ノード２１０がフェイルオーバー後にデータを更新しないようにすることを保証する唯一の方法は、ＵＰＳ２１２の電源を切ることである。有利なことに、以下に説明するように、待機ノード２２０へのフェイルオーバーが検出されたときは、電源を切る命令を該待機ノード２２０からＵＰＳ２１２に送るために、本明細書でさらに詳細に説明されるＳＴＯＮＩＴＨがクラスタ・マネージャによって実行される。

【００３６】
ここで図３を参照すると、本発明の方法、システム、及びプログラムに係るクラスタ・マネージャのブロック図が示されている。図示されるように、クラスタ・マネージャ３２２には、効率的なフェイルオーバーを実行するのに利用される、ハートビート・ツール４０２と、ｄｒｂｄスクリプト４０４と、ｍｏｎ４０６と、ＳＴＯＮＩＴＨ機能４０８とを含む多数のコンポーネントが組み込まれる。クラスタの他の状況を管理するために、他のコンポーネントをクラスタ・マネージャに組み込んでも良いことが理解されるであろう。さらに、フェイルオーバーを管理するために、付加的なコンポーネントをクラスタ・マネージャ３２２に組み込んでも良いことが理解されるであろう。
【００３７】
ハートビート・ツール４０２は、Ｊ２ＥＥ準拠ハードウェア・スタックを用いるＨＡクラスタ内部のフェイルオーバーを管理するために構成される、Ｌｉｎｕｘ用のハートビート・パッケージを含むことが好ましい。具体的には、ハートビート・ツール４０２は、一般に、クラスタの２つのノード間で「ハートビート」要求を送信することによって作動する。図２で説明されたように、ハートビート要求は、各々のノードにおけるネットワーク・アダプタ間でクロスケーブルを通して送信することができる。サーバ・システムのクラスタ上で稼働するＪ２ＥＥ準拠ミドルウェア・スタックに使用されるときは、ハートビート・ツール４０２によって送信されるハートビート要求は、スタックの異なる層全体に配信される。
【００３８】
ハートビート要求が戻されない場合には、２次ノードは、１次ノードが故障したものと想定し、該１次ノード上で稼働していたＩＰ、データ、及びサービスを引き継ぐことができる。２次ノードが、１次ノード上で稼働していたＩＰ、データ、及びサービスを引き継ぐときは、ハートビート・ツール４０２は、待機モードで待機している該２次ノードのコンポーネントを起動し、ＩＰアドレスを該２次ノードのコンポーネントに割り当て、他のフェイルオーバー・タスクを実行する。
【００３９】
ｄｒｂｄ４０４は、フェイルオーバーの際のデータ切り替えを改善するための、ＨＡクラスタのデータを管理する関連スクリプトをもつカーネル・モジュールである。この切り替えは、ｄｒｂｄ４０４によって管理されるブロック・デバイスをミラーリングすることにより行われる。ｄｒｂｄは、ｄｒｂｄモジュールを読み込み、ＨＡクラスタの関連システム及び共有記憶装置のＩＰアドレスを用いて構成するスクリプトである。
【００４０】
Ｊ２ＥＥ準拠ミドルウェア・スタックに使用されるときは、ｄｒｂｄ管理ブロック・デバイスは、該ミドルウェア・スタックが稼働することができる記憶装置を提供する。当初は、１次ノードのみがｄｒｂｄ管理ブロック・デバイスから読み込むか又は書き込むことができるように、クラスタが構成され、ｄｒｂｄパーティションがマウントされる。フェイルオーバーが発生したときは、２次ノードのみがｄｒｂｄ管理ブロック・デバイスから読み込み／書き込みができるようにｄｒｂｄパーティションをマウントするために、ｄｒｂｄ４０４のデータディスク・スクリプトがハートビート・ツール４０２によって実行される。
【００４１】
ｍｏｎ４０６は、Ｊ２ＥＥ準拠ミドルウェア・スタック内部の重要なシステム・サービスを監視する監視スクリプトを定期的に実行するサービス監視デーモンである。サービスが機能しなくなったか又は異常終了したことが発見された場合には、ｍｏｎ４０６は、該サービスを再開させて、ミドルウェア・スタックのすべてのコンポーネントが１次サービスの範囲内で引き続き稼働し続けることを保証する。異常終了は、例えば、プログラミング・エラー、又は、ＲＡＭを原因とする一時的なクリティカル・リソース制約などの突発的なオペレーティング・システム事象によって発生することがある。具体的には、ｍｏｎがサービスを再開させるときは、該ｍｏｎは、休止サービスとはプロセス識別子（ＰＩＤ

）は異なるが、同一の仮想ＩＰアドレスをもつサービスの新たなインスタンスを再開させる。

【００４２】

ＳＴＯＮＩＴＨ４０８は、フェイルオーバーの際のデータ保全を保証するために、ハートビート・ツール４０２によって呼び出される機能である。具体的には、ＳＴＯＮＩＴＨ４０８は、図２に示されるように、ＵＰＳ２１２及び２２２へのシリアルケーブル構成を含む。ハートビート・ツール４０２がＳＴＯＮＩＴＨ４０８を呼び出すとき、その呼び出しは、シャットダウンさせるノードを指定する。ＳＴＯＮＩＴＨは、信号を送信して、要求されたＵＰＳの電源を切る。

【００４３】

監視・構成コントローラ４１０は、ＨＡクラスタ内部のハードウェア及びソフトウェアのステータスを監視するように指定された多数の監視コントローラを含む。本発明の利点によると、ＨＡクラスタの多数のハードウェア及びソフトウェア・コンポーネントに関するステータス情報は、遠隔集中エンタープライズ・コンソールに送られる。好ましくは、監視・構成コントローラ４１０は、Ｊａｖａ（商標）Ｍａｎａｇｅｍｅｎｔ Ｅｘｔｅｎｓｉｏｎｓ（ＪＭＸ）を補完して、ＨＡクラスタのハードウェア及びソフトウェア・コンポーネントを監視し、障害及び潜在的な問題を検出し、該クラスタを自動的に危機的な状態から回復させる。一実施形態においては、監視コントローラは、監視情報をＴｉｖｏｌｉ（商標）Ｅｎｔｅｒｐｒｉｓｅ Ｃｏｎｓｏｌｅ（ＴＥＣ）に送るＴｉｖｏｌｉ（商標）Ｍｏｎｉｔｏｒｉｎｇによって使用可能にされる。

【００４４】

具体的には、ハートビート・ツール４０２及びｍｏｎ４０６が、ノード内部の特定のコンポーネント及びサービスの特定のインスタンスを監視する一方で、監視・構成コントローラ４１０は、これらのツールによって監視された状態を検出し、フェイルオーバーを開始するようにハートビート・ツール４０２が起動されるか又はサーバを再始動させるようにｍｏｎ４０６が起動されるときには、システムの全ステータスを検出する。このように、監視・構成コントローラ４１０は、故障、エラー、及び非理想的な状態が発生したときに、ノードの多数のコンポーネントのステータスを収集することによって、オープン・ソース・ツールを補完するものである。

【００４５】

本発明の１つの利点として、遠隔集中監視コンソールは、集められた情報を用いて構成変更を判断することができる。具体的には、本発明の利点によると、監視・構成コントローラ４１０の監視コントローラの各々は、ＨＡクラスタの各々のハードウェア・コンポーネントと、Ｊ２ＥＥ準拠ミドルウェア・スタックの層の各々とを監視するように構成される。このように、コンソールは、ハードウェア及びミドルウェア層に関する監視情報に基づいて、どのミドルウェア層が、要求をキャッシュするためにより多くのメモリを必要とするか、要求を処理するためにより多くのスレッドを必要とするか、又は他の幾つかの方法で再構成されることを必要とするかを判断することができる。コンソールは、構成変更を監視・構成コントローラ４１０の構成コントローラに送信することができ、次いで、この構成コントローラは、ＨＡクラスタの構成を調整する。一実施形態においては、構成コントローラは、ＨＡクラスタの構成特性を管理するＴｉｖｏｌｉ（商標）Ｃｏｎｆｉｇｕｒａｔｉｏｎ Ｍａｎａｇｅｒである。

【００４６】

本発明の別の利点として、エンタープライズ・システムにおいては、コンソールは、集められた情報を用いて、どのＨＡクラスタがハードウェア及びソフトウェアの更新を必要とするかを判断する。例えば、監視情報について、コンソールは、どの記憶装置が故障している様子の交換が必要なハードウェアを有するか、どの記憶装置が最大容量に達して更新が必要なハードウェアを有するか、及び、どの記憶装置が機能していないか又は確実に稼働していないソフトウェアを有するかを判断することができる。

【００４７】

本発明のさらに別の利点として、監視・構成コントローラ４１０は、クラスタ・マネージャ３２２内部の他の監視コンポーネントと情報交換を行って、コンソールに送られるステータス情報を収集する。例えば、ｍｏｎ４０６が、監視状態にあるサービスのいずれかの故障を検出したときは、監視・構成コントローラ４１０は、システムにおける故障の全体像を収集することができるように、通知を遠隔集中監視コンソールに送信する。さらに、ハートビート・ツール４０２がシステムの１つのノードから別のノードへのフェイルオーバーを開始したときは、監視・構成コントローラ４１０は、ノード故障の統計情報を収集することができるように、通知を遠隔集中監視コンソールに送信する。

【００４８】

ここで図４を参照すると、本発明の方法、システム、及びプログラムに係る、フェイルオーバー前のＨＡクラスタにおけるソフトウェア構成の一実施形態のブロック図が示されている。図示されるように、１次ノード２１０及び２次ノード２２０はサーバ・システムのクラスタを表し、各々のクラスタはＩＰアドレスに割り当てられる。

【００４９】

本発明の利点として、クラスタ・マネージャ３２２は、１次ノード２１０及び２次ノード２２０上で稼働して、故障を監視し、サービスを再開し、故障が検出されたときにはフェイルオーバーを制御する。図示されるように、クラスタ・マネージャ３２２は、１次ノード２１０と２次ノード２２０との間で共有される記憶装置上に配置されるｄｒｂｄパーティション２３０を構成する。

【００５０】

１次ノード２１０は、ミドルウェア・スタックのすべてのアクティブなコンポーネント、すなわち、ロード・バランサ３１２、ＨＴＴＰサーバ３１４、ウェブ・アプリケーション・サーバ（ＷＡＳ）３１６、メッセージング・コントローラ３１８、及びデータベース・サーバ３２０を含む。２次ノード２２０は、アクティブなＨＴＴＰサーバ３３４及びＷＡＳ３３６を含むが、ロード・バランサ３３２、メッセージング・コントローラ３３８、及びデータベース３４０は、待機モードである。

【００５１】

ロード・バランサ３１２及び３３２は、クラスタ化することもできるＨＴＴＰサーバとＷＡＳサーバとの間で要求の負荷を平準化することが好ましい。好ましくは、ロード・バランサ３１２及び３３２は、サーバの可用性、容量、作業負荷、及び他の基準を用いて、インテリジェント負荷平準化を行う。一実施形態によると、ロード・バランサ３１２及び３３２は、ＩＢＭ　ＷｅｂＳｐｈｅｒｅ（商標）Ｅｄｇｅ　Ｓｅｒｖｅｒを介して実装することができる。

【００５２】

図示されるように、ロード・バランサ３１２及び３３２は、Ｌｉｎｕｘベースのハートビートとは無関係のハートビートを実装することができる。代替的には、Ｌｉｎｕｘベースのハートビート・モニタ３３２及び３４２が、ロード・バランサ３１２及び３３２のステータスを監視するようにしてもよい。

【００５３】

ＨＴＴＰサーバ３１４及び３３４は、ＨＴＴＰ要求を受信し、ＨＴＴＰ要求をそれぞれＷＡＳ３１６及び３３６の間で分散させるように設計されたサーバのクラスタを含むことができる。さらに、ＨＴＴＰサーバ３１４及び３３４は、サーブレット及びＥＪＢについての要求などの他の要求を受信したときに、サーブレット・コンテナ及びＥｎｔｅｒｐｒｉｓｅ　Ｊａｖａ（商標）Ｂｅａｎ（ＥＪＢ）コンテナなどのイネーブラを呼び出すことができるようにされる。一実施形態においては、ＨＴＴＰサーバ３１４及び３３４は、ＩＢＭのＷｅｂＳｐｈｅｒｅ（商標）、特にＷｅｂＳｐｈｅｒｅ（商標）ｖ．５．０に組み込まれたＨＴＴＰサーバを介して実装することができる。ＷｅｂＳｐｈｅｒｅ（商標）コンポーネントの多数の複製を１か所から制御することができるため、ＷｅｂＳｐｈｅｒｅ（商標）ｖ．５．０は有利である。このように、構成変更は、多数のサーバ・システムに配置されたソフトウェア・コンポーネントの多数のインスタンスを生じさせる１つの場所

で行うことができる。

【００５４】

本発明の利点として、ＨＴＴＰサーバ３１４及び３３４は、１次ノードが稼働状態になった後にクラスタ・マネージャ３２２のハートビート・ツールがＨＴＴＰサーバをアクティブにする、アクティブ／アクティブ構成で稼働させられる。ＨＴＴＰサーバ３１４及び３３４をアクティブ／アクティブ構成で稼働させることによって、要求負荷を２つの（又は、それ以上の）サーバにわたって分割して、クライアント要求を処理する速度を速めることができる。さらに、ＨＴＴＰサーバ３１４及び３３４をアクティブ／アクティブ構成で稼働させることによって、フェイルオーバー時の起動時間が短くなる。

【００５５】

ＷＡＳ３１６及び３３６は、ミッションクリティカルなサービスを顧客に提供するウェブ・アプリケーションに対応可能なサーバのクラスタを含み、具体的には、これらのサーバは、Ｊ２ＥＥアプリケーションに対応可能である。一実施形態によると、ＷＡＳ３１６及び３３６は、Ｊ２ＥＥアプリケーション及びサービスに対応するために必要なサーブレット、ＥＪＢ、及び他のＪ２ＥＥコンポーネントのホストとなる、ＩＢＭのＷｅｂＳｐｈｅｒｅ（商標）５．０によってサポートされるＷｅｂＳｐｈｅｒｅ（商標）Ａｐｐｌｉｃａｔｉｏｎ　Ｓｅｒｖｅｒである。

【００５６】

ＷＡＳ３１６は、メッセージング・コントローラ３１８及びデータベース・サーバ３２０と情報交換を行って、メッセージング・コントローラ及びデータベースと一体化したアプリケーション・サーバ機能を提供する。本発明の利点として、ＷＡＳ３１６及びＷＡＳ３３６は、アクティブ／アクティブ構成で稼働させられる。具体的には、システムを初期化するときは、メッセージング・コントローラ３１８及びデータベース・サーバ３２０が利用可能になると、クラスタ・マネージャ３２２のハートビート・ツールは、ＷＡＳ３３６を立ち上げて、アクティブ／アクティブ構成を作り出す。アクティブ／アクティブ構成で稼働させることによって、要求負荷をシステムの多数のクラスタにわたって分割して、クライアント要求を処理する速度を速めることができる。さらに、アクティブ／アクティブ構成で稼働させることによって、フェイルオーバー時の起動時間が短くなる。

【００５７】

メッセージング・コントローラ３１８及び３３８は、非同期要求を聴取し、これらの要求をローカル・キューに格納して、Ｊ２ＥＥベースのシステムと通信するキューを提供するためのコントローラを含む。メッセージング・コントローラ３１８及び３３８は、ＩＢＭ　ＭＱＳｅｒｉｅｓ（商標）、ＩＢＭ　ＷｅｂＳｐｈｅｒｅ（商標）ＭＱ、又はＪａｖａ（商標）Ｍｅｓｓａｇｉｎｇ　Ｓｅｒｖｉｃｅ（ＪＭＳ）を補完する他のメッセージ・コントローラを実装することができる。

【００５８】

本発明の利点として、メッセージング・コントローラ３１８及び３３８は、クラスタ・マネージャ３２２のｄｒｂｄが永続的リソースをｄｒｂｄパーティション２３０のメッセージング・キューで管理し、クラスタ・マネージャ３２２のハートビート・ツールがフェイルオーバー時にメッセージング・コントローラ３３８の起動を管理する、アクティブ／待機構成で稼働する。

【００５９】

データベース・サーバ３２０及び３４０は、永続的記憶装置を制御する。データベース・サーバ３２０及び３４０は、ＩＢＭ　ＤＢ２　ＵＤＢ　Ｅｎｔｅｒｐｒｉｓｅ　Ｅｄｉｔｉｏｎ又は他のリレーショナル・データベース管理システムなどのデータベース制御システムを介して実装することができる。

【００６０】

本発明の利点として、データベース・サーバ３２０及び３４０は、クラスタ・マネージャ３２２のｄｒｂｄが永続的リソースをｄｒｂｄパーティション２３０のデータベースで管理し、クラスタ・マネージャ３２２のハートビート・ツールがフェイルオーバー時にデ

ータベース・サーバ３４０の起動を管理する、アクティブ／待機構成で稼働する。

【００６１】

メッセージング・コントローラ３１８及び３３８並びにデータベース・サーバ３２０及び３４０が、アクティブ／待機構成で稼働し、最小限のデータ損失で迅速にフェイルオーバーを行うために、メッセージング・コントローラ３１８及びデータベース・サーバ３２０は、キュー及びデータベースを格納するためのルートとしてｄｒｂｄパーティション２３０がマウントされる位置を指定するように構成される。さらに、クラスタ・マネージャ３２２は、メッセージング・コントローラ３１８及びデータベース・サーバ３２０の仮想ＩＰアドレスを用いて、ｄｒｂｄ及びハートビート・ツールを構成する。

【００６２】

さらに、本発明の利点として、クラスタ・マネージャ３２２のｍｏｎ機能は、メッセージング・コントローラ３１８及びデータベース・サーバ３２０によって提供されるサービスなどの重要なシステム・サービスを監視する監視スクリプトを定期的に実行する。サービスが機能しなくなったか又は異常終了したことが発見された場合には、ｍｏｎは、該サービスを再開させて、ミドルウェア・スタックのすべてのコンポーネントが１次サービスの範囲内で引き続き稼働し続けることを保証する。

【００６３】

効率的なフェイルオーバーを達成するようにミドルウェアの各々の階層を構成し、クラスタ・マネージャ３２２を通じてミドルウェアの各々の階層を制御する方法は、他のタイプのミドルウェアに応用できることに注目することが重要である。このように、Ｊ２ＥＥ互換ミドルウェア・ソフトウェア・スタックから利用可能な機能は拡張し続けるので、各々のミドルウェア・コンポーネントを、アクティブ／アクティブ構成又はアクティブ／待機構成のいずれかで構成し、クラスタ・マネージャ３２２によって監視し、フェイルオーバーの際に制御することができる。

【００６４】

ここで図５を参照すると、本発明の方法、システム、及びプログラムに係る、フェイルオーバー後のＨＡクラスタにおけるソフトウェア構成の一実施形態のブロック図が示されている。図示されるように、フェイルオーバー後は、１次ノード２１０は、故障ノードとして表示される。２次ノード２２０は、すべてをアクティブなノードとして引き継ぐことになる。

【００６５】

故障が検出され、２次ノード２２０が１次ノード２１０を「休止」として指定するときには、ハードウェア及びソフトウェアの問題が存在する。具体的には、１次ノード２１０が、必要な時間内にハートビート要求に応答せず、その後間もなく作動可能になる場合がある。先に説明されたように、１次ノード２１０及び２次ノード２２０が共に作動可能になる状況を回避するために、クラスタ・マネージャ３２２のハートビート・ツールは、ＳＴＯＮＩＴＨを呼び出して、１次ノード２１０へのＵＰＳ電源を切ることになる。ＳＴＯＮＩＴＨによって制御できる安価なＵＰＳを実装することにより、データ保全を達成することが可能であり、１次ノードが実際には休止していないときに生じる可能性のあるＨＡの「スプリット・ブレイン」問題が回避される。

【００６６】

次に、フェイルオーバーの際に、ロード・バランサのハートビートは、ロード・バランサ３３２の起動を管理する。起動されると、クラスタ・マネージャ３２２のハートビート・ツールは、１次ノード２１０の仮想ＩＰ１アドレスをロード・バランサ３３２に割り当てる。その結果、仮想ＩＰアドレスに対する要求は、負荷平準化クラスタのＩＰアドレスの変更が生じないようにロード・バランサ３３２に転送される。

【００６７】

フェイルオーバーの際にはＨＴＴＰサーバ３３４及びＷＡＳ３３６はすでにアクティブなので、クラスタ・マネージャ３２２のハートビート・ツールは、これらのコンポーネントを起動させる必要はない。しかしながら、メッセージング・コントローラ３３８及びデ

ータベース・サーバ３４０は待機状態にあるので、クラスタ・マネージャ３２２のハートビート・ツールは、これらの層のフェイルオーバーを管理する必要がある。まず、ハートビート・ツールは、仮想ＩＰ２アドレスを引き継ぐ。次に、ハートビート・ツールは、ｄｒｂｄのデータディスク・サービスを開始して、ｄｒｂｄミラー・パーティションを構成し、マウントする。最後に、ハートビート・ツールは、仮想ＩＰ２アドレスが設定され、ミラーｄｒｂｄパーティション２３０上で立ち上がるメッセージ・キュー及びデータベース・インスタンスをもつ、メッセージ・コントローラ３３８及びデータベース・サーバ３４０を起動することになる。代替的に、図示されてはいないが、仮想ＩＰ２アドレスは１度に１つのノードのみが利用可能であるため、データベース・サーバ３４０は、待機状態ではなく、アクティブ・モードにしておいてもよい。データベース・サーバ３４０は、フェイルオーバー時には、要求が着信するまでｄｒｂｄパーティション２３０上のデータにアクセスしようとしないので、データベース・サーバ３４０には仮想ＩＰ２アドレスが設定され、ミラーｄｒｂｄパーティション２３０は要求が着信する前にアクセス可能である。逆に、メッセージング・コントローラ３３８などの幾つかの層は、起動時に直接データを読み込むものであり、フェイルオーバー前には２次ノード２２０はｄｒｂｄパーティション２３０上のデータを利用できないため、フェイルオーバー前に２次ノード２２０上で起動された場合には破壊されることになる。

【００６８】

ここで図６を参照すると、ＨＡシステムにおけるＪ２ＥＥ準拠ミドルウェア内部の独立系ソフトウェア・ベンダー（ＩＳＶ）アプリケーションの一実施例のブロック図が示されている。図示されるように、アクティブなＷＡＳ６０２と、アクティブなＩＢＭ　ＭＱＳｅｒｉｅｓ（商標）サーバ６１０と、アクティブなＩＢＭ　ＤＢ２サーバ６１４とが、ｄｒｂｄ６３０とインターフェース接続しているＪ２ＥＥ準拠ミドルウェア・スタックの１次ノードの一部を表している。参照数字６２０で示されるように、明細登録又はトランザクション完了が、アクティブなＷｅｂＳｐｈｅｒｅ（商標）Ａｐｐｌｉｃａｔｉｏｎ　Ｓｅｒｖｅｒ６０２で受信される。ＩＳＶは、特定のタイプの着信要求を処理するように、サーブレット又はＥＪＢをプログラムすることができる。例えば、参照数字６２０で示されるように、ルックアップ・サーブレット６０４は、項目をキャッシュ・レジスタで走査するときに、その価格を調べる価格ルックアップ（ＰＬＵ）を処理するＩＳＶウェブ・アプリケーションである。ルックアップ・サーブレット６０４は、次に、トランザクション・サーブレット６０８又は別のサーブレット若しくはＥＪＢといった別のコンポーネントによって非同期的に完了させられる保持トランザクションについての要求を送信する。しかしながら、まず、参照数字６２２で示されるように、その情報をＭＱリスナ６１２に送り、ＭＱキュー６３２上に置いて、次の着信要求を受信するようにルックアップ・サーブレット６０４を解放し、トランザクションがＭＱキュー６３２を介して順序良く厳密に１度だけ記録されることを保証する。次に、参照数字６２４で示されるように、ＭＤＢ６０６を呼び出して、トランザクションをＭＱキュー６３２から取り出し、参照数字６２６で示されるように該トランザクションをトランザクション・サーブレット６０８に送信する。最終的に、トランザクション・サーブレット６０８は、ＰＬＵを処理し、参照数字６２８で示されるように、ＤＢ２　６３４に格納するために結果をＩＢＭ　ＤＢ２コントローラ６１６に出力する。

【００６９】

具体的には、要求がすでにスタックの層間で移行し始めた後でフェイルオーバーが生じたとしても、該スタックは各々のトランザクションが厳密に一度だけ記録されることを保証するため、図６は、フェイルオーバーの際のＨＡシステムにおけるＪ２ＥＥ準拠ミドルウェア・スタックの利点を示す。さらに、アクティブ層のＭＱＳｅｒｉｅｓ（商標）サーバ６１０及びＤＢ２サーバ６１４は、１次ノードのみがアクセス可能であるが、フェイルオーバーの際には２次ノードによるアクセスのために迅速に再マウントされるｄｒｂｄパーティション６３０とインターフェース接続するため、図６は、フェイルオーバーの際のＨＡシステムにおけるＪ２ＥＥ準拠ミドルウェア・スタックの利点を示す。

【００７０】
ここで図７を参照すると、ＨＡクラスタのＪ２ＥＥ準拠ミドルウェア・スタックにｄｒｂｄパーティションを構成するための処理及びプログラムの概略的な論理フローチャートが示されている。図示されるように、処理は、ブロック７００で開始し、その後、ブロック７０２に進む。ブロック７０２は、ｄｒｂｄパーティションを構成し、マウントするステップを示す。次のブロック７０４は、ｄｒｂｄパーティション上のメッセージ・キュー及びデータベースをアクティブにするステップを示す。その後のブロック７０６は、フェイルオーバーの際にｄｒｂｄパーティションへのアクセスを効率的に移すために、ｄｒｂｄパーティションにアクセスするメッセージング・サーバ及びデータベース・サーバの仮想ＩＰアドレスを記録するステップを示し、処理は終了する。

【００７１】
ここで図８を参照すると、ＨＡクラスタのＪ２ＥＥ準拠ミドルウェア・スタックの構成及びフェイルオーバーを、ハートビート・コントローラを通じて制御するための処理及びプログラムの概略的な論理フローチャートが示されている。示されるように、処理は、ブロック８００で開始し、その後、ブロック８０２に進む。ブロック８０２は、１次ノードのミドルウェア層をアクティブにするステップを示す。その後のブロック８０４は、２次ノードのＨＴＴＰサーバ及びＷＡＳミドルウェア層をアクティブにするステップを示す。さらに、アクティブ／アクティブ構成で稼動するように指定される他のミドルウェア層がアクティブにされる。その後のブロック８０６は、２次ノードから１次ノードへのハートビート要求を定期的に起動するステップを示す。ブロック８０８は、ハートビートの戻りが２次ノードによって検出されたかどうかの判断を示す。ハートビートの戻りが検出された場合には、処理は８０６に戻る。ハートビートの戻りが検出されない場合には、処理はブロック８１０に移行する。

【００７２】
ブロック８１０は、ＳＴＯＮＩＴＨを呼び出して、１次ノードの電源を切るステップを示す。次のブロック８１２は、仮想ＩＰアドレスを１次ノードから引き継いで、２次ノードの冗長化コンポーネントに割り当てるステップを示す。その後のブロック８１４は、データディスク・スクリプトを呼び出して、２次ノードによるアクセスのためにｄｒｂｄパーティションを再マウントするステップを示す。次いで、ブロック８１６は、２次ノード上の待機ミドルウェア層と、ｄｒｂｄパーティション上の起動データとをアクティブにするステップを示す。フェイルオーバーの際に、ハートビート・ツール及び他のクラスタ管理サービスによって付加的なステップを実施しても良いことが理解されるであろう。

【００７３】
ここで図９を参照すると、Ｊ２ＥＥ準拠ミドルウェア・スタックによって提供されるサービスを監視するためにｍｏｎ機能を制御する処理及びプログラムの概略的な論理フローチャートが示されている。図示されるように、処理は、ブロック９００で開始し、その後、ブロック９０２に進む。ブロック９０２は、ミドルウェアによって提供されるサービスを監視するためのスケジュールを設定するステップを示す。次のブロック９０４は、予定された時間監視を起動させるかどうかの判断を示す。予定された時間監視が起動されない場合には、処理はブロック９０４で繰り返される。予定された時間監視が起動された場合には、処理はブロック９０６に移行する。ブロック９０６は、予定されたサービスのステータスを監視するステップを示す。その後のブロック９０８は、サービスが何らかの方法で休止又は故障として検出されたかどうかの判断を示す。サービスが休止として検出されない場合には、処理は終了する。サービスが休止として検出された場合には、処理はブロック９１０に移行する。ブロック９１０は、新たなＰＩＤを用いて同一のサービスを再開するステップを示し、処理は終了する。

【００７４】
ここで図１０を参照すると、本発明の方法、システム、及びプログラムに従って、Ｊ２ＥＥミドルウェア・スタックが稼動し、遠隔エンタープライズ・コンソールによって管理される多数のＨＡシステムを含むエンタープライズ・ネットワークのブロック図が示され

ている。図示されるように、ＨＡシステム１２０２及びＨＡシステム１２０４が、ネットワーク１０２を介してＨＡシステム１２０２及び１２０４を監視し、遠隔的に制御する遠隔エンタープライズ・コンソール１２１０に通信接続される。多数のＨＡシステムを、単一又は多数の遠隔集中コンソールによって監視し、制御できることが理解されるであろう。

【００７５】

本発明の利点として、ＨＡシステム１２０２及び１２０４の各々は、小売トランザクション及び他のミッションクリティカルな作業を処理することができる。一実施形態によると、ＨＡシステム１２０２及び１２０４の各々には、図４及び図５に示されたミドルウェア・スタックなどの、Ｊ２ＥＥアプリケーションを使用可能にする冗長化Ｊ２ＥＥ準拠ミドルウェア・スタックによって、高可用性を与えることができる。具体的には、ＨＡシステム１２０２及び１２０４の各々は、図３に示されるように、監視・構成コントローラ４１０を稼働させるクラスタ・マネージャを含む。

【００７６】

有利なことに、エラー、故障、又は非理想的な状態がＨＡシステム１２０２及び１２０４のいずれかで発生したときは、監視・構成コントローラ４１０は、そのエラー、故障、又は非理想的な状態の時点におけるシステムの状態を検出し、次に、情報を収集して、遠隔エンタープライズ・コンソール１２１０に対する報告を作成する。本発明の利点として、ハートビート・モニタ又はｍｏｎ機能が故障又はエラーを検出した場合には、監視・構成コントローラ４１０が起動して、その故障又はエラーを検出し、故障又はエラーの時点のシステム状態を判断する。

【００７７】

遠隔エンタープライズ・コンソール１２１０は、監視情報をデータベースに格納することが好ましい。次に、遠隔エンタープライズ・コンソール１２１０は、ＨＡシステム１２０２及び１２０４から受信したエラー又は故障の情報を分析し、場合によっては構成変更をＨＡシステムに戻して、フェイルオーバーの回避、及びフェイルオーバー効率の改善を試みる第１のコントローラを含むことが好ましい。さらに、遠隔エンタープライズ・コンソール１２１０は、多数のＨＡシステムから受信した故障情報、エラー情報、又は他の情報を比較して、どのシステムが修復及び更新を必要とし、どのシステムが性能要件を満たしていないかを判断する第２のコントローラを含むことができる。遠隔エンタープライズ・コンソール１２１０は、ＨＡシステム１２０２及び１２０４についての性能統計を収集して、その表示を制御することができる。

【００７８】

ここで図１１を参照すると、本発明の方法、システム、及びプログラムに係るＨＡクラスタ・マネージャ内部の監視コントローラを制御するための処理及びプログラムの概略的な論理フローチャートが示されている。図示されるように、処理は、ブロック１０００で開始し、その後、ブロック１００２に進む。ブロック１００２は、故障又はエラーが、ＨＡシステムのミドルウェア・スタックを監視するハートビート・モニタ、ｍｏｎ、又は他の監視コントローラから検出されたかどうかの判断を示す。故障又はエラーが検出されない場合には、ブロック１００２において処理が繰り返される。故障又はエラーが検出された場合には、処理は１００４に移行する。ブロック１００４は、故障又はエラーの時点で利用可能なシステム情報を収集し、分析するステップを示す。次のブロック１００６は、故障又はエラーの情報、及び利用可能なシステム情報を、ＨＡシステムを監視する遠隔集中コンソールに送信するステップを示し、処理は終了する。

【００７９】

ここで図１２を参照すると、ＨＡシステムのクラスタ・マネージャを遠隔的に制御して該ＨＡシステムを再構成するための処理及びプログラムの概略的な論理フローチャートが示されている。図示されるように、処理は、ブロック１１００で開始し、その後、ブロック１１０２に進む。ブロック１１０２は、ミドルウェア・スタックが稼働しているＨＡシステムを再構成するための構成要求を、遠隔エンタープライズ・コンソールから受信した

かどうかの判断を示す。要求を受信しない場合には、ブロック１１０２において処理が繰り返される。要求を受信した場合、処理はブロック１１０４に移行する。ブロック１１０４は、ハートビート・モニタを呼び出して、ＨＡシステムのフェイルオーバー設定を再構成するステップを示し、処理は終了する。さらに、ＨＡシステムのクラスタ・マネージャ内部の他のコントローラを呼び出して、ＨＡシステムの他のソフトウェア及びハードウェア構成を調整することができる。

【００８０】

ここで図１３を参照すると、クラスタにおける多数のＨＡシステムを管理するために遠隔エンタープライズ・コンソールを制御する処理及びプログラムの概略的な論理フローチャートが示されている。図示されるように、処理は、ブロック１３００で開始し、その後、ブロック１３０２に進む。ブロック１３０２は、監視情報がＨＡシステムから受信されたかどうかの判断を示す。監視情報が受信されない場合には、ブロック１３０２において処理が繰り返される。監視情報が受信された場合には、処理はブロック１３０４に移行する。具体的には、遠隔エンタープライズ・コンソールは、監視情報についての要求をＨＡシステムの各々に定期的に送信することができ、各々のＨＡシステムもまた、監視情報を自動的に送信することができる。

【００８１】

ブロック１３０４は、多数のＨＡシステムからの監視情報を格納するエンタープライズ・データベースに監視情報を追加するステップを示す。次のブロック１３０６は、監視情報が再構成のきっかけとなる場合に、ＨＡシステムの再構成を要求するステップを示す。具体的には、遠隔エンタープライズ・コンソールは、特定の形式のエラーが監視情報中に検出されたときに要求されることになる所定の構成を含むことができる。代替的には、システム管理者が、特定の形式のエラーに対する構成のタイプを推奨するようにしても良い。その後のブロック１３０８は、監視情報に基づいて、ＨＡシステムについての性能統計を再計算するステップを示す。具体的には、性能統計の計算は、ある種の監視エラー又は変動についてのみ起動されることになる。次のブロック１３１２は、このＨＡシステムの性能を、エンタープライズ・ネットワークの他のＨＡシステムの性能、及び、該エンタープライズ・ネットワークについて設定された性能要件と比較するステップを示す。次いで、ブロック１３１４は、性能の比較結果をチャート及びグラフで表示するステップを示す。例えば、チャートは、ＨＡシステムの位置の図形表現を描き、どのシステムが故障したかについての図形標識を提供し、他のＨＡシステムに対する各々のＨＡシステムの性能を示す図形標識を提供することができる。さらに、各々のシステムの実時間性能と、報告されたあらゆるエラーとを表示することができる。次のブロック１３１６は、ＨＡシステムの脆弱性に対する修正措置を推奨するステップを示し、処理は終了する。例えば、この推奨は、どのＨＡシステムが交換を必要とするか、どのＨＡシステムが更新を必要とするか、及び、どのＨＡシステムがソフトウェア更新又は微調整を必要とするかを示すものとすることができる。図１３に示される処理は、多数の高可用性サーバから受信される監視情報に関して実施することができる処理の形式の例であり、本発明の範囲から逸脱することなく、他の同様の分析又は出力を実施できることが理解されるであろう。

【００８２】

本発明は、好ましい実施形態に関して具体的に示され、説明されてきたが、当業者であれば、本発明の精神及び範囲から逸脱することなく、形態及び詳細の様々な変更を行うことができることを理解するであろう。

【図面の簡単な説明】

【００８３】

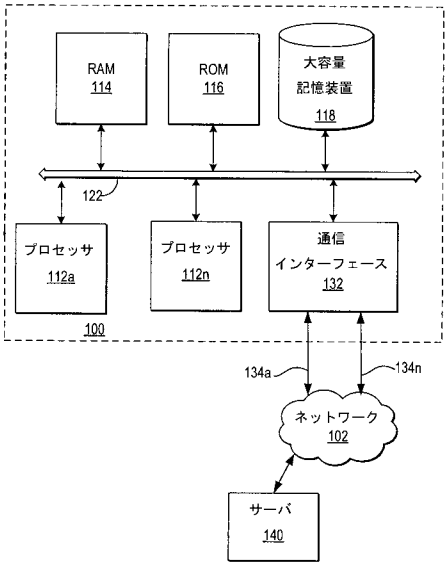
【図１】本発明の方法、システム、及びプログラムを実装することができるサーバ・システムを示すブロック図である。

【図２】フェイルオーバーの際にミドルウェアを効率的に移行させるための高可用性クラスタのハードウェア構成を示すブロック図である。

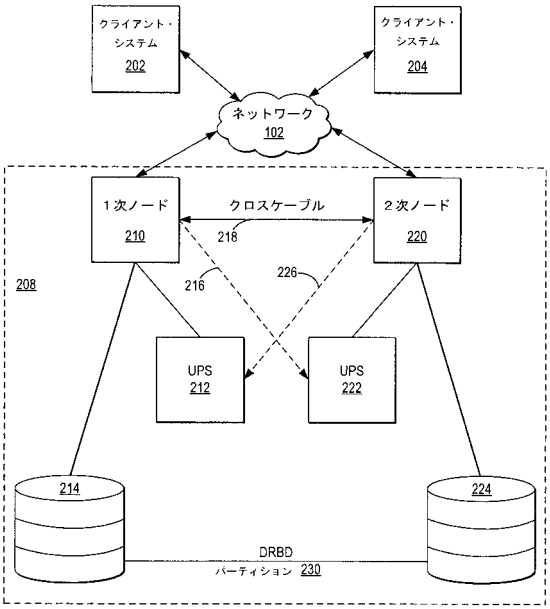
【図３】本発明の方法、システム、及びプログラムに係るクラスタ・マネージャを示すブ

ロック図である。
【図４】本発明の方法、システム、及びプログラムに係るフェイルオーバー前のＨＡクラスタのソフトウェア構成の一実施形態を示すブロック図である。
【図５】本発明の方法、システム、及びプログラムに係るフェイルオーバー後のＨＡクラスタのソフトウェア構成の一実施形態を示すブロック図である。
【図６】ＨＡシステムにおけるＪ２ＥＥ準拠ミドルウェア内部の独立系ソフトウェア・ベンダ・アプリケーションの実装の一実施形態を示すブロック図である。
【図７】ｄｒｂｄパーティションをＨＡクラスタのＪ２ＥＥ準拠ミドルウェア・スタックに構成するための処理及びプログラムを示す概略的なフローチャートである。
【図８】ＨＡクラスタのＪ２ＥＥ準拠ミドルウェア・スタックの構成及びフェイルオーバーを、ハートビート・コントローラを介して制御するための処理及びプログラムを示す概略的なフローチャートである。
【図９】Ｊ２ＥＥ準拠ミドルウェア・スタックによって提供されるサービスを監視するｍｏｎ機能を制御するための処理及びプログラムを示す概略的なフローチャートである。
【図１０】本発明の方法、システム、及びプログラムに従って、Ｊ２ＥＥミドルウェア・スタックを実行する多数のＨＡシステムを含むエンタープライズ・ネットワークを示すブロック図である。
【図１１】本発明の方法、システム、及びプログラムに従ってＨＡクラスタ内部の監視コントローラを監視するための処理及びプログラムを示す概略的なフローチャートである。
【図１２】ＨＡシステムのクラスタ・マネージャを遠隔的に制御して、該ＨＡシステムを再構成するための処理及びプログラムを示す概略的なフローチャートである。
【図１３】多数のＨＡシステムをクラスタ状態で管理する遠隔エンタープライズ・コンソールを制御するための処理及びプログラムを示す概略的なフローチャートである。
【符号の説明】
【００８４】
１０２：ネットワーク
２０２、２０４：クライアント・システム
２０８：高可用性システム
２１２、２２２：ＵＰＳ
２１４、２２４：データ記憶システム
２１６、２２６：シリアルケーブル
２１８：クロスケーブル
２１０：１次ノード
２２０：２次ノード
２３０：ＤＲＢＤパーティション

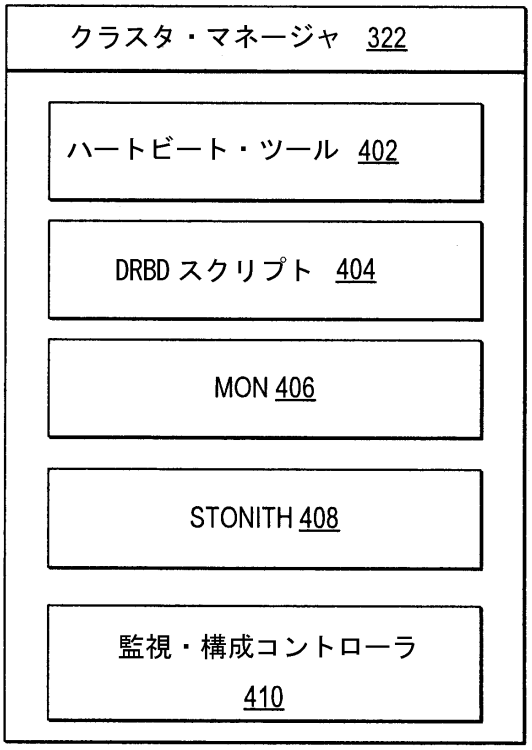
【図１】


RAM
114
ROM
116
大容量
記憶装置
118
122
プロセッサ
112a
プロセッサ
112n
通信
インターフェース
132
100
134a
134n
ネットワーク
102
サーバ
140

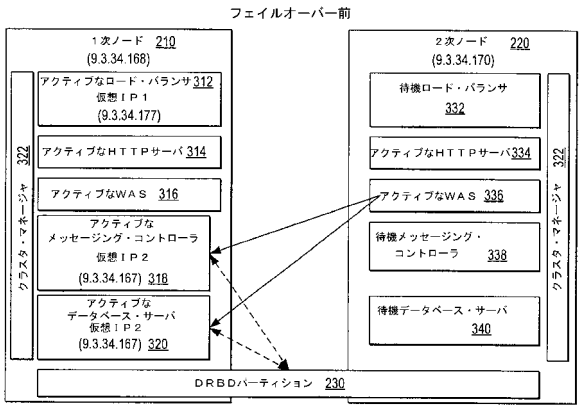
【図２】


クライアント・
システム
202
クライアント・
システム
204
ネットワーク
102
１次ノード
210
クロスケーブル
２次ノード
220
218
216
226
208
UPS
212
UPS
222
214
224
DRBD
パーティション 230

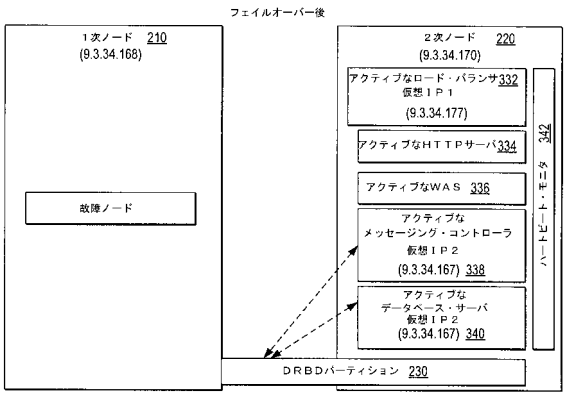
【図３】


クラスタ・マネージャ 322
ハートビート・ツール 402
DRBD スクリプト 404
MON 406
STONITH 408
監視・構成コントローラ
410

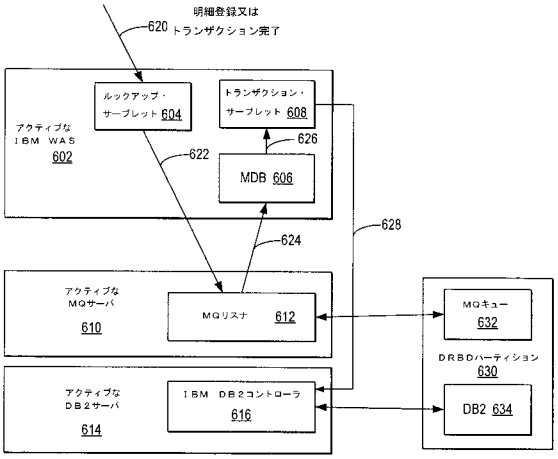
【図４】


フェイルオーバー前
１次ノード 210
(9.3.34.168)
２次ノード 220
(9.3.34.170)
アクティブなロード・バランサ 312
仮想ＩＰ１
(9.3.34.177)
待機ロード・バランサ
332
アクティブなＨＴＴＰサーバ 314
アクティブなＨＴＴＰサーバ 334
アクティブなＷＡＳ 316
アクティブなＷＡＳ 336
アクティブな
メッセージング・コントローラ
仮想ＩＰ２
(9.3.34.167) 318
待機メッセージング・
コントローラ 338
アクティブな
データベース・サーバ
仮想ＩＰ２
(9.3.34.167) 320
待機データベース・サーバ
340
クラスタ・マネージャ 322
クラスタ・マネージャ 322
DRBDパーティション 230

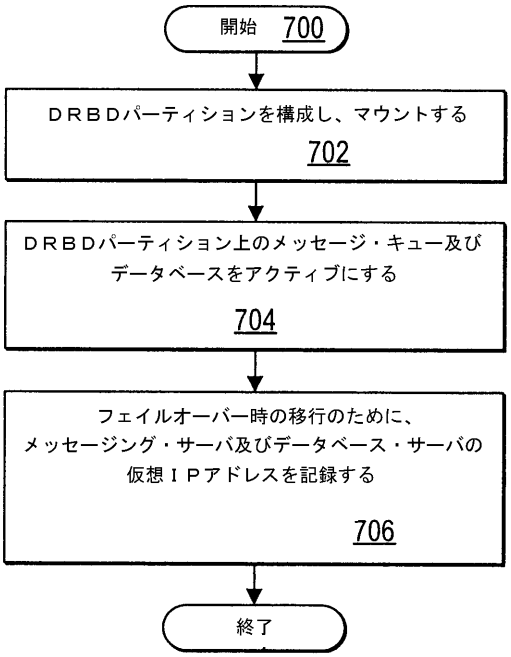
【図５】


フェイルオーバー後
１次ノード 210
(9.3.34.168)
２次ノード 220
(9.3.34.170)
アクティブなロード・バランサ 332
仮想ＩＰ１
(9.3.34.177)
故障ノード
アクティブなＨＴＴＰサーバ 334
アクティブなＷＡＳ 336
アクティブな
メッセージング・コントローラ
仮想ＩＰ２
(9.3.34.167) 338
アクティブな
データベース・サーバ
仮想ＩＰ２
(9.3.34.167) 340
ハートビート・モニタ 342
DRBDパーティション 230

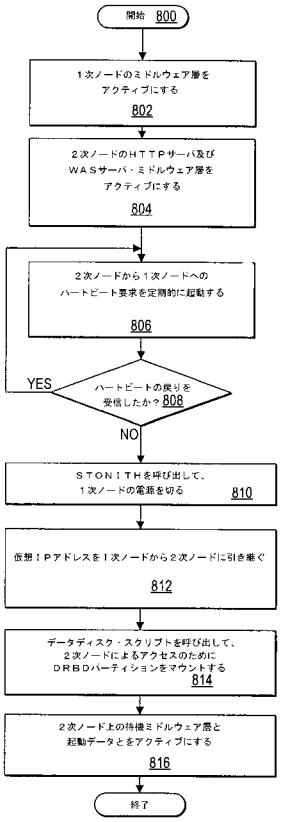
【図６】


明細登録又は
620 トランザクション完了
ルックアップ・
サーブレット 604
トランザクション・
サーブレット 608
アクティブな
IBM WAS
602
626
622
MDB 606
628
624
アクティブな
MQサーバ
610
MQリスナ 612
MQキュー
632
DRBDパーティション
630
アクティブな
DB2サーバ
614
IBM DB2コントローラ
616
DB2 634

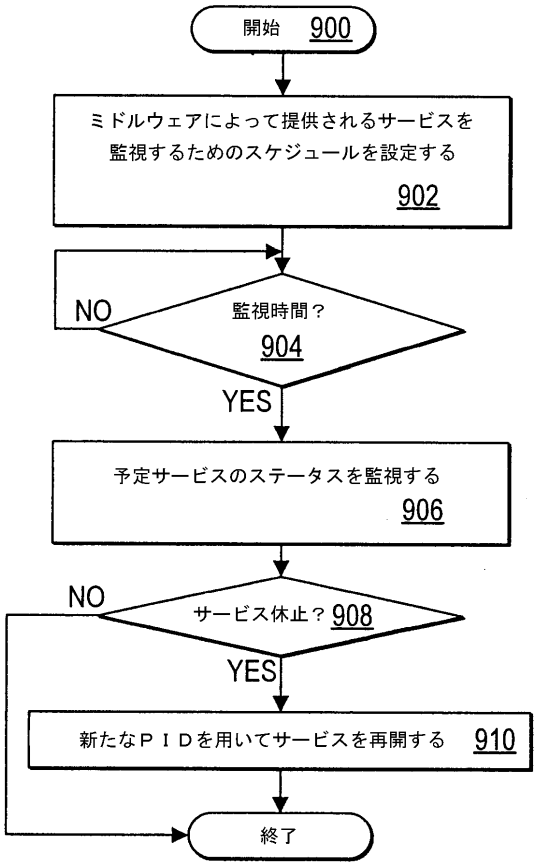
【図７】


開始 700
ＤＲＢＤパーティションを構成し、マウントする
702
ＤＲＢＤパーティション上のメッセージ・キュー及び
データベースをアクティブにする
704
フェイルオーバー時の移行のために、
メッセージング・サーバ及びデータベース・サーバの
仮想ＩＰアドレスを記録する
706
終了

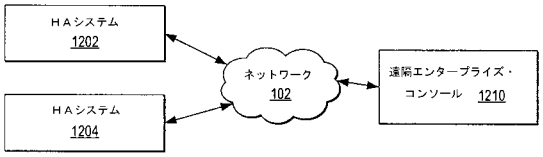
【図８】


開始 800
1次ノードのミドルウェア層を
アクティブにする
802
2次ノードのＨＴＴＰサーバ及び
ＷＡＳサーバ・ミドルウェア層を
アクティブにする
804
2次ノードから1次ノードへの
ハートビート要求を定期的に起動する
806
YES
ハートビートの戻りを
受信したか？ 808
NO
ＳＴＯＮＩＴＨを呼び出して、
1次ノードの電源を切る 810
仮想ＩＰアドレスを1次ノードから2次ノードに引き継ぐ
812
データディスク・スクリプトを呼び出して、
2次ノードによるアクセスのために
ＤＲＢＤパーティションをマウントする
814
2次ノード上の待機ミドルウェア層と
起動データとをアクティブにする
816
終了

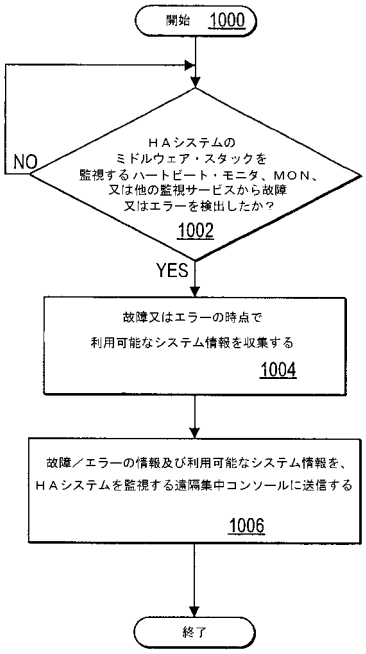
【図９】


開始 900
ミドルウェアによって提供されるサービスを
監視するためのスケジュールを設定する
902
NO
監視時間？
904
YES
予定サービスのステータスを監視する
906
NO
サービス休止？ 908
YES
新たなＰＩＤを用いてサービスを再開する 910
終了

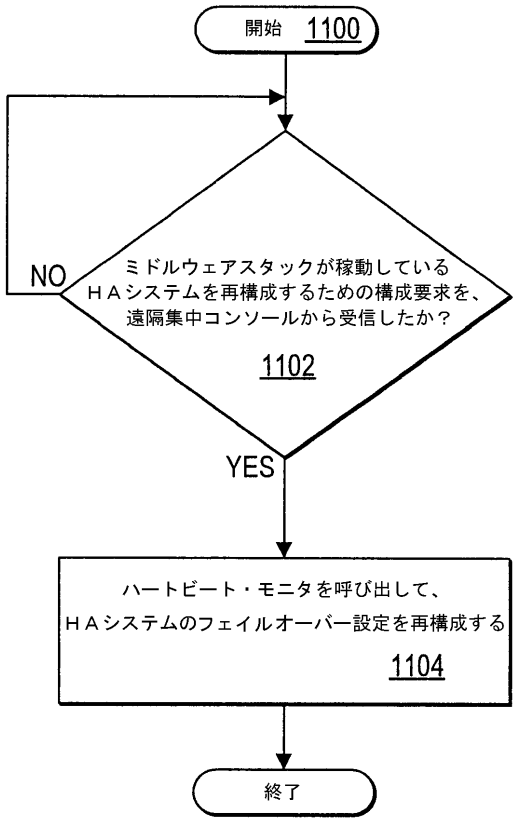
【図１０】



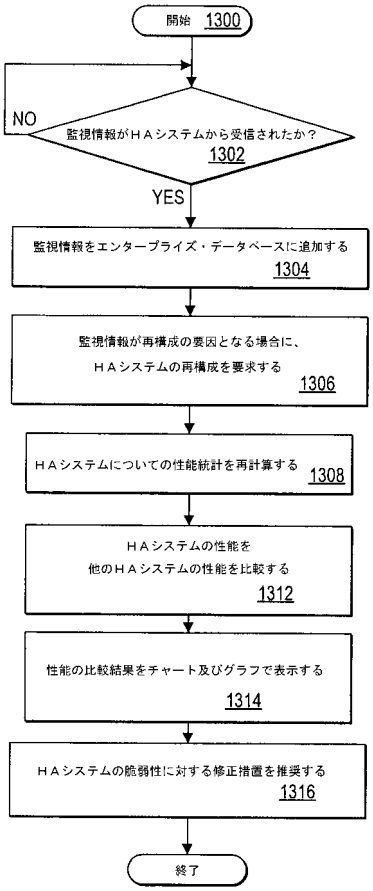
【図１１】



【図１２】



【図１３】



フロントページの続き

(74)代理人 100086243
弁理士 坂口 博
(72)発明者 フランシスコ・デ・ラ・クルーズ
アメリカ合衆国 ７８７１７ テキサス州 オースティン モースベリー・ドライブ １４５１３
(72)発明者 マイケル・エイ・パオリニ
アメリカ合衆国 ７８７５０ テキサス州 オースティン ウォレス・コーブ ６４０７
(72)発明者 ダグラス・スコット・ロザート
アメリカ合衆国 ７８７５８ テキサス州 オースティン ホビー・ホース・コート １１９０１ ナンバー１８１５
(72)発明者 ラダクリシュナン・セスラマン
アメリカ合衆国 ７８７５８ テキサス州 オースティン ストーンホロー・ドライブ １１９１５ ナンバー１４１８

合議体
審判長 吉岡 浩
審判官 宮司 卓佳
審判官 冨吉 伸弥

(56)参考文献 特開平１０－２５４８４４（ＪＰ，Ａ）
特開２００３－２９６１４１（ＪＰ，Ａ）
特開平９－９１２３３（ＪＰ，Ａ）
特開平９－３１１８４２（ＪＰ，Ａ）
特開平６－１７５８６８（ＪＰ，Ａ）
特開２０００－２０３３６（ＪＰ，Ａ）
特開２００２－３１２１８９（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F11/16,G06F13/00