(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6662847号
(P6662847)
(45)発行日　令和2年3月11日(2020.3.11)　(24)登録日　令和2年2月17日(2020.2.17)

(51)Int.Cl.		F I		
G06F 11/20	(2006.01)	G06F	11/20	630
G06F 11/14	(2006.01)	G06F	11/14	602Z
G06F 11/34	(2006.01)	G06F	11/34	123
G06F 15/173	(2006.01)	G06F	15/173	683E

請求項の数 23　(全 33 頁)

(21)出願番号　特願2017-505069 (P2017-505069)
(86)(22)出願日　平成27年7月20日 (2015.7.20)
(65)公表番号　特表2017-527893 (P2017-527893A)
(43)公表日　平成29年9月21日 (2017.9.21)
(86)国際出願番号　PCT/US2015/041121
(87)国際公開番号　WO2016/018663
(87)国際公開日　平成28年2月4日 (2016.2.4)
審査請求日　平成30年7月20日 (2018.7.20)
(31)優先権主張番号　14/445,369
(32)優先日　平成26年7月29日 (2014.7.29)
(33)優先権主張国・地域又は機関
米国 (US)

(73)特許権者　506018363
サウジ　アラビアン　オイル　カンパニー
サウジアラビア国　３１３１１　ダーラン、　イースタン　アベニュー　１
(74)代理人　100097320
弁理士　宮川　貞二
(74)代理人　100155192
弁理士　金子　美代子
(74)代理人　100131820
弁理士　金井　俊幸
(74)代理人　100100398
弁理士　柴田　茂夫
(72)発明者　アル－ワハビ，ハリド　エス．
サウジアラビア王国　ダーラン　３１３１１，ピー．オー．ボックス　８１６６
最終頁に続く

(54)【発明の名称】分散コンピューティング用のプロアクティブ障害回復モデル

(57)【特許請求の範囲】
【請求項１】
親／子型の関係で通信するようにマッピングされた、複数のコンピューティングノードの仮想ツリー状コンピューティング構造を構築するステップであって、前記コンピューティングノードのそれぞれに対して、１つ又は複数の直接の子供が初期設定の回復ノードとして指定され、別のノードがチェックポイントノードとして指定される、前記構築するステップと；
前記仮想ツリー状コンピューティング構造の各コンピューティングノードに対して、コンピューティングノードに関連する平均故障間隔（ＭＴＢＦ）を計算するために、ハードウェアプロセッサによって、ノード障害予測モデルを実行するステップと；
計算された前記ＭＴＢＦと、最大閾値及び最小閾値との比較に基づいて、第１のコンピューティングノードのチェックポイントを実行することを決定するステップと；
前記第１のコンピューティングノードから、前記第１のコンピューティングノードのために指定された初期設定の回復ノードへ、プロセスを移行するステップと；
前記第１のコンピューティングノードのために指定された初期設定の回復ノード上で前記プロセスの実行を再開するステップと；を備える、
コンピュータに実装される方法。
【請求項２】
各コンピューティングノードに対し、少なくともコンピューティング能力パラメータの値及びノード位置パラメータの値を収集するステップと；

前記ノード位置パラメータに基づいて、前記コンピューティングノードをコレクション（集合体）に分割するステップと；
前記コンピューティング能力パラメータに基づいて、各コレクション内の前記コンピューティングノードをソートするステップと；をさらに備える、
請求項１に記載のコンピュータに実装される方法。
【請求項３】
複数のコンピューティングノードの仮想ツリー状コンピューティング構造を構築するステップと；
前記仮想ツリー状コンピューティング構造の各コンピューティングノードに対して、コンピューティングノードに関連する平均故障間隔（ＭＴＢＦ）を計算するために、ハードウェアプロセッサによって、ノード障害予測モデルを実行するステップと；
計算された前記ＭＴＢＦと、最大閾値及び最小閾値との比較に基づいて、第１のコンピューティングノードのチェックポイントを実行するかどうかを決定するステップと；
前記第１のコンピューティングノードから、回復ノードとして機能する異なるコンピューティングノードへ、プロセスを移行するステップと；
前記異なるコンピューティングノード上で前記プロセスの実行を再開するステップと；
各コンピューティングノードに対し、少なくともコンピューティング能力パラメータの値及びノード位置パラメータの値を収集するステップと；
前記ノード位置パラメータに基づいて、前記コンピューティングノードをコレクションに分割するステップと；
前記コンピューティング能力パラメータに基づいて、各コレクション内の前記コンピューティングノードをソートするステップと；を備える、
コンピュータに実装される方法。
【請求項４】
ソートした前記コンピューティングノードのレベルを決定するために、上限及び下限を特定するステップと；
前記コンピューティング能力パラメータ並びに前記上限及び前記下限に基づいて、各コレクション内の前記コンピューティングノードを水平レベルにソートするステップと；
前記水平レベルの配置及び垂直の配置を、各コンピューティングノードに関連したそれぞれのノード記録情報テーブルに記録するステップと；
指定された初期設定の回復ノードを前記ノード記録情報テーブルに登録するステップと；をさらに備える、
請求項２または請求項３に記載のコンピュータに実装される方法。
【請求項５】
前記上限及び前記下限が、各コンピューティングノードに対して収集されたコンピューティング能力及びノード位置パラメータのクロスプロットから決定され、
前記垂直の配置が、各コンピューティングノードに対するノード位置パラメータに少なくとも基づいて決定される、
請求項４に記載のコンピュータに実装される方法。
【請求項６】
前記ＭＴＢＦが、ネットワーク又はデータストレージの障害に少なくとも基づいて計算される、
請求項１または請求項３に記載のコンピュータに実装される方法。
【請求項７】
前記コンピューティングノードのＭＴＢＦが前記最小閾値未満である場合にチェックポイントを作成するステップと；
前記ＭＴＢＦに等しくなるように、前記コンピューティングノードに関連する前記最小閾値を更新するステップと；をさらに備える、
請求項１または請求項３に記載のコンピュータに実装される方法。
【請求項８】

前記第１のコンピューティングノードの障害が発生したことを判断するステップと；
前記第１のコンピューティングノードに対して取得された最新のチェックポイントをプロセス状態として用いるステップと；をさらに備える、
請求項７に記載のコンピュータに実装される方法。
【請求項９】
コンピュータ読取可能命令を格納している非一時的なコンピュータ読取可能媒体であって、
コンピュータにより実行可能な前記命令は：
親／子型の関係で通信するようにマッピングされた、複数のコンピューティングノードの仮想ツリー状コンピューティング構造を構築することであって、前記コンピューティングノードのそれぞれに対して、１つ又は複数の直接の子供が初期設定の回復ノードとして指定され、別のノードがチェックポイントノードとして指定される、前記構築することと；
前記仮想ツリー状コンピューティング構造の各コンピューティングノードに対して、コンピューティングノードに関連する平均故障間隔（ＭＴＢＦ）を計算するために、ノード障害予測モデルを実行することと；
計算された前記ＭＴＢＦと、最大閾値及び最小閾値との比較に基づいて、第１のコンピューティングノードのチェックポイントを実行することを決定することと；
前記第１のコンピューティングノードから、前記第１のコンピューティングノードのために指定された初期設定の回復ノードへ、プロセスを移行することと；
前記第１のコンピューティングノードのために指定された初期設定の回復ノード上で前記プロセスの実行を再開することと；を含む動作をコンピュータに実行させるためのものである、
非一時的なコンピュータ読取可能媒体。
【請求項１０】
前記動作は：
各コンピューティングノードに対し、少なくともコンピューティング能力パラメータの値及びノード位置パラメータの値を収集することと；
前記ノード位置パラメータに基づいて、前記コンピューティングノードをコレクション（集合体）に分割することと；
前記コンピューティング能力パラメータに基づいて、各コレクション内の前記コンピューティングノードをソートすることと；をさらに含む、
請求項９に記載のコンピュータ読取可能媒体。
【請求項１１】
コンピュータ読取可能命令を格納している非一時的なコンピュータ読取可能媒体であって、
コンピュータにより実行可能な前記命令は：
複数のコンピューティングノードの仮想ツリー状コンピューティング構造を構築することと；
前記仮想ツリー状コンピューティング構造の各コンピューティングノードに対して、コンピューティングノードに関連する平均故障間隔（ＭＴＢＦ）を計算するために、ノード障害予測モデルを実行することと；
計算された前記ＭＴＢＦと、最大閾値及び最小閾値との比較に基づいて、第１のコンピューティングノードのチェックポイントを実行するかどうかを決定することと；
前記第１のコンピューティングノードから、回復ノードとして機能する異なるコンピューティングノードへ、プロセスを移行することと；
前記異なるコンピューティングノード上で前記プロセスの実行を再開することと；
各コンピューティングノードに対し、少なくともコンピューティング能力パラメータの値及びノード位置パラメータの値を収集することと；
前記ノード位置パラメータに基づいて、前記コンピューティングノードをコレクション

に分割することと；
前記コンピューティング能力パラメータに基づいて、各コレクション内の前記コンピューティングノードをソートすることと；を含む動作をコンピュータに実行させるためのものである、
非一時的なコンピュータ読取可能媒体。
【請求項１２】
前記動作は：
ソートした前記コンピューティングノードのレベルを決定するために、上限及び下限を特定することと；
前記コンピューティング能力パラメータ並びに前記上限及び前記下限に基づいて、各コレクション内の前記コンピューティングノードを水平レベルにソートすることと；
前記水平レベルの配置及び垂直の配置を、各コンピューティングノードに関連したそれぞれのノード記録情報テーブルに記録することと；
指定された初期設定の回復ノードを前記ノード記録情報テーブルに登録することと；をさらに含む、
請求項１０または請求項１１に記載のコンピュータ読取可能媒体。
【請求項１３】
前記上限及び前記下限が、各コンピューティングノードに対して収集されたコンピューティング能力及びノード位置パラメータのクロスプロットから決定され、
前記垂直の配置が、各コンピューティングノードに対するノード位置パラメータに少なくとも基づいて決定される、
請求項１２に記載のコンピュータ読取可能媒体。
【請求項１４】
前記ＭＴＢＦが、ネットワーク又はデータストレージの障害に少なくとも基づいて計算される、
請求項９または請求項１１に記載のコンピュータ読取可能媒体。
【請求項１５】
前記動作は：
前記コンピューティングノードのＭＴＢＦが前記最小閾値未満である場合にチェックポイントを作成することと；
前記ＭＴＢＦに等しくなるように、前記コンピューティングノードに関連する前記最小閾値を更新することと；をさらに含む、
請求項９または請求項１１に記載のコンピュータ読取可能媒体。
【請求項１６】
前記動作は：
前記第１のコンピューティングノードの障害が発生したことを判断することと；
前記第１のコンピューティングノードに対して取得された最新のチェックポイントをプロセス状態として用いることと；をさらに含む、
請求項１５に記載のコンピュータ読取可能媒体。
【請求項１７】
メモリストレージと相互運用可能な少なくとも１つのハードウェアプロセッサを備え：
親／子型の関係で通信するようにマッピングされた、複数のコンピューティングノードの仮想ツリー状コンピューティング構造を構築する、なお、前記コンピューティングノードのそれぞれに対して、１つ又は複数の直接の子供が初期設定の回復ノードとして指定され、別のノードがチェックポイントノードとして指定される；
前記仮想ツリー状コンピューティング構造の各コンピューティングノードに対して、コンピューティングノードに関連する平均故障間隔（ＭＴＢＦ）を計算するために、ノード障害予測モデルを実行する；
計算された前記ＭＴＢＦと、最大閾値及び最小閾値との比較に基づいて、第１のコンピューティングノードのチェックポイントを実行することを決定する；

前記第１のコンピューティングノードから、前記第１のコンピューティングノードのために指定された前記初期設定の回復ノードへ、プロセスを移行する；および、
前記第１のコンピューティングノードのために指定された初期設定の回復ノード上で前記プロセスの実行を再開する；ように構成された、
コンピュータシステム。
【請求項１８】
各コンピューティングノードに対し、少なくともコンピューティング能力パラメータの値及びノード位置パラメータの値を収集する；
前記ノード位置パラメータに基づいて、前記コンピューティングノードをコレクション（集合体）に分割する；および、
前記コンピューティング能力パラメータに基づいて、各コレクション内の前記コンピューティングノードをソートする；ようにさらに構成された、
請求項１７に記載のコンピュータシステム。
【請求項１９】
メモリストレージと相互運用可能な少なくとも１つのハードウェアプロセッサを備え：
複数のコンピューティングノードの仮想ツリー状コンピューティング構造を構築する；
前記仮想ツリー状コンピューティング構造の各コンピューティングノードに対して、コンピューティングノードに関連する平均故障間隔（ＭＴＢＦ）を計算するために、ノード障害予測モデルを実行する；
計算された前記ＭＴＢＦと、最大閾値及び最小閾値との比較に基づいて、第１のコンピューティングノードのチェックポイントを実行するかどうかを決定する；
前記第１のコンピューティングノードから、回復ノードとして機能する異なるコンピューティングノードへ、プロセスを移行する；
前記異なるコンピューティングノード上で前記プロセスの実行を再開する；
各コンピューティングノードに対し、少なくともコンピューティング能力パラメータの値及びノード位置パラメータの値を収集する；
前記ノード位置パラメータに基づいて、前記コンピューティングノードをコレクションに分割する；および、
前記コンピューティング能力パラメータに基づいて、各コレクション内の前記コンピューティングノードをソートする；ように構成された、
コンピュータシステム。
【請求項２０】
ソートした前記コンピューティングノードのレベルを決定するために、上限及び下限を特定する；
前記コンピューティング能力パラメータ並びに前記上限及び前記下限に基づいて、各コレクション内の前記コンピューティングノードを水平レベルにソートする；
前記水平レベルの配置及び垂直の配置を、各コンピューティングノードに関連したそれぞれのノード記録情報テーブルに記録する；および、
指定された初期設定の回復ノードを前記ノード記録情報テーブルに登録する；ようにさらに構成された、
請求項１８または請求項１９に記載のコンピュータシステム。
【請求項２１】
前記上限及び前記下限が、各コンピューティングノードに対して収集されたコンピューティング能力及びノード位置パラメータのクロスプロットから決定され、
前記垂直の配置が、各コンピューティングノードに対するノード位置パラメータに少なくとも基づいて決定される、
請求項２０に記載のコンピュータシステム。
【請求項２２】
前記ＭＴＢＦが、ネットワーク又はデータストレージの障害に少なくとも基づいて計算される、

請求項１７または請求項１９に記載のコンピュータシステム。
【請求項２３】
前記コンピューティングノードのＭＴＢＦが前記最小閾値未満である場合にチェックポイントを作成する；
前記ＭＴＢＦに等しくなるように、前記コンピューティングノードに関連する前記最小閾値を更新する；
前記第１のコンピューティングノードの障害が発生したことを判断する；および、
前記第１のコンピューティングノードに対して取得された最新のチェックポイントをプロセス状態として用いる；ようにさらに構成された、
請求項１７または請求項１９に記載のコンピュータシステム。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、分散コンピューティング用のプロアクティブ障害回復モデルに関する。
優先権の主張
本願は、２０１４年7月２９日に出願された米国特許出願第１４／４４５，３６９号に基づく優先権を主張し、当該米国特許出願のすべての記載内容を援用する。
【背景技術】
【０００２】
何千もの科学技術アプリケーションプロセスを用いる分散コンピューティングシステム（例えば、同種（クラスタ）、異種（グリッド及びクラウド）、等）上で地震データ処理、三次元リザーバ不確実性モデリング、及びシミュレーション等のクリティカル／リアルタイム科学技術アプリケーションを実行することは、所望の解答を生成するために、何日も又は何週間ものデータ処理を命じることが可能な高機能コンピューティング能力を必要とする。長時間のジョブ実行の成功は、システムの信頼性にかかっている。スーパーコンピュータ上に展開される大多数の科学技術アプリケーションは、そのプロセスのうちの１つだけが失敗した場合に失敗する可能性があるため、分散システムにおける故障許容は、複合コンピューティング環境において重要な特徴である。何らかの種類のコンピュータ処理障害をリアクティブ（事後対処的）に許容することは、通常、１つ以上のプロセスのステータス（状態）を定期的にチェックポインティングすることを可能にするかどうかの選択を伴い、これは、高性能コンピューティング環境において広く適用可能な効果的な技法である。しかし、この技法は、最適なチェックポイント間隔及びチェックポイントデータのための安定したストレージ（格納）位置を選択することに関するオーバーヘッド問題を有している。加えて、現在の障害回復モデルは、通常、数種類のコンピューティング障害に限られており、それらの有用性及び効率を制限するコンピューティング障害（単数又は複数）の場合に、手動で呼び出されている。
【発明の概要】
【０００３】
本開示は、実装に従って、分散コンピューティング用のプロアクティブ障害回復モデルを提供するための、コンピュータに実装される方法、コンピュータプログラム製品、及びコンピュータシステムを含む方法及びシステムについて説明する。１つのコンピュータに実装される方法は、複数のコンピューティングノードの仮想ツリー状コンピューティング構造を構築するステップと、仮想ツリー状コンピューティング構造の各コンピューティングノードに対して、コンピューティングノードに関連する平均故障間隔（ＭＴＢＦ）を計算するよう、ハードウェアプロセッサによって、ノード障害予測モデルを実行するステップと、計算したＭＴＢＦと、最大及び最小閾値との比較に基づいて、コンピューティングノードのチェックポイントを実行するかどうか決定するステップと、コンピューティングノードから回復ノードとして機能する異なるコンピューティングノードへプロセスを移行するステップと、異なるコンピューティングノード上でプロセスの実行を再開するステッ

プとを含む。
【０００４】
この局面の他の実装は、１つ以上のコンピュータ読取り可能媒体／ストレージデバイス上に記録され、それぞれが方法のアクションを実行するよう構成された、対応するコンピュータシステム、装置、及びコンピュータプログラムを含む。１つ以上のコンピュータのシステムは、動作において、システムにアクションを実行させる、システム上にインストールされるソフトウェア、ファームウェア、ハードウェア、若しくは、ソフトウェア、ファームウェア、又はハードウェアの組み合わせを有することによって、特定の動作又はアクションを実行するよう構成することができる。１つ以上のコンピュータプログラムは、データ処理装置によって実行される場合に、装置にアクションを実行させる命令を含むことによって、特定の動作又はアクションを実行するよう構成することができる。
【０００５】
前記及び他の実装は、以下の特徴のうちの１つ以上を、単独又は組み合わせで、それぞれ任意に含むことができる。
【０００６】
一般的な実装と組み合わせることができる第１の局面は、更に、各コンピューティングノードに対する少なくともコンピューティング能力及びノード位置パラメータ値を収集するステップと、それらのノード位置パラメータに基づいて、コンピューティングノードをコレクション（集合体）に分割するステップと、コンピューティング能力パラメータに基づいて、各コレクション内のノードをソートするステップとを含む。
【０００７】
先の局面のいずれかと組み合わせることができる第２の局面は、更に、ソートしたコンピューティングノードのレベルを決定するよう上限及び下限を特定するステップと、コンピューティング能力パラメータ並びに上限及び下限に基づいて、各コレクション内のコンピューティングノードを水平レベルにソートするステップと、水平レベル配置及び垂直配置を、各コンピューティングノードに関連するノード記録情報テーブルに記録するステップと、指定された回復ノードを各ノード記録情報テーブルに登録するステップとを含む。
【０００８】
先の局面のいずれかと組み合わせることができる第３の局面では、上限及び下限が、各コンピューティングノードに対して収集されたコンピューティング能力及びノード位置パラメータのクロスプロットから決定され、垂直配置が、各コンピューティングノードに対するノード位置パラメータに少なくとも基づいて決定される。
【０００９】
先の局面のいずれかと組み合わせることができる第４の局面では、ＭＴＢＦが、ネットワーク又はデータストレージ障害に少なくとも基づいて計算される。
【００１０】
先の局面のいずれかと組み合わせることができる第５の局面は、コンピューティングノードのＭＴＢＦが下限未満である場合にチェックポイントを作成するステップと、ＭＴＢＦに等しくなるように、コンピューティングノードに関連する下限を更新するステップとを更に含む。
【００１１】
先の局面のいずれかと組み合わせることができる第６の局面は、コンピューティングノードの障害が発生したことを判断するステップと、コンピューティングノードに対して取得された最新のチェックポイントをプロセス状態として用いるステップとを更に含む。
【００１２】
本明細書中に説明する主題は、以下の利点のうちの１つ以上を実現するように、特定の実装において実装することができる。第１に、説明する障害回復モデルシステム及び方法は、部分的／深刻な計算ノード（例えば、コンピュータサーバ、等）障害が発生した場合でさえも、計算プロセスの信頼性のある、継続的な動作を可能にする安価なフレームワーク設計を有し、ビジネス継続性最適化を向上している。障害回復モデルシステムは、継続

的な動作を可能にし、失敗したジョブ実行を最適に実行するための高性能定格を達成する。障害回復モデルはプロアクティブ（リアクティブではなく）であるため、コストはジョブを再処理することに対して更に低減され、障害回復実施からのコスト回避並びに時間及び労力両方の節約を可能にする。第２に、フレームワークは、膨大な数の計算ノードに対して適応性がある。第３に、フレームワーク設計は、異なる災害回復原理要因を考慮している。第４に、説明するシステム及び方法は、不必要なプロセスチェックポインティングによって生じるオーバーヘッドを極めて最小限にする。第５に、説明するシステム及び方法は、処理を最適化するためにどのような種類の負荷分散技法も実施するよう構成することができる。第６に、システム及び方法は、動作のために局所型又は集中型のチェックポイントストレージに依存しない。第７に、システム及び方法は、チェックポイントプロセスの最適配置を制御するために障害断定モデルに依存している。第８に、提案するシステム及び方法設計は、高度なビジネス継続性最適化を可能にする。他の利点は、当業者にとって明らかであろう。第９に、このフレームワーク設計内の障害断定モデルは、どのような種類の障害（電力供給、ソフトウェア、ハードウェア、ネットワーク、等）も捕捉及び対処できる。

【００１３】

本明細書の主題のひとつ以上の実装の詳細を、添付図面および以下の説明において述べる。主題の他の特徴、局面、及び利点は、明細書、図面、及び特許請求の範囲から明らかとなろう。

【図面の簡単な説明】

【００１４】

【図１Ａ】図１Ａは、実装に従って分散コンピューティング用のプロアクティブ障害回復モデルを提供するための方法を示す。

【図１Ｂ】図１Ｂは、実装に従って分散コンピューティング用のプロアクティブ障害回復モデルを提供するための方法を示す。

【図１Ｃ】図１Ｃは、実装に従って分散コンピューティング用のプロアクティブ障害回復モデルを提供するための方法を示す。

【００１５】

【図２】図２は、実装に従ってノード仮想ツリー状構造を構築するために用いることができる、ノードから収集されたパラメータのクロスプロット例を示す。

【００１６】

【図３】図３は、実装によるノードの仮想ツリー状構造例を示す。

【００１７】

【図４Ａ】図４Ａは、実装に従ってＭＴＢＦの計算に用いられるノード性能値を示す。

【００１８】

【図４Ｂ】図４Ｂは、実装に従ってノードに対するＭＴＢＦを計算するために用いられる典型的な学術的数式を示す。

【００１９】

【図５】図５は、実装に従ってＭＴＢＦに関連するチェックポイント間隔配置を示すグラフである。

【００２０】

【図６】図６は、部分的ノード障害が発生した場合の、どのように回復モデルが実装に従って回復に対して用いられるかの、ノードの仮想ツリー状構造例を示す。

【００２１】

【図７】図７は、実装に従ってノードが半故障を経験している場合の、仮想ツリー状構造のノードに対するチェックポインティングデータストレージノードを示す。

【００２２】

【図８】図８は、実装に従ってアプリケーション計算に参加しているノードを示す。

【００２３】

【図９】図９Ａ及び９Ｂは、実装に従って独立及び従属プロセスに関するチェックポイン

ティングノードリクエストを示す。
【００２４】
【図１０】図１０は、実装に従って分散コンピューティング用のプロアクティブ障害回復モデルを提供するために用いられるコンピューティングデバイス例を示すブロック図である。
【００２５】
種々の図面における同様の参照番号及び符号は、同様の構成要素を示している。
【発明を実施するための形態】
【００２６】
以下の詳細な説明は、当業者が開示された主題を製造し、用いることを可能にするよう表され、１つ以上の特定の実装の文脈において提供される。開示された実装に対する様々な変更は、当業者に容易に明らかとなり、本明細書中で定義される一般原理は、開示の適用範囲から逸脱することなく、他の実装及び用途に適用されてもよい。従って、本開示は、説明及び／又は図示する実装に限定する意図はないが、本明細書中に開示する原理及び特徴と一致する最も広い適用範囲と一致するものとする。
【００２７】
この開示は、概して、コンピューティングノード（例えば、コンピュータサーバ、等）障害の場合にビジネス継続性最適化を可能にするよう、分散コンピューティング用のプロアクティブ（事前対策的）障害回復モデル（ＦＲＭ）を提供するための、コンピュータに実装される方法、コンピュータプログラム製品、及びコンピュータシステムを含む方法及びシステムを説明する。以下の説明は、特定の実装に焦点を合わせているが、具体的な実装は、説明する主題の適用範囲を、他の使用に対し、及び、この開示と一致する方法で限定することを意味していない。
【００２８】
何千ものプロセスを用いる分散コンピューティングシステム（例えば、同種（クラスタ）、異種（グリッド及びクラウド）、等）上で地震データ処理及び三次元リザーバ不確実性シミュレーション及びモデリング等のクリティカル／リアルタイム科学技術アプリケーションを実行することは、高機能コンピュータ出力を必要とし、科学技術アプリケーションは、所望の解答を生成するために何日も、又は時には、何週間もデータを処理することに費やす可能性がある。長時間の実行の成功は、システムの信頼性にかかっている。スーパーコンピュータ上に展開される大多数の科学技術アプリケーションは、そのプロセスのうちの１つだけが失敗した場合に失敗する可能性があるため、分散システムにおける故障許容は、複合コンピューティング環境において重要な特徴である。何らかの種類のコンピュータ処理障害をリアクティブ（事後対処的）に許容することは、通常、１つ以上のプロセスのステータス（状態）を定期的にチェックポインティングすることを可能にするかどうかの選択を伴う。
【００２９】
チェックポインティングは、高性能コンピューティング環境において広く適用可能な効果的な技法であり、分散システムにおけるプロセス実行中の障害の場合に用いられる最も効果的な故障許容型の技法である。チェックポインティングにおいて、ノード上で実行するプロセスの状態は、ハードディスク、フラッシュメモリ、等のような信頼性があり、安定したストレージ上に定期的に保存される。いくつかの実装において、チェックポインティングは、オペレーティングシステムが後でプロセスを再構築するために用いることができる、実行しているプロセス（例えば、上で説明した「プロセスの状態」）を説明するファイルを作成する。例えば、チェックポイントファイルは、チェックポインティングされたプロセスのスタック、ヒープ、レジスタ（単数又は複数）に関するデータを含むことができる。チェックポイントファイルは、また、ペンディング信号のステータス、シグナルハンドラ、アカウント記録、端末状態、及び、所定時点でのプロセスを再構築するために必要なその他必要なデータを含むことができる。プロセスは、従って、プロセスを再度新たに再開することによって開始するのではなく、特定のチェックポイントが取得された点

において、及び、そこから実行を継続することを可能にしている。

【００３０】

高い水準において、ＦＲＭは、一貫した対象アプリケーション／プロセススループットを維持し、プロセス再処理実行時間を最適化／最小化するためにチェックポイントの最小限必要なセットを維持し、回復生存ノード（以下でより詳細に説明する）の中で最適な負荷分散ストラテジを達成し、ディスクレス又は入力／出力操作を最小限にし、チェックポイントデータを安定及び安全なストレージに格納し、及び／又は、メモリオーバーヘッドを最小限にするよう構成される。場合により、ＦＲＭは、また、処理ジョブが復旧に参加しているノードにおいてホストとして処理されていない限り、実行モード中である処理ジョブをブロックしない非ブロックチェックポインティングの使用を介して、チェックポインティング待ち時間（プロセスがチェックポイントリクエストを開始し、グローバル（広域）チェックポイントプロセスがそれを完了するまでの時間）を低減することができる。結果として、処理ジョブ実行待ち時間が低減される。

【００３１】

より詳細には、説明するＦＲＭは、場合により、１）無障害コンピューティングの場合は高性能、安定性コンピューティングシステムを支持する拡張可能な仮想ツリー状構造、及び、障害発生の場合は回復リソースの高稼働率、並びに、２）各チェックポイントリクエストの有効性及びそれに関する必要性を測定することによって、チェックポイントベースのアルゴリズムの協調及びコンテキスト切り替えのオーバーヘッドを最小限にするために典型例において用いられる障害予測モデル（ＦＰＭ）として実装される。

【００３２】

仮想ツリー状構造コンピューティングトポロジー設計

【００３３】

階層ツリー状コンピューティングトポロジー設計は、回復ノードと、異なる物理的な位置に存在してもよい遠隔の指定されたチェックポイントデータ格納ノードとの両方の割り当てに対する多数の選択を可能にする。通常の実装において、分散コンピューティング作業に参加している全てのコンピューティングノード（ノード）は、仮想ツリー状構造を構築するよう２つの異なるパラメータ：１）コンピューティング能力（ｃｏｍｐｕｔｉｎｇ　ｐｏｗｅｒ）（ＣＰ－Ｙ軸上に置かれる）及び２）ノード位置（ｎｏｄｅ　ｌｏｃａｔｉｏｎ）（ＮＬＯＣ－Ｘ軸上に置かれる）によって決定されるような仮想ツリー状構造に仮想的に置かれる。他の実装において、他のパラメータが、明白／有意の分類を可能にするため、及び、説明する仮想ツリー状構造を構築するため、Ｙ又はＸ軸のどちらか一方で収集及び／又は用いられてもよい。

【００３４】

ノード障害予測モデル

【００３５】

障害予測は、長い間、主に、実際の生産システムからの現実的な障害データの不足による研究問題に挑戦することとして見なされてきた。しかし、計算された平均故障間隔（ＭＴＢＦ）、ノードの信頼性を表すために用いられる統計パラメータは、ノードのための近い将来の所定期間内の故障率に対する良好な指標となることができる。

【００３６】

分散コンピューティング環境における障害は、いくつかの実装において、ロバストで総合的な障害回復モデルを確実にすると見なさなければならない５つの異なるカテゴリに分類することができる。例えば、カテゴリは以下を含むことができる：１）クラッシュ障害－サーバは停止するが、それが停止するまで正しく動作する；２）脱落障害（受信又は送信脱落のどちらか）－サーバが着信リクエストに応答することに失敗する、サーバが着信メッセージを受信することに失敗する、サーバがメッセージを送信することに失敗する；３）タイミング障害－サーバの応答が指定された時間間隔の外側に位置する；４）応答障害（値又は状態遷移障害）－サーバの応答が不正確；応答の値が間違っている、サーバが正しい制御フローから外れている；５）任意障害－サーバが任意応答を任意時間に生成し

ている恐れがある。
【００３７】
一般に、分散コンピューティングノード構造（例えば、ノードＮ０；Ｎ１；Ｎ２；．．．；Ｎｎ）は、ネットワーク接続によって、ローカル又はグローバル（例えば、インターネット又は他のネットワーククラウド）に接続される。各ノードは、通常、それ自体の物理的なメモリ及びローカルディスク（例えば、独立型（スタンドアローン）コンピューティングデバイス）を有し、安定した共有ストレージは、ノード間で共有される大量のデータセットのために配備される。科学技術、リアルタイム、等のアプリケーションにおいて、ノードのプロセス間の通信は、メッセージパッシングインターフェース（ＭＰＩ）、プロセス間のグローバル送信／受信リクエストのために指定される共有メモリ、及び／又は他の通信方法を介して達成することができる。通常、各プロセスは異なるノード上に常駐するが、２つ以上の別々のプロセスが単一ノード上で実行されてもよい。
【００３８】
プロセス間の通信チャンネルが安定し、信頼性があり、分散コンピューティングシステム内の各ノードが揮発性である（ノードが障害により分散コンピューティングシステムを離れるか、回復後に分散コンピューティングシステムに接合する可能性があることを意味する）と仮定し－また、故障したノードがコンピューティング環境から切り離されるフェイルストップモデルと仮定すると－ノードの障害は、故障したノード上のすべてのプロセスが動作を停止する原因となる（故障ノード上の影響を受けたプロセスの全てのデータが失われる）。ここで、ＦＲＭ回復／スペアノードが、それぞれ影響を受けたプロセスのために、（障害が物理的に修復されるまでアプリケーションを一時停止するのではなく）最後のチェックポイントから処理を継続するために用いることができる。
【００３９】
特定のノードに対するノード障害状況は、通常、決定される（例えば、ノードのソフトウェアエージェント－各ノードは通常それ自体のサービスデーモン「エージェント」を有する－によって予測される）／仮想ツリー状構造内の特定のノードと同じレベルに常駐するいずれかのノードによって（コンピューティング環境の構造を記述する、各ノードに保存された記録テーブル内の指定された参加ノードに）協力的に通知される。特定ノードの障害の断定は、近い将来のいつかに対する障害のリスクの評価を可能にし、特定ノードに関連するプロセス状態のより細かい進展（より高い粒度）を保存するようプロアクティブステップをとるための重要な指標である。従って、深刻な障害が、故障しそうだと既に予測された場合に、特定のノードに発生した場合、多大な再処理時間は、特定ノードに関連するプロセスを回復するために用いることができる利用可能なノード状態におけるより細かい粒度によって回避することができる。通常、ノード障害の断定は、チェックポイントを取得／格納するコストに対するバランスを保つ。
【００４０】
図１Ａ～１Ｃは、実装に従って分散コンピューティング用のプロアクティブ障害回復モデルを提供するための（下位方法１００ａ、１００ｂ、１００ｃに分割される）集合的な方法１００を示している。他の実装において、分散コンピューティング用のプロアクティブ障害回復モデルを提供することは、説明する各ステップ／操作よりも多くを含む、より多くの、又は、より少ないステップ／操作を含むことができる。方法１００（又はその個々の下位方法のいずれか）は、適切なシステム、環境、ソフトウェア、及び／又はハードウェア、又は、必要に応じて、システム、環境、ソフトウェア、及び／又はハードウェアの組み合わせ（例えば、以下で図１０において説明するコンピュータシステム）によって実行されてもよい。いくつかの実装において、方法１００の様々なステップは、並列に、組み合わせで、ループで、又は何らかの順序で実行することができる。
【００４１】
コンピューティングノードの仮想ツリー状構造化モデルを構築
【００４２】
図１Ａに目を向けると、１０２において、ノードの仮想ツリー状構造化モデルが、分散

コンピューティングシステム内で利用可能なノードを用いて構築されている。ツリー状構造は、ノードのツリーが実際にはツリー状構造で配置されていないが、親／子型の関係で通信するよう、この方法でマッピングされているため、「仮想的」と考えられる。当業者によって正しく認識されるように、コンピューティング能力（ＣＰ）及びノード位置（ＮＬＯＣ）パラメータの使用は、ノードの仮想ツリー状構造化モデルを構築する唯一可能な実装であり、この開示と一致する他のパラメータ（例えば、計算ハードウェア型及び／又はバージョン、ソフトウェアバージョン、等）は、他の実装において用いられてもよい。ＣＰ及びＮＬＯＣパラメータの使用は、説明する主題をどのような形でも限定することを意味せず、他のパラメータは、この開示の適用範囲（ｓｃｏｐｅ）内として想定される。通常の実装において、ＣＰはＹ軸上にあると考えられ、ＮＬＯＣ（又は他のパラメータ（単数又は複数））は、ノードの仮想ツリー状構造化モデルのＸ軸上にあると考えることができる。

【００４３】

１０３ａにおいて、少なくともコンピューティング能力（ＣＰ）、ノード位置（ＮＬＯＣ）、及び／又は他のパラメータは、例えば、計算プロセスの処理に参加する分散コンピューティングシステムの全てのノードのために収集される。いくつかの実装において、この収集されたデータは、仮想ツリー作成プロセス（不図示）による使用のために、データ構造、ファイル、等（例えば、ノード記録情報テーブル）内に置かれる。いくつかの実装において、各ノードは、全ての他のノード及び関連するパラメータを知っている。例えば、各ノードは、分散コンピューティングシステム内のノードのための収集されたパラメータ情報を含むデータ構造／ファイルに対するアクセスを有することができる。この情報は、各ノードが、兄弟、下位、等を知ることができるよう用いられてもよい。１０３ａから、方法１００ａは１０３ｂに進む。

【００４４】

１０３ｂにおいて、ノードは、それらの位置（ＮＬＯＣ）に基づいてコレクション毎に分割される。１０３ｂから、方法１００ａは１０３ｃに進む。

【００４５】

１０３ｃにおいて、ノードは、ノードＣＰパラメータに基づいて、各コレクション内でソートされる。１０３ｃから、方法１００ａは１０３ｄに進む。

【００４６】

１０３ｄにおいて、各レベルの下限及び上限（すなわち、閾値）は、ノードから収集されたパラメータのクロスプロットから決定される。ここで図２に目を向けると、図２は、実装に従ってノード仮想ツリー状構造を構築するために用いることができるノードから収集されたパラメータのクロスプロット例を示している。図示の通り、各ノード（ハッシング／パターンは異なる位置（ＮＬＯＣ）を示すための色を表すことができることに留意されたい－例えば、全ての「青色」にプロットされたノードは特定の位置におけるものである一方で、全ての「緑色」にプロットされたノードは異なる特定の位置におけるものである）は、いくつかの実装において、Ｘ軸２０２上のメモリパラメータ値（例えば、低から高－８ＧＢ～６４ＧＢ範囲のコンピュータサーバメモリを表す）及びＹ軸２０４上のＣＰパラメータ値（低から高－１．６～３．５ＧＨｚ範囲のプロセッサクロックを表す）に従ってプロットすることができる。当業者によって正しく認識されるように、これは、クロスプロットを生成する多数の可能な方法のうちの１つに過ぎない。この開示と一致する何らかの適切なパラメータの使用は、この開示の適用範囲内にあるよう想定される。

【００４７】

いくつかの実装において、ノードの水平配置（すなわち、ノードが一部である水平「線」）は、図２のクロスプロット内のノードの位置により、ＣＰに基づいている。例えば、ノードのＣＰパラメータ及び位置に基づいて、ノードは、仮想ツリー状構造内の下部、中間、又は上部位置にあってもよい。図示の実施例において、水平配置は、場合により、概して、下部に付けられたノードの大部分が、最も高いＣＰパラメータ値（より高い計算能力）を有するノードである一方で、ツリー状構造で、ノードが置かれる位置が高ければ高

いほど、ＣＰパラメータ値は低くなる（より低い計算能力）という可能性がある。
【００４８】
いくつかの実装において、ノードの垂直配置（例えば、上で説明した水平「線」に沿った左又は右－図３下部において、ノード３０４ｂが置かれているような）は、例えば、物理的位置、サブネット、帯域幅速度、電力供給ライン、等の異なる分類可能な基準によって決まり、初期設定により仮想ツリー状構造のバランスを保つ。例えば、クロスプロットにおいて用いられるＸ軸基準がツリー構造内の特定ノードの配置を案内する場合、仮想ツリー状構造はバランスが保たれ、正しい物理的災害－回復セットアップが計算環境に適用されるかどうかに関する指標として用いることができる。ノードを垂直に分けるために用いられる追加基準は、数ある中でも、物理的位置、サブネット、帯域幅速度、電力供給ライン、及び／又は他の追加基準を含むことができる。１０３ｄから、方法１００ａは１０３ｅに進む。
【００４９】
１０３ｅにおいて、各ノードのための水平及び／又は垂直ツリー配置入力が、ノード記録情報テーブルに行われる。１０３ｅから、方法１００ａは１０３ｆに進む。
【００５０】
１０３ｆにおいて、各ノードのためのノード記録情報テーブルは、ノードのための関連して指定されたチェックポイント及び／又は回復ノード（単数又は複数）が格納される。仮想ツリー状構造内の水平／垂直位置に基づいて、上位及び下位ノードを、仮想ツリー状構造内の各ノードのために決定することができ、ノード記録情報テーブルをこの情報で更新することができる。加えて、各特定ノードに対して、別のノード（例えば、１つ以上の直接的な子供）を、特定ノードのための初期設定の回復ノードとして指定することができ、別のノード（単数又は複数）を、特定ノードのチェックポイントノードとして指定することができる。通常、チェックポイントノードは、仮想ツリー状構造内の特定ノードの兄弟、子供、又は上位ノードではない。いくつかの実装において、特定ノードのための指定した回復ノード及びチェックポイントノードは、同じであってもよい。１０３ｆから、方法１００ａは１０４（ノード障害予測モデル）に進む。
【００５１】
ここで図３に目を向けると、図３は、実装による（例えば、上記の１０３ａ～１０３ｆによって構築されるような）ノードの仮想ツリー状構造例３００を示している。上記の１０３ｂで説明したように、３０２ａ、３０２ｂ、・・・、３０２ｎは、位置（ＮＬＯＣ）、例えば、異なるネットワークサブネットにおける場所による、ノードのコレクションを示している。上記の１０３ｃ及び１０３ｄで説明したように、３０４ａ、３０４ｂ、・・・、３０４ｎは、ＣＰパラメータによってソートされ、ノードから収集されたパラメータのクロスプロットによって水平レベルに分割されたノードである。例えば、ノード３０４ａは、ノード３０４ｎよりも高いＣＰパラメータ値を有していてもよい。更に、ノード（例えば、３０６ａ及び３０６ｂ）は、例えば、物理的位置、サブネット、帯域幅速度、等の上で説明した（又は他の）異なる分類可能な基準に基づいて、同じ水平レベル内で垂直に切り離される。一意のノード識別（ノードＩＤ）値例が、いくつかのノードの内部（例えば、３０４ｎはレベル１、ノード１のために「Ｎ１（１）」を表示し、ノード３０６ｂはレベル２、ノード１３のために「Ｎ２（１３）」を表示する）等に（Ｎ２の下のノードＩＤは示されていないが）示されている。いずれかの適切な一意のノード識別子も、この開示の適用範囲内にあるよう想定される。
【００５２】
仮想ツリー状構造のトポロジーは順応性であることに留意されたい。例えば、より多くの、又は、より少ないノードが特定位置に追加される場合、ノードＣＰ値は変化し、ノードはより高いＣＰ／メモリモデル等のためにスワップされ、そのＮＬＯＣ分割されたコレクション内の関係ツリーは更新することができ、他のＮＬＯＣ分割されたコレクションにおける他のノードとの関係も、更新することができる。例えば、新規ノードが分散コンピューティングシステムに追加される場合、仮想ツリー状構造基準プロセスは、再度実行す

ることができる。場合により、ツリーは部分的又は全体的に再構築することができる。
【００５３】
図４Ａに目を向けると、図４Ａは、実装に従ってＭＴＢＦの計算に用いられるノード性能値４００ａを示している。ここで、４０２は障害開始点（障害が始まった時間（又は「ダウンタイム」））であり、４０４は回復開始点４０４（処理が再開した時間（又は「アップタイム」））である。４０６は障害間の時間（「ダウンタイム」と「アップタイム」との間の差は２つの事象間で動作した時間）である。４０８は障害を表している。
【００５４】
図４Ｂに目を向けると、図４Ｂは、実装に従ってノードに対するＭＴＢＦを計算するために用いられる典型的な学術的数式４００ｂを示している。ここで、ＭＴＢＦは、（例えば、ノードに対する）多数の観測された障害４０８（例えば、再度ノードにとっての）によって分割される運転期間の合計である。当業者によって正しく認識されるように、より多くの、又は、より少ないデータ値を用いて、この開示と一致するＭＴＢＦ又は類似した値の他の変化は、ノード障害の予測及び以下で説明するような結果として生じる動作に用いられてもよい。１０４から、方法１００ａは方法１００ｂ（図１Ｂ）に進む。
【００５５】
図１Ｂにおいて、プロセス状態をチェックポインティングするか、及び／又は、プロセス（ジョブ）を移行させるかどうかの決定１００ｂが行われる。チェックポイント時間計算とは、ノードのチェックポイントを取得することのシステムオーバーヘッドを、（例えば、ノードに対するＭＴＢＦに基づいて）より必要であると考えられる時間に最小化することである。
【００５６】
１０６において、方法１００ａにおいて計算されたノードに対するＭＴＢＦは、最小閾値（ＭｉｎＬｉｍｉｔ）及び最大閾値（ＭａｘＬｉｍｉｔ）と比較される。最初に、ＭｉｎＬｉｍｉｔ及びＭａｘＬｉｍｉｔは、いくつかの所定の時間値に設定される。ＭｉｎＬｉｍｉｔ値は、（例えば、ノードに関する次の健全性チェックを実行する場合に決定するよう）必要に応じて変更することができる。ＭａｘＬｉｍｉｔも、（例えば、増加するＭＴＢＦ値を反映するよう）必要に応じて変更することができる。
【００５７】
１０６において、ＭＴＢＦがＭｉｎＬｉｍｉｔとＭａｘＬｉｍｉｔとの間（例えば、ＭｉｎＬｉｍｉｔ以上で、ＭａｘＬｉｍｉｔ以下）である場合、方法１００ｂは１０８に進む。１０８において、ノードのチェックポイントは全く取得されない。１０８から、方法１００ｂは１１０に進む。
【００５８】
１１０において、次のチェックポイントを取得するための時間が、ノードのためのソフトウェアエージェントによってノードに対して実行される新規ＭＴＢＦ計算に基づいて調整される（障害の観点－最新の時間においていくつの障害が起こったか－からノードの現在の状態を評価する）。この様に、チェックポイント間隔は、ノードの状態に応じて動的に調整することができる。例えば、最初のチェックポイントの後、仮に５分間が次のチェックポイントのために設定された場合、５分間待機する。次のチェックポイントの後、ＭＴＢＦ評価が、最新の５分間内の障害（もしあれば）に基づいて実行される。計算されたＭＴＢＦに基づいて、チェックポイント間隔を、上又は下に調整することができる（例えば、図５におけるように）。１１０から、方法１００ｂは、図１Ａに関して説明する１０４に進む。
【００５９】
仮に１０６において、ＭＴＢＦがＭａｘＬｉｍｉｔよりも大きい場合、方法１００ｂは１１２に進む。
１１２において、ノードのチェックポイントは全く取得されず、ＭａｘＬｉｍｉｔはＭＴＢＦと等しくなるよう更新される。いくつかの実装において、特定の閾値より上のＭａｘＬｉｍｉｔは、ＭａｘＬｉｍｉｔが高すぎる旨の警告の生成を開始できる。１１２から

、方法１００ｂは１１０に進む。
【００６０】
仮に１０６において、ＭＴＢＦがＭｉｎＬｉｍｉｔよりも小さい場合、方法１００ｂは１１４に進む。１１４において、ノードのチェックポイントが取得され、ＭｉｎＬｉｍｉｔは現在計算されているＭＴＢＦ値と等しくなるよう更新される。いくつかの実装において、チェックポイントは、数ある中でも、プロセス状態（レジスタ内容）、メモリ内容、通信状態（例えば、開いたファイル及びメッセージチャンネル）、関連するカーネルコンテキスト、及び／又は待ち行列のジョブを含むことができる。１１４から、方法１００ｂは１１６に進む。
【００６１】
１１６において、移行すべきジョブ（プロセス）がプロアクティブ障害回復のための閾値に基づいているか（例えば、１０６において判断されたようにＭＴＢＦ＜ＭｉｎＬｉｍｉｔ値）どうかに関して、決定が行われる。閾値は、特定ノードにおける障害頻度に応じて、主観的に強要されてもよい。ジョブを移行すべきではないと判断された場合、方法１００ｂは１１０に進む。ジョブを移行すべきと判断された場合、方法１００ｂは、ジョブ移行を実行するよう１１８に進む。
【００６２】
ここで図５に目を向けると、図５は、実装によるＭＴＢＦに関連するチェックポイント間隔配置を示すグラフである。図示の通り、チェックポイント間の時間（チェックポイント間隔）（例えば、５０２ａ、５０２ｂ、及び５０２ｃ）は、ＭＴＢＦが減少するにつれて（例えば、それぞれチェックポイント間隔例に対応する５０４ａ、５０４ｂ、及び５０４ｃ）短くなる。ＭＴＢＦが増加するにつれて（例えば、５０４ｄにおいて）、チェックポイント間隔が減少する（例えば、５０２ｄにおいて）。これは、障害のリスクが増加するにつれて（減少したＭＴＢＦのために）、より短い間隔でノードに対するチェックポイントを作成して、オーバーヘッドを最小限にし、ノードが故障した場合に（失う処理を最小限にするよう、実際の障害時間に近い点で回復できる）ビジネス継続性の最適性を最大化することが有利であるであることを確実にするためである。
【００６３】
ここで図１Ｃに目を向けると、図１Ｃは、ノード間でジョブを移行するための方法フロー１００ｃを示している。
【００６４】
１１９ａにおいて、回復ノードソフトウェアエージェントは、どのノードが移行するためのジョブのホストを務めるべきかを決定するようネゴシエートする。いくつかの実装において、ネゴシエーションは、負荷分散目的である。他の実装において、他のパラメータ／基準が、ネゴシエーションの目的で用いられてもよい。１１９ａから、方法１００ｃは１０９ｂに進む。
【００６５】
１１９ｂにおいて、ノードの障害が発生したかどうかに関する判定が行われる。ノードの障害は発生していないと判定された場合、方法１００ｃは１１９ｃに進む。ノードの障害が発生したと判定された場合、方法１００ｃは１１９ｄに進む。
【００６６】
１１９ｃにおいて、ノードのプロセス状態が抽出される。いくつかの実装において、プロセス状態は、数ある中でも、プロセス状態（レジスタ内容）、メモリ内容、通信状態（例えば、開いたファイル及びメッセージチャンネル）、関連するカーネルコンテキスト、及び／又は待ち行列のジョブを含むことができる。１１９ｃから、方法１００ｃは１１９ｅに進む。
【００６７】
１１９ｄにおいて、最後のチェックポイントが、現在のノード状態の代わりに用いられる（ノードが故障し、「ダウン」している／それからプロセス状態を取得することができないため）。いくつかの実装において、チェックポイントは、数ある中でも、プロセス状

態（レジスタ内容）、メモリ内容、通信状態（例えば、開いたファイル及びメッセージチャンネル）、関連するカーネルコンテキスト、及び／又は待ち行列のジョブを含むことができる。１１９ｄから、方法１００ｃは１１９ｅに進む。

【００６８】

１１９ｅにおいて、障害のある上位回復ノードソフトウェアエージェントは、信号／通信を新規の回復ノードに転送し続ける（例えば、通信プロトコルレベルでメッセージを検索することによって）。いくつかの実装において、故障したノードが修復された場合、信号／通信を転送する責務は、（修復されたノードが信号／通信を転送するプロセスに引き渡された時点で）修復されたノードによって行われてもよいことに留意されたい。１１９ｅから、方法１００ｃは１１９ｆに進む。

【００６９】

１１９ｆにおいて、プロセス譲渡が、「ダウン」したノードから回復ノードへ行われる。譲渡された状態は、通常、プロセスのアドレス空間、実行ポイント（レジスタ内容）、通信状態（例えば、開いたファイル及びメッセージチャンネル）、及び／又は他のオペレーティングシステムに依存する状態を含んでいる。１１９ｆから、方法１００ｃは１１９ｇに進む。

【００７０】

１１９ｇにおいて、プロセスは、回復ノード上で実行するよう再開される。１１９ｇから、方法１００ｃは停止する。図１Ｂに戻ると、１１８から特定のダウンノードのために方法１１８は停止する。通常の実装において、故障ノードは、ジョブ移行がそのために行われた場合に現在の計算実行から隔離され、故障ノードは、たとえ修復されたとしても、同じジョブファミリーに戻り、参加することはできない（例えば、ノードは、仮想ツリー状構造内の１つ以上のノードに対するノード記録情報テーブルから削除され、新しい計算実行の開始まで待機しなければならない）。処理は、ノード障害予測モデルを用いる異なるノードのために図１Ａに戻る。他の実装において、故障ノードを修復することが可能である（例えば、修復されたノードは、仮想ツリー状構造ノード記録情報テーブルに再度組み込まれ、ノード障害予測モデルによって処理され、処理、信号／通信を転送することを開始する、等が可能である）。

【００７１】

図１Ａに戻ると、１０４で、いくつかの実装において、ノード障害予測モデルが、数ある中でも、ノードのチェックポイントを実行するか、１つのノードから別のノードへジョブ（プロセス）を移行するかどうか等を決定するために、各ノードに対する現在の機械状態を評価するよう実行される。各ノードに対するＭＴＢＦの計算は、障害状況のために評価すべき特定のノードとツリー構造内で同じレベルに存在する少なくとも１つのノード上及び／又は特定のノード自体上に常駐する／実行するソフトウェアエージェントによって計算される。例えば、図３において、ノード３０４ｂが存在するレベルのいずれかのノードもノード３０４ｂに対するＭＴＢＦを判断でき、及び、この判断の適切なノードを通知できる。

【００７２】

１０４に図示の通り、いくつかの実装において、障害予測モデルに用いられる値は、例えば、「健全性チェック」型／「ハートビート」プログラムによって生成される１つ以上のシステムログ１０５ａ（例えば、ネットワーク接続障害／拒否、パケット損失、計算速度低下、低いメモリ状態、ストレージ／ネットワーク問題、等）、チェックする期間に対する調整１０５ｂ（例えば、計算に参加する各ノード上で実行される関数によって行われる次の健全性チェックを実行するために待機する期間を表す動的に計算される値。関数は、次の期間を決定するよう、チェックする期間１０５ｂ値を動的に計算した後、呼び出される）、及び、定期的に（例えば、ｔｐ期間）収集される障害クラスディクショナリ１０５ｃを含んでいる。いくつかの実装において、問題、障害、等の種類は、重み付けされてもよい（例えば、ネットワーク／ストレージがより重要である、等）。

【００７３】

１０５ｄにおいて、障害種類頻度は、ｔｐ期間毎に障害頻度を計算する場合に影響を測定するよう、障害の各クラスに主観的に割り当てられる重み値により障害毎に分類することによって計算される。例えば、電力供給及びネットワーク接続性は監視することができ、それらの障害種類頻度は、特定の期間ｔｐに計算することができる。他の実装において、障害種類頻度は、何らかの適切なアルゴリズムを用いて計算することができる。１０５ｅで、いくつかの実装において、ＭＴＢＦは、１を、計算した障害種類頻度で除算することによって計算することができる。

【００７４】

図６は、部分的ノード障害が発生した場合の、どのように回復モデルが実装に従って回復に対して用いられるかの、ノードの仮想ツリー状構造例６００を示している。仮想ツリー状構造において回復を実現することは、最初に、回復すべき故障ノードの下位の同じレベルのいずれかのノードからの通知によって達成され、そうでなければ、通知は故障ノードの上位に送信される。

【００７５】

実施例において、ノード６０２が故障した場合、ノード６０２の直接の子供（下位）６０４又は、その親ノードより高い計算能力を有する更なる下位６０６は、問題が生じたこと（例えば、ノード６０２との接続の損失、ノード６０２からのデータ受信が停止、ノード５０２に対するハートビート検出がノード障害を示している、等）を検出する。問題は、ビジネス継続性を最適化するよう、ノード６０２によって実行されるジョブを扱うために、どの他のノード（単数又は複数）がノード６０２と置き換えるべきか、である。選択は、親ノード６０４若しくは下位ノード６０６又は６０８のどちらか一方である。この場合、ノード６０６又は６０８には、ビジネス継続性を最適化するよう、早い期間で元はノード６０２上で実行されていたジョブを完了させるために、（それらの十分に高いコンピューティング能力ＣＰにより）ノード６０２に対するジョブを割り当てることができる。ノード６０２の下位を用いる決定は、また、下位ノードの負荷分散解析によって判断されるような子ノードの負荷追加に依存してもよい。

【００７６】

ノードの兄弟－下位（故障ノードの直接的な下位のうちの１つも活動していない場合）から、いずれかの、活動していて、利用可能な、軽く負荷がかかった（システム内の負荷分散によって決定される）ノードを割り当てることによるいずれかの故障ノードに対する回復は、世代停止と称される。例えば、ノード６１０が故障した場合、ノード６１０の子供が、ノード６１０の故障を検出するべきである。しかし、この実施例において、ノード６１０の全ての子供もダウンしている。次いで、ノード５１０のジョブを完了させるために、どのノード（単数又は複数）がノード５１０と置き換えるべきか、という問題ができる。ここで、上位ノード６１２は、（例えば、メッセージパッシングインターフェース（ＭＰＩ）及び／又は他のプロトコル等の、２つの異なるノード上に存在する２つのジョブ間の通信プロトコルによって）ノード６１０の障害を通知され、少なくとも故障したノード６１０のジョブ（及び、場合により、ノード６１０の下位のジョブ）の処理を引き継ぐよう、その同じレベルにある親族を探し求める。ここで、ノード６１２は、それとその子供が故障したノード６１０のジョブの処理を引き受けることができるかどうかを見るために、ノード６１４と通信する（ノード６１４は異なるサブネット内にあってもよい－図３を参照）。この実施例において、ノード６１４がこの障害回復タスクを引き受けたと仮定すると、それは、ビジネス継続性を最適化するよう、（例えば、コンピューティング能力、主題、負荷分散等に基づいて）必要に応じてジョブをそれらの子供等に委任することができるその直接的な子供にジョブを委任することができる。また、この実施例において、ノード６１４の１つの又は両方の直接的な子ノードも、故障ノード６１０又はノード６１４及びその右側に対するその兄弟６１８（又は６１２のレベルにある他のノード）に元々関連付けられたジョブに取り組むその助けも確保するよう、異なる親ノード（例えば、ノード６１６）の子ノードと通信することができる。通常、この場合、回復は、最も多くのボトムアップノード（より高いＣＰ値を有する）から開始して実行されるべきであり、こ

こで、他の兄弟の下位ノードの最も多くの下位ノードが、葉ノードのみの回復に参加し、それ故、葉ノードはそれらの上位ツリーの回復に参加する。

【００７７】

ここで図７に目を向けると、図７は、実装により、ノードが半故障を経験している場合の、仮想ツリー状構造のノードに対するチェックポインティングデータストレージノードを示している。例えば、ノード６１０は、チェックポイントデータを格納するための指定された１つ以上のチェックポインティングデータストレージノード７０２を有している。同様に、ノード７０４も、ノード６１０に対するチェックポインティングデータストレージノードとして指定された１つ以上のチェックポインティングデータストレージノード７０２を有している。いくつかの実装において、多数のノードは、同じチェックポインティングデータストレージノード（単数又は複数）を共有できる。他の実装において、チェックポインティングデータストレージノード７０２は、１つ以上のチェックポインティングデータストレージノードの故障が多くのチェックポインティングデータの損失の原因とならないように、チェックポインティングデータストレージノードの数を拡張するよう、１つのノード又は少数のノード（例えば、同じサブネット内、兄弟、等）によってのみ用いられる。ノードが故障した場合、そのノードに対する回復を課せられているノードは、初期設定で指定された回復ノード及び故障ノードに対するチェックポインティングデータストレージノードを決定するよう、ノード記録情報テーブルにアクセスできる。

【００７８】

通常の実装において、各プロセスは１つだけの永久チェックポイントを維持している。これは、全体のストレージオーバーヘッドを低減し、未使用／破棄チェックポイントを処分するためのガーベージコレクション活動の必要性を排除する。いくつかの実装において、各ノードのためのチェックポイントデータは、同時のそれらのノードの故障の可能性が低いため、ノードの同じレベル（例えば、兄弟内）にあるノードに保存される。通常の実装において、チェックポイントデータストレージノードは、ローカルジョブのための安全状態を獲得するリスクを最小限にするこの方法で実装される。チェックポイントデータストレージノードは、それらのジョブのためにメモリに常駐する作業セットデータを含む実行モードでの、又は、待ち行列内に入れられるかのどちらか一方の現在関係するジョブに対する情報を有している。

【００７９】

ここで図８に目を向けると、図８は、実装によりアプリケーション計算に参加しているノードを示している。先に説明したように、チェックポイントリクエストは、断定モデル又は所望の信頼性の度合いがチェックポイントを必要とするノードのソフトウェアエージェントによって開始される。ノードが独立したプロセス（例えば、スレッド、等）Ｘを有する場合、それは、（独立プロセスＸに参加する他のノードが全く存在しないように）他のノードに対するリクエストを伝えることなく、ツリー内のそれ自体の対応するチェックポイントストレージノードによるチェックポインティング活動を単に実行する。

【００８０】

しかし、プロセスが（例えば、他のプロセスに従属する）従属プロセスだった場合、チェックポイント開始ノードが、最後のチェックポイント／正常通信以降通信しており、リクエストをそれらの全てに伝えているプロセスの全てを識別する場合に、最小限のチェックポイントアプローチが適用される。リクエストを受信すると、各プロセスは、それ以上プロセスを識別できなくなるまで、順に、それが通信したプロセス全てを識別し、それらにリクエストを伝える等を行う。

【００８１】

図８を参照すると、従属プロセスノード８０２及び独立プロセスノード８０４の両方が識別される。例えば、従属プロセスノード例に対して、ノード８０６は、特定のプロセスに対する最上位ノードである。ノード８０６のプロセスに従属するプロセスを実行するノードは、対応する矢印によって示されている。従属プロセスノード（例えば、８０６）に対して、障害は、それらが共に作業しているため、全ての参加プロセスノード（例えば、

８１２、等）に（例えば、親ノード８１０によって）通信され、他のノードにとって、他の従属プロセスノード（例えば、８０６）が復旧し、次いで、従属プロセスが、それらが中断された箇所を続けることができるまで、それらのプロセスの状態を保存することが必要である。しかし、ノード８０８は、独立プロセスを実行しており、何の従属性も有していないため、回復は、独立ノードでそれ自体を単に心配する必要がある。

【００８２】

図９Ａ及び９Ｂを参照すると、図９Ａ及び９Ｂは、実装により、それぞれ独立及び従属プロセス９００ａ及び９００ｂに関するチェックポインティングノードリクエストを示している。図９Ａは独立プロセスを示している。例えば、独立プロセスＮ３　９０２がチェックポイントリクエストを受信した場合、Ｎ３のチェックポイントは、他のプロセスに関係なく実行される。図９Ｂのような従属プロセスＮ２　９０２ｂの場合において、一旦チェックポイントリクエストが受信されると、プロセスＮ２　９０２ｂは、チェックポイントリクエストをそれに直接従属するプロセス、例えば、従属プロセスＮ３　９０４ｂに渡す。従属プロセス９０４Ｂは、次いで、チェックポイントリクエストをそれに直接従属するプロセス（例えば、従属プロセスＮ４１　９０６ｂ及びＮ４６　９０８ｂ）、及びその他に渡す。チェックポイントは、従属プロセスからチェックポイントを通知／要求するための時間のため、上の「親」従属プロセス（例えば、従属プロセスＮ３　９０４ｂ）に対するチェックポイントよりも低いレベルの従属プロセス（例えば、従属プロセスＮ４１　９０６ｂ及びＮ４６　９０８ｂ）のために若干遅く発生する可能性があることに留意されたい。いくつかの実装において、各従属プロセスは、要求する従属プロセスに、そのチェックポインティング操作が完了する時を通知できる。

【００８３】

図１０に目を向けると、図１０は、実装により分散コンピューティング用のプロアクティブ障害回復モデルを提供するために用いられるコンピューティングデバイス例１０００を示すブロック図である。いくつかの実装において、ＥＤＣＳ１０００はコンピュータ１００２及びネットワーク１０３０を含んでいる。他の実装において、多数のコンピュータ及び／又はネットワークが、上で説明した方法（単数又は複数）を実行するよう共に作業できる。

【００８４】

図示のコンピュータ１００２は、コンピュータサーバ等のコンピューティングデバイスを包含するよう意図されているが、デスクトップコンピュータ、ラップトップ／ノートブックコンピュータ、無線データポート、スマートフォン、携帯情報端末（ＰＤＡ）、タブレットコンピューティングデバイス、これらのデバイス内部の１つ以上のプロセッサ、又は、コンピューティングデバイスの物理的及び／又は仮想インスタンスの両方を含むその他の適切な処理装置を含むこともできる。コンピュータ１００２は、キーパッド、キーボード、タッチスクリーン、又はユーザ情報を受け入れることができる他のデバイス（不図示）等の入力デバイスと、デジタルデータ、視覚及び／又は音声情報、又はユーザインターフェースを含む、コンピュータ１００２の動作に関連する情報を伝達する出力デバイス（不図示）とを含むコンピュータを備えていてもよい。

【００８５】

いくつかの実装において、コンピュータ１００２は、クライアント及び／又はサーバとして機能できる。通常の実装において、コンピュータ１００２は、並列処理ノード及び、また、この開示と一致する（不図示だとしても）ホスト、数ある中でも、ソフトウェアエージェント又は他のアプリケーション、プロセス、方法、等（例えば、アプリケーション１００７）のいずれか一方として作動する。図示のコンピュータ１００２は、ネットワーク１０３０と通信可能に連結される。いくつかの実装において、コンピュータ１００２の１つ以上のコンポーネントは、並列処理及び／又はクラウドコンピューティングベースの環境内で動作するよう構成されてもよい。コンピュータ１００２の実装は、また、ネットワーク１０３０上のメッセージパッシングインターフェース（ＭＰＩ）又は他のインターフェースを用いて通信できる。

【００８６】
高いレベルにおいて、コンピュータ１００２は、実装に従い分散コンピューティング用のプロアクティブ障害回復モデルを提供することに関連するデータ及び情報を受信、送信、処理、格納、又は管理するよう動作可能な電子コンピューティングデバイスである。いくつかの実装によれば、コンピュータ１００２は、また、シミュレーションサーバ、アプリケーションサーバ、電子メールサーバ、ｗｅｂサーバ、キャッシングサーバ、ストリーミングデータサーバ、解析サーバ、及び／又はその他のサーバを含むか、それらと通信可能に連結されてもよい。
【００８７】
コンピュータ１００２は、ネットワーク１０３０上で（例えば、別のコンピュータ１００２上で実行する）アプリケーション１００７からのリクエストを受信し、適切なソフトウェアアプリケーション１００７において前記リクエストを処理することによって受信したリクエストに応答することができる。加えて、リクエストは、また、内部ユーザ（例えば、コマンドコンソールから、又は、他の適切なアクセス方法によって）、外部又は第三者、他の自動化アプリケーション、並びにその他の適切な団体、個人、システム、又はコンピュータからコンピュータ１００２に送信されてもよい。
【００８８】
コンピュータ１００２のコンポーネントのそれぞれは、システムバス１００３を用いて通信できる。いくつかの実装において、コンピュータ１００２のいずれか及び／又は全てのコンポーネント、ハードウェア及び／又はソフトウェアの両方は、アプリケーションプログラミングインターフェース（ＡＰＩ）１０１２及び／又はサービス層１０１３を用いてシステムバス１００３上で、互いに、及び／又は、インターフェース１００４とインターフェース接続してもよい。ＡＰＩ１０１２は、ルーチン、データ構造、及びオブジェクトクラスに対する仕様を含んでいてもよい。ＡＰＩ１０１２は、コンピュータ言語非依存又は依存のどちらか一方であってもよく、完全なインターフェース、単一の関数、又はＡＰＩのセットでさえも参照してもよい。サービス層１０１３は、ソフトウェアサービスをコンピュータ１００２、及び／又は、コンピュータ１００２がその一部であるシステムに提供する。コンピュータ１００２の機能性は、このサービス層を用いて全てのサービス消費者のためにアクセス可能であってもよい。サービス層１０１３によって提供されるもののようなソフトウェアサービスは、規定されたインターフェースを介して再利用可能な、規定されたビジネス機能性を提供する。例えば、インターフェースは、拡張マークアップ言語（ＸＭＬ）形式、又は他の適切な形式において、データを提供するＪＡＶＡ（登録商標）、Ｃ＋＋、又は他の適切な言語で書かれたソフトウェアであってもよい。コンピュータ１００２の統合型コンポーネントとして説明される一方で、代替の実装が、コンピュータ１００２の他のコンポーネントに関連して、独立型コンポーネントとしてＡＰＩ１０１２及び／又はサービス層１０１３を説明してもよい。その上、ＡＰＩ１０１２及び／又はサービス層１０１３のいずれか又は全ての部分は、この開示の適用範囲から逸脱することなく、別のソフトウェアモジュールの子又はサブモジュール、企業用アプリケーション、又はハードウェアモジュールとして実装されてもよい。
【００８９】
コンピュータ１００２はインターフェース１００４を含んでいる。図１０において単一のインターフェース１００４として図示しているが、２つ以上のインターフェース１００４が、特定の必要性、要望、又はコンピュータ１００２の特定の実装に従って用いられてもよい。インターフェース１００４は、ネットワーク１０３０に（図示であろうと不図示であろうと）接続される並列処理環境を含む分散環境内の他のシステムと通信するためにコンピュータ１００２によって用いられる。一般に、インターフェース１００４は、適切な組み合わせでソフトウェア及び／又はハードウェア内で符号化され、ネットワーク１０３０と通信するよう可能なロジックを備えている。より詳細には、インターフェース１００４は、ネットワーク１０３０上での通信に関連する１つ以上の通信プロトコルに対応するソフトウェアを備えていてもよい。

【００９０】
コンピュータ１００２はプロセッサ１００５を含んでいる。図１０において単一のプロセッサ１００５として図示しているが、２つ以上のプロセッサが、特定の必要性、要望、又はコンピュータ１００２の特定の実装に従って用いられてもよい。一般に、プロセッサ１００５は、命令を実行し、コンピュータ１００２の動作を行うよう、データを操作する。特に、プロセッサ１００５は、分散コンピューティング用のプロアクティブ障害回復モデルを提供するために必要とされる機能を実行する。
【００９１】
コンピュータ１００２は、コンピュータ１００２及び／又はコンピュータがその一部であるシステムの他のコンポーネントのためのデータを保持するメモリ１００６も含んでいる。図１０において単一のメモリ１００６として図示しているが、２つ以上のメモリが、特定の必要性、要望、又はコンピュータ１００２の特定の実装に従って用いられてもよい。メモリ１００６は、コンピュータ１００２の統合型コンポーネントとして図示される一方で、代替の実装において、メモリ１００６はコンピュータ１００２の外部にあってもよい。いくつかの実装において、メモリ１００６は、方法１００に関して説明された何らかのデータ（例えば、チェックポイントデータ有効範囲スコア、同一性スコア、深さ比、等）及び／又はこの開示と一致するその他の適切なデータのうちの１つ以上を保持及び／又は参照することができる。
【００９２】
アプリケーション１００７は、特に、いくつかの実装に対して、分散コンピューティング用のプロアクティブ障害回復モデルを提供するために必要とされる機能に関して、特定の必要性、要望、若しくは、コンピュータ１００２及び／又はコンピュータ１００２がその一部であるシステムの特定の実装による機能を提供するアルゴリズムソフトウェアエンジンである。例えば、アプリケーション１００７は、ソフトウェアホスト、科学技術処理アプリケーション、チェックポインティングアプリケーション、回復アプリケーション、及び／又はこの開示と一致する（図示であろうと不図示であろうと）その他の種類のアプリケーション（又はその一部）として機能できる。単一のアプリケーション１００７として図示しているが、アプリケーション１００７は、コンピュータ１００２上で多数のアプリケーション１００７として実装されてもよい。加えて、コンピュータ１００２と一体化するものとして図示されているが、代替の実装において、アプリケーション１００７は、コンピュータ１００２の外部にあり、それとは別に実行できる。
【００９３】
この開示と一致する機能を実行する分散コンピュータシステムに関連するいくつの数のコンピュータ１００２が存在してもよい。更に、用語「クライアント」、「ユーザ」、及び他の適切な用語は、この開示の適用範囲から逸脱することなく、必要に応じて、交換可能に用いられてもよい。その上、この開示は、多くのユーザ／プロセスが１つのコンピュータ１００２を用いてもよいこと、又は、１つのユーザ／プロセスが多数のコンピュータ１００２を用いてもよいことを想定している。
【００９４】
本明細書中で説明した主題及び機能的動作の実装は、デジタル電子回路において、又は、本明細書中に開示した構造及びそれらの構造的均等物を含む無形で具現化されたコンピュータソフトウェア又はファームウェアにおいて、若しくはコンピュータハードウェアにおいて、又は、それらのうちの１つ以上の組み合わせにおいて実装することができる。本明細書中で説明した主題の実装は、データ処理装置による実行のため、又は、その動作を制御するよう、無形の非一時的なコンピュータストレージ媒体上で符号化された、１つ以上のコンピュータプログラム、すなわち、コンピュータプログラム命令の１つ以上のモジュールとして実装することができる。代替として、又は、加えて、プログラム命令は、人工的に生成される伝搬信号、例えば、データ処理装置による実行のために適切な受信機装置への送信のための情報を符号化するよう生成される、機械生成される電気的、光学的、又は電磁的信号上で符号化されてもよい。コンピュータストレージ媒体は、機械読取可能

ストレージデバイス、機械読取可能ストレージ基板、ランダム又はシリアルアクセスメモリデバイス、又はそれらのうちの１つ以上の組み合わせであってもよい。

【００９５】

用語「データ処理装置」は、データ処理ハードウェアを指し、例として、プログラム可能なプロセッサ、コンピュータ、若しくはマルチプロセッサ又はコンピュータを含むデータを処理するための全ての種類の装置、デバイス、及び機械を包含する。装置は、特殊用途論理回路、例えば、中央処理装置（ＣＰＵ）、コプロセッサ（例えば、グラフィック／視覚処理装置（ＧＰＵ／ＶＰＵ））、ＦＰＧＡ（フィールドプログラマブルゲートアレイ）、又はＡＳＩＣ（特定用途向け集積回路）であってもよく、又は、更にそれらを含むこともできる。いくつかの実装において、データ処理装置及び／又は特殊用途論理回路は、ハードウェアベース及び／又はソフトウェアベースであってもよい。装置は、コンピュータプログラムのための実行環境を生成するコード、例えば、プロセッサファームウェア、プロトコルスタック、データベース管理システム、オペレーティングシステム、又はそれらのうちの１つ以上の組み合わせを構成するコードを任意に含むことができる。本開示は、従来のオペレーティングシステム、例えば、ＬＩＮＵＸ、ＵＮＩＸ（登録商標）、ＷＩＮＤＯＷＳ（登録商標）、ＭＡＣ　ＯＳ、ＡＮＤＲＯＩＤ（登録商標）、ＩＯＳ又はその他の適切な従来のオペレーティングシステムを用いるか、又はそれらを用いないデータ処理装置の利用を想定している。

【００９６】

プログラム、ソフトウェア、ソフトウェアアプリケーション、モジュール、ソフトウェアモジュール、スクリプト、又はコードとも称されるか、それらとして説明されてもよいコンピュータプログラムは、コンパイル又はインタープリタ言語、若しくは宣言型又は手続き型言語を含むどのような形態のプログラミング言語で書かれてもよく、それは、独立型プログラムとして、若しくは、モジュール、コンポーネント、サブルーチン、又はコンピューティング環境での使用に適した他の単位として含むどのような形態で展開されてもよい。コンピュータプログラムは、ファイルシステム内のファイルに対応してもよいが、そうである必要はない。プログラムは、他のプログラム又はデータ、例えば、マークアップ言語文書に格納される１つ以上のスクリプトを保持するファイルの一部に、問題とするプログラム専用の単一ファイル内に、若しくは、多数の整合させたファイル、例えば、１つ以上のモジュール、サブプログラム、又はコードの一部を格納するファイル内に、格納することができる。コンピュータプログラムは、１つのコンピュータ、又は、一箇所に位置するか、多数箇所にわたって分散され、通信ネットワークによって相互接続される多数のコンピュータ上で実行されるよう展開することができる。様々な形で説明したプログラムの一部は、様々なオブジェクト、方法、又は他のプロセスを介して様々な特徴及び機能を実装する個々のモジュールとして示されているが、プログラムは、代わりに、必要に応じて、多くのサブモジュール、サードパーティサービス、コンポーネント、ライブラリ等を含んでいてもよい。反対に、様々なコンポーネントの特徴及び機能は、必要に応じて、単一のコンポーネントに結合することができる。

【００９７】

本明細書中に説明するプロセス及び論理フローは、入力データ上で操作し、出力を生成することによって機能を実行するよう、１つ以上のコンピュータプログラムを実行する１つ以上のプログラム可能なコンピュータによって実行することができる。プロセス及び論理フローは、また、特殊用途論理回路、例えば、ＣＰＵ、ＦＰＧＡ、又はＡＳＩＣによって実行することができ、装置は、また、それらとして実装することができる。

【００９８】

コンピュータプログラムの実行に適したコンピュータは、汎用又は特殊用途マイクロプロセッサ、両方、又はその他の種類のＣＰＵに基づくことができる。一般に、ＣＰＵは、読出し専用メモリ（ＲＯＭ）又はランダムアクセスメモリ（ＲＡＭ）若しくは両方から命令及びデータを受信する。コンピュータの必須要素は、命令を実施又は実行するためのＣＰＵと、命令及びデータを格納するための１つ以上のメモリデバイスとである。一般に、

コンピュータは、データを格納するための１つ以上の大容量ストレージデバイス、例えば、磁気、光磁気ディスク、又は光学ディスクも含むか、動作可能に結合され、それからデータを受信するか、それにデータを送信するか、又は両方を行う。しかし、コンピュータはかかるデバイスを有する必要はない。その上、コンピュータは、例えば、ほんの数例を挙げると、携帯電話、携帯情報端末（ＰＤＡ）、携帯用音声又は動画プレーヤ、ゲーム機、全地球測位システム（ＧＰＳ）受信機、又は可搬型ストレージデバイス、例えば、ユニバーサルシリアルバス（ＵＳＢ）フラッシュドライブの、別のデバイスに組み込まれてもよい。

【００９９】

コンピュータプログラム命令及びデータを格納することに適したコンピュータ読取り可能媒体（必要に応じて、一時的又は非一時的）は、例として、半導体メモリデバイス、例えば、消去可能なプログラム可能読出し専用メモリ（ＥＰＲＯＭ）、電気消去可能なプログラム可能読出し専用メモリ（ＥＥＰＲＯＭ）、及びフラッシュメモリデバイス；磁気ディスク、例えば、内部ハードディスク又はリムーバブルディスク；光磁気ディスク；並びに、ＣＤ－ＲＯＭ、ＤＶＤ＋／－Ｒ、ＤＶＤ－ＲＡＭ、及びＤＶＤ－ＲＯＭディスクを含む全ての形態の不揮発性メモリ、媒体、及びメモリデバイスを含んでいる。メモリは、キャッシュ、クラス、フレームワーク、アプリケーション、バックアップデータ、ジョブ、ｗｅｂページ、ｗｅｂページテンプレート、データベーステーブル、ビジネス及び／又は動的情報を格納するリポジトリ、及び、何らかのパラメータ、変数、アルゴリズム、命令、規則、制約、又はそれらに対する参照を含むその他の適切な情報を含んでいる様々なオブジェクト又はデータを格納してもよい。加えて、メモリは、ログ、ポリシー、セキュリティ又はアクセスデータ、報告ファイル、並びにその他等のその他の適切なデータを含んでいてもよい。プロセッサ及びメモリは、特殊用途論理回路によって補われてもよく、又は、それに組み込まれてもよい。

【０１００】

ユーザとの対話を提供するため、本明細書中に説明する主題の実装は、情報をユーザに表示するための表示デバイス、例えば、ＣＲＴ（陰極線管）、ＬＣＤ（液晶ディスプレイ）、ＬＥＤ（発光ダイオード）、又はプラズマモニタと、ユーザがコンピュータに入力を提供できるキーボードと、ポインティングデバイス、例えば、マウス、トラックボール、又はタッチパッドとを有するコンピュータ上で実装することができる。入力は、また、感圧性を有するタブレットコンピュータ表面等のタッチスクリーン、静電容量又は電気検知を用いるマルチタッチスクリーン、又は他の種類のタッチスクリーンを用いて、コンピュータに提供されてもよい。他の種類のデバイスが、同様にユーザとの対話を備えるよう用いられてもよく；例えば、ユーザに提供されるフィードバックは、いずれかの形態の感覚フィードバック、例えば、視覚フィードバック、音声フィードバック、又は触覚フィードバックであってもよく；そして、ユーザからの入力は、音響、音声、又は触覚入力を含むいずれかの形態で受信されてもよい。加えて、コンピュータは、ユーザによって用いられるデバイスにドキュメントを送信し、それからドキュメントを受信することによって、例えば、ｗｅｂブラウザから受信したリクエストに応答して、ユーザのクライアントデバイス上のｗｅｂブラウザにｗｅｂページを送信することによって、ユーザと対話できる。

【０１０１】

用語「グラフィカルユーザインターフェース」、すなわちＧＵＩは、１つ以上のグラフィカルユーザインターフェース及び特定のグラフィカルユーザインターフェースの表示のそれぞれを説明するために単数形又は複数形で用いられてもよい。従って、ＧＵＩは、情報を処理し、ユーザに情報結果を効果的に表すｗｅｂブラウザ、タッチスクリーン、又はコマンドラインインターフェース（ＣＬＩ）を含むが、これらに限定されない、何らかのグラフィカルユーザインターフェースを表してもよい。一般に、ＧＵＩは、ビジネススイートユーザによって操作可能な対話フィールド、プルダウンリスト、及びボタン等の、そのいくつか又は全てがｗｅｂブラウザに関連する複数のＵＩ要素を含んでいてもよい。これら及び他のＵＩ要素は、ｗｅｂブラウザの機能に関するか、それを表してもよい。

【０１０２】
本明細書中に説明する主題の実装は、例えば、データサーバのようなバックエンドコンポーネントを含むか、例えば、アプリケーションサーバのミドルウェアコンポーネントを含むか、ユーザが本明細書中に説明する主題の実装と対話できるグラフィカルユーザインターフェース又はＷｅｂブラウザを有する、例えば、クライアントコンピュータのフロントエンドコンポーネントを含むコンピューティングシステム、又は１つ以上のかかるバックエンド、ミドルウェア、又はフロントエンドコンポーネントの何らかの組み合わせにおいて実装することができる。システムのコンポーネントは、有線及び／又は無線デジタルデータ通信、例えば、通信ネットワークの何らかの形態又は媒体によって相互接続することができる。通信ネットワークの例は、ローカルエリアネットワーク（ＬＡＮ）、無線アクセスネットワーク（ＲＡＮ）、大都市エリアネットワーク（ＭＡＮ）、広域ネットワーク（ＷＡＮ）、ワールドワイド・インターオペラビリティ・フォー・マイクロウェーブ・アクセス（ＷＩＭＡＸ）、例えば、８０２．１１ａ／ｂ／ｇ／ｎ及び／又は８０２．２０を用いる無線ローカルエリアネットワーク（ＷＬＡＮ）、インターネットの全て又は一部、及び／又はその他の通信システム又は１つ以上の位置におけるシステムを含んでいる。ネットワークは、例えば、インターネットプロトコル（ＩＰ）パケット、フレームリレーフレーム、非同期転送モード（ＡＴＭ）セル、音声、動画、データ、及び／又はネットワークアドレス間の他の適切な情報により通信してもよい。
【０１０３】
コンピューティングシステムは、クライアント及びサーバを含むことができる。クライアント及びサーバは、一般に、互いに遠隔にあり、通常、通信ネットワークを介して相互に作用する。クライアント及びサーバの関係は、それぞれのコンピュータ上で動作し、互いにクライアント－サーバ関係を有するコンピュータプログラムのために生じる。
【０１０４】
いくつかの実装において、コンピュータシステムのいずれか又は全てのコンポーネント、ハードウェア及び／又はソフトウェアの両方は、アプリケーションプログラミングインターフェース（ＡＰＩ）及び／又はサービス層を用いて、互いに、及び／又は、インターフェースとインターフェース接続してもよい。ＡＰＩは、ルーチン、データ構造、及びオブジェクトクラスに対する仕様を含んでいてもよい。ＡＰＩは、コンピュータ言語非依存又は依存のどちらか一方であってもよく、完全なインターフェース、単一の関数、又はＡＰＩのセットでさえも参照してもよい。サービス層はソフトウェアサービスをコンピューティングシステムに提供する。コンピューティングシステムの様々なコンポーネントの機能性は、このサービス層を介して全てのサービス消費者のためにアクセス可能であってもよい。ソフトウェアサービスは、定義されたインターフェースを介して再利用可能な、定義されたビジネス機能性を提供する。例えば、インターフェースは、拡張マークアップ言語（ＸＭＬ）形式、又は他の適切な形式において、データを提供するＪＡＶＡ（登録商標）、Ｃ＋＋、又は他の適切な言語で書かれたソフトウェアであってもよい。ＡＰＩ及び／又はサービス層は、コンピューティングシステムの他のコンポーネントに関する統合型及び／又は独立型コンポーネントであってもよい。その上、サービス層のいずれか又は全ての部分は、この開示の適用範囲から逸脱することなく、別のソフトウェアモジュールの子又はサブモジュール、企業用アプリケーション、又はハードウェアモジュールとして実装されてもよい。
【０１０５】
この明細書は、多くの具体的な実装詳細を含んでいる一方で、これらは、いずれかの発明の適用範囲に関するか、又は、請求されるであろう適用範囲に関する限定として解釈すべきではないが、どちらかというと、特定の発明の特定の実装に対して特有であってもよい特徴の説明として解釈すべきである。別々の実装の文脈において本明細書中で説明するある特定の特徴も、単一の実装において組み合わせて実装することができる。反対に、単一の実装の文脈において説明される様々な特徴も、別々に多数の実装において、又は、何らかの適切な部分的組み合わせにおいて実装することができる。その上、特徴は、ある特

定の組み合わせにおいて機能するように上で説明され、更にはそういうものとして最初に主張されたかもしれないが、主張した組み合わせからの１つ以上の特徴は、いくつかの場合において、その組み合わせから削除されてもよく、主張した組み合わせは、部分的組み合わせ又は部分的組み合わせの変形例に向けられてもよい。

【０１０６】

同様に、動作は特定の順序で図面に示されているが、これは、かかる動作が図示した特定の順序又は一連の順序で実行されること、又は、図示したすべての動作が所望の結果を達成するよう実行されることが必要であると理解すべきではない。ある特定の状況において、マルチタスク及び並列処理が有利であってもよい。その上、上で説明した実装における様々なシステムモジュール及びコンポーネントの分離及び／又は統合は、すべての実装においてかかる分離及び／又は統合が必要であると理解すべきではなく、説明したプログラムコンポーネント及びシステムは、一般に、単一のソフトウェア製品に共に統合されるか、多数のソフトウェア製品にパッケージ化されてもよいことは、言うまでもない。

【０１０７】

主題の特定の実装を説明してきた。説明した実装の他の実装、代替、及び置換は、当業者に対して明らかなように、以下の特許請求の範囲の適用範囲内である。例えば、特許請求の範囲において列挙される動作は、異なる順序で実行され、依然として所望の結果を達成することができる。

【０１０８】

従って、実装例の上記説明は、この開示を定義し、限定するものではない。他の変更、代用、又は代替も、この開示の精神及び適用範囲から逸脱することなく可能である。

【０１０９】

特許請求の範囲

［第１の局面］

複数のコンピューティングノードの仮想ツリー状コンピューティング構造を構築するステップと；

前記仮想ツリー状コンピューティング構造の各コンピューティングノードに対して、コンピューティングノードに関連する平均故障間隔（ＭＴＢＦ）を計算するために、ハードウェアプロセッサによって、ノード障害予測モデルを実行するステップと；

計算された前記ＭＴＢＦと、最大閾値及び最小閾値との比較に基づいて、コンピューティングノードのチェックポイントを実行するかどうかを決定するステップと；

前記コンピューティングノードから、回復ノードとして機能する異なるコンピューティングノードへ、プロセスを移行するステップと；

前記異なるコンピューティングノード上で前記プロセスの実行を再開するステップと；を備える、

コンピュータに実装される方法。

［第２の局面］

前記各コンピューティングノードに対し、少なくともコンピューティング能力パラメータ値及びノード位置パラメータ値を収集するステップと；

前記ノード位置パラメータに基づいて、コンピューティングノードをコレクションに分割するステップと；

前記コンピューティング能力パラメータに基づいて、前記各コレクション内のノードをソートするステップと；をさらに備える、

第１の局面の方法。

［第３の局面］

ソートした前記コンピューティングノードのレベルを決定するために、上限及び下限を特定するステップと；

前記コンピューティング能力パラメータ並びに前記上限及び前記下限に基づいて、各コレクション内のコンピューティングノードを水平レベルにソートするステップと；

前記水平レベル配置及び垂直配置を、各コンピューティングノードに関連したノード記

録情報テーブルに記録するステップと；
指定された回復ノードを前記各ノード記録情報テーブルに登録するステップと；をさらに備える、
第２の局面の方法。
［第４の局面］
前記上限及び前記下限が、前記各コンピューティングノードに対して収集されたコンピューティング能力及びノード位置パラメータのクロスプロットから決定され、
前記垂直配置が、前記各コンピューティングノードに対するノード位置パラメータに少なくとも基づいて決定される、
第３の局面の方法。
［第５の局面］
前記ＭＴＢＦが、ネットワーク又はデータストレージ障害に少なくとも基づいて計算される、
第１の局面の方法。
［第６の局面］
前記コンピューティングノードのＭＴＢＦが前記下限未満である場合にチェックポイントを作成するステップと；
前記ＭＴＢＦに等しくなるように、前記コンピューティングノードに関連する下限を更新するステップと；をさらに備える、
第１の局面の方法。
［第７の局面］
前記コンピューティングノードの障害が発生したことを判断するステップと；
前記コンピューティングノードに対して取得された最新のチェックポイントをプロセス状態として用いるステップとをさらに備える；
第６の局面の方法。
［第８の局面］
コンピュータ読取可能命令を格納している非一時的なコンピュータストレージ媒体であって、
コンピュータにより実行可能な前記命令が、
複数のコンピューティングノードの仮想ツリー状コンピューティング構造を構築する；
前記仮想ツリー状コンピューティング構造の各コンピューティングノードに対して、コンピューティングノードに関連する平均故障間隔（ＭＴＢＦ）を計算するために、ノード障害予測モデルを実行する；
計算された前記ＭＴＢＦと、最大閾値及び最小閾値との比較に基づいて、コンピューティングノードのチェックポイントを実行するかどうかを決定する；
前記コンピューティングノードから、回復ノードとして機能する異なるコンピューティングノードへ、プロセスを移行する；および、
前記異なるコンピューティングノード上で前記プロセスの実行を再開する；ように構成されている、
非一時的なコンピュータストレージ媒体。
［第９の局面］
前記各コンピューティングノードに対し、少なくともコンピューティング能力パラメータ値及びノード位置パラメータ値を収集する；
前記ノード位置パラメータに基づいて、コンピューティングノードをコレクションに分割する；および、
前記コンピューティング能力パラメータに基づいて、前記各コレクション内のノードをソートする；命令をさらに含む、
第８の局面の媒体。
［第１０の局面］
ソートした前記コンピューティングノードのレベルを決定するために、上限及び下限を

特定する；
前記コンピューティング能力パラメータ並びに前記上限及び前記下限に基づいて、各コレクション内のコンピューティングノードを水平レベルにソートする；
前記水平レベル配置及び垂直配置を、各コンピューティングノードに関連したノード記録情報テーブルに記録する；および、
指定された回復ノードを前記各ノード記録情報テーブルに登録する；命令をさらに含む、
第９の局面の媒体。
［第１１の局面］
前記上限及び前記下限が、前記各コンピューティングノードに対して収集されたコンピューティング能力及びノード位置パラメータのクロスプロットから決定され、
前記垂直配置が、前記各コンピューティングノードに対するノード位置パラメータに少なくとも基づいて決定される、
第１０の局面の媒体。
［第１２の局面］
前記ＭＴＢＦが、ネットワーク又はデータストレージ障害に少なくとも基づいて計算される、
第８の局面の媒体。
［第１３の局面］
前記コンピューティングノードのＭＴＢＦが前記下限未満である場合にチェックポイントを作成する；および、
前記ＭＴＢＦに等しくなるように、前記コンピューティングノードに関連する下限を更新する；命令をさらに含む、
第８の局面の媒体。
［第１４の局面］
前記コンピューティングノードの障害が発生したことを判断する；および、
前記コンピューティングノードに対して取得された最新のチェックポイントをプロセス状態として用いる；命令をさらに含む、
第１３の局面の媒体。
［第１５の局面］
メモリストレージと相互運用可能な少なくとも１つのハードウェアプロセッサを備え：
複数のコンピューティングノードの仮想ツリー状コンピューティング構造を構築する；
前記仮想ツリー状コンピューティング構造の各コンピューティングノードに対して、コンピューティングノードに関連する平均故障間隔（ＭＴＢＦ）を計算するために、ノード障害予測モデルを実行する；
計算された前記ＭＴＢＦと、最大閾値及び最小閾値との比較に基づいて、コンピューティングノードのチェックポイントを実行するかどうかを決定する；
前記コンピューティングノードから、回復ノードとして機能する異なるコンピューティングノードへ、プロセスを移行する；および、
前記異なるコンピューティングノード上で前記プロセスの実行を再開する；ように構成された、
コンピュータシステム。
［第１６の局面］
前記各コンピューティングノードに対し、少なくともコンピューティング能力パラメータ値及びノード位置パラメータ値を収集する；
前記ノード位置パラメータに基づいて、コンピューティングノードをコレクションに分割する；および、
前記コンピューティング能力パラメータに基づいて、前記各コレクション内のノードをソートする；ようにさらに構成された、
第１５の局面のシステム。

［第１７の局面］
ソートした前記コンピューティングノードのレベルを決定するために、上限及び下限を特定する；
前記コンピューティング能力パラメータ並びに前記上限及び前記下限に基づいて、各コレクション内のコンピューティングノードを水平レベルにソートする；
前記水平レベル配置及び垂直配置を、各コンピューティングノードに関連したノード記録情報テーブルに記録する；および、
指定された回復ノードを前記各ノード記録情報テーブルに登録する；ようにさらに構成された、
第１６の局面のシステム。
［第１８の局面］
前記上限及び前記下限が、前記各コンピューティングノードに対して収集されたコンピューティング能力及びノード位置パラメータのクロスプロットから決定され、
前記垂直配置が、各コンピューティングノードに対するノード位置パラメータに少なくとも基づいて決定される、
第１７の局面のシステム。
［第１９の局面］
前記ＭＴＢＦが、ネットワーク又はデータストレージ障害に少なくとも基づいて計算される、
第１５の局面のシステム。
［第２０の局面］
前記コンピューティングノードのＭＴＢＦが前記下限未満である場合にチェックポイントを作成する；
前記ＭＴＢＦに等しくなるように、前記コンピューティングノードに関連する下限を更新する；
前記コンピューティングノードの障害が発生したことを判断する；および、
前記コンピューティングノードに対して取得された最新のチェックポイントをプロセス状態として用いる；ようにさらに構成された、
第１５の局面のシステム。

【図１Ａ】

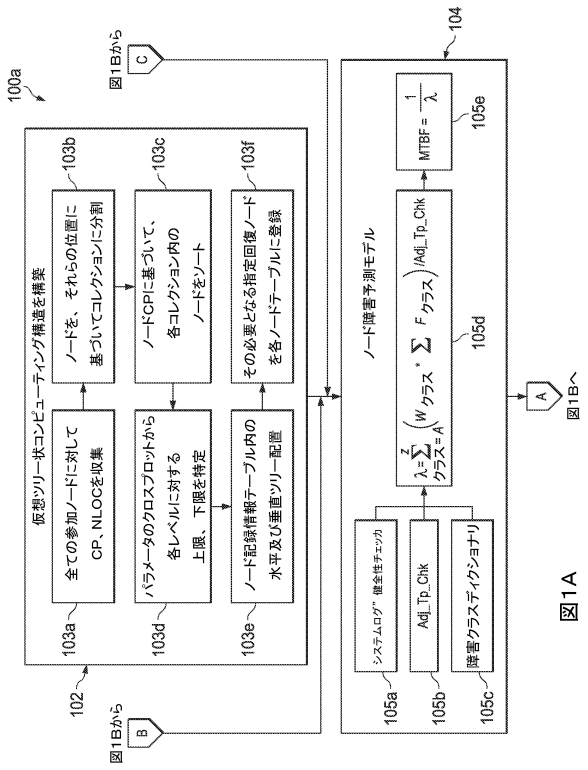


図1A

【図１Ｂ】

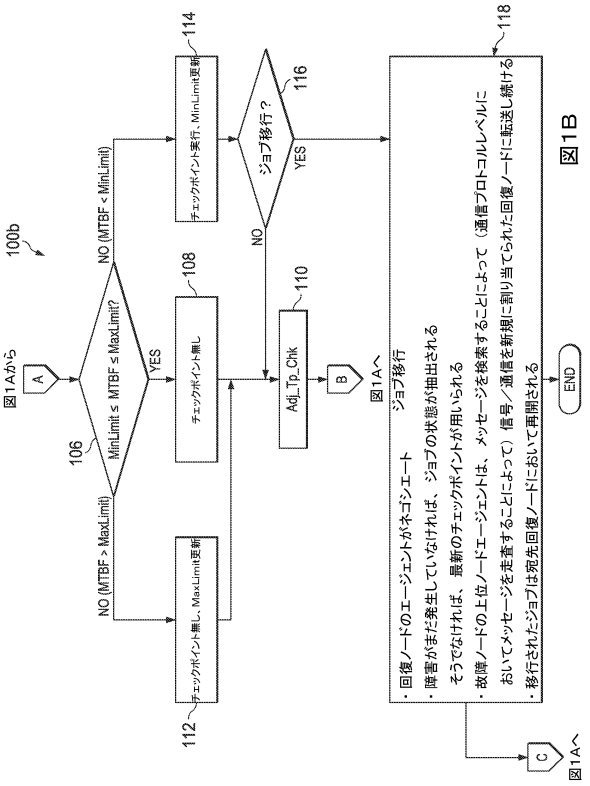


図1B

【図１Ｃ】

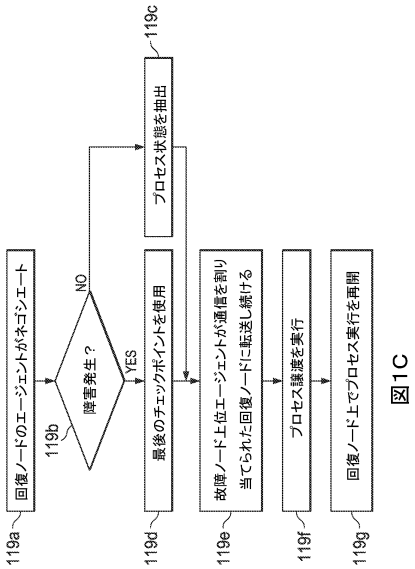


図1C

【図２】

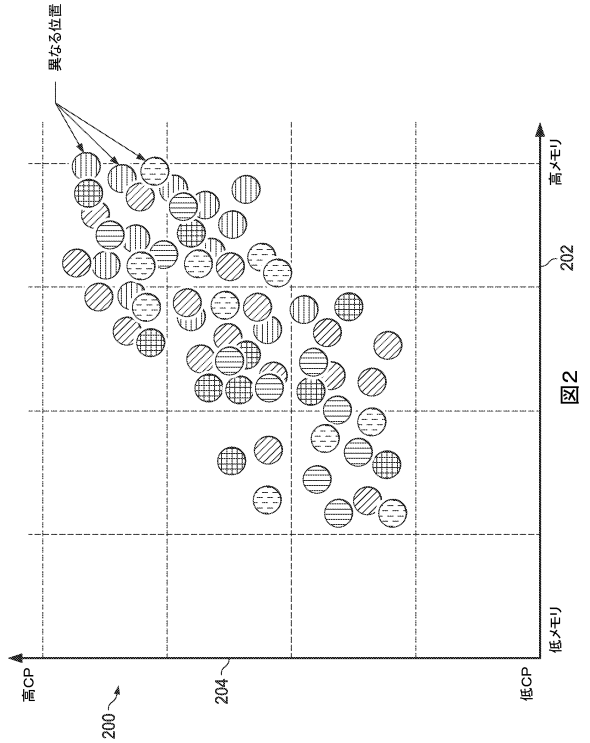


図2

【図３】

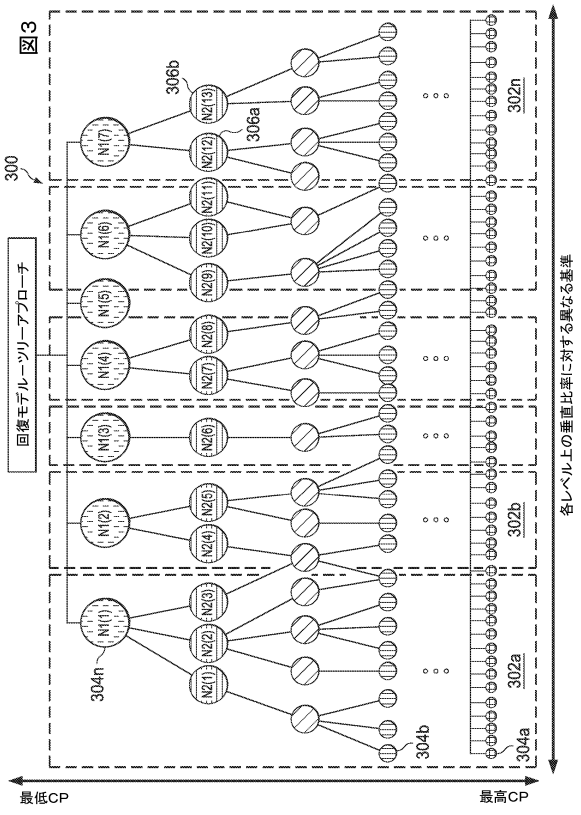
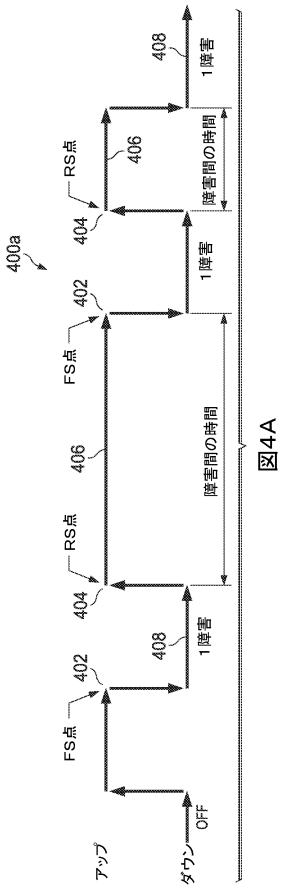
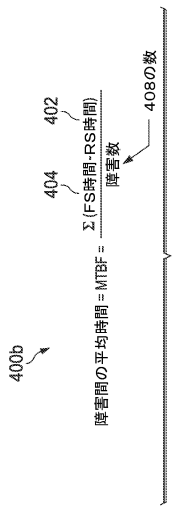


図3
300
回復モデルーツリーアプローチ
306b
306a
304n
304b
304a
302a
302b
302n
各レベル上の垂直比率に対する異なる基準
最低CP
最高CP

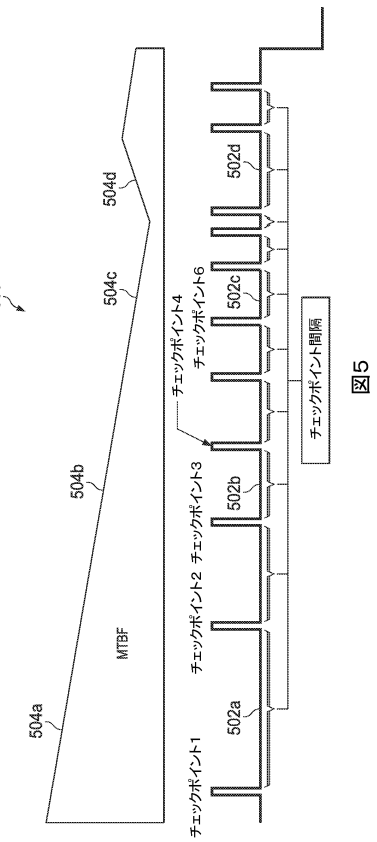
【図４Ａ】


400a
FS点
RS点
1障害
障害間の時間
アップ
ダウン
OFF
図4A

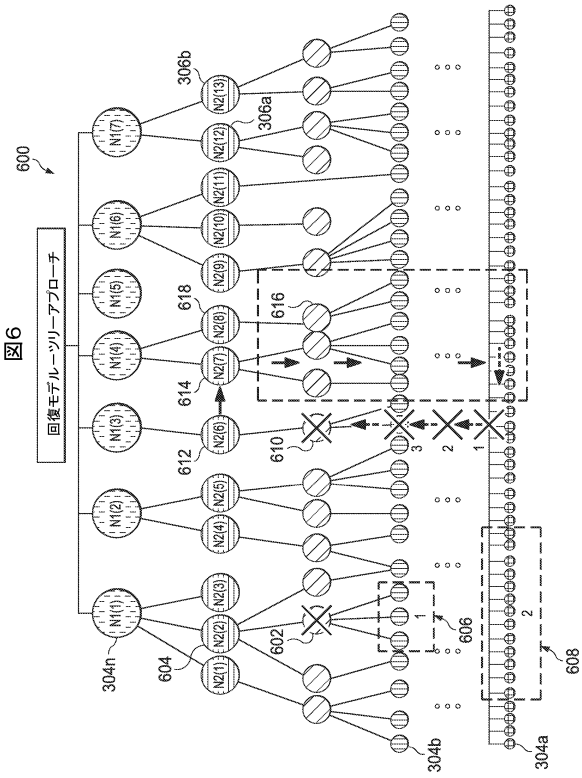
【図４Ｂ】


400b
障害間の平均時間 = MTBF = Σ(FS時間-RS時間) / 障害数
402
404
408の数
図4B

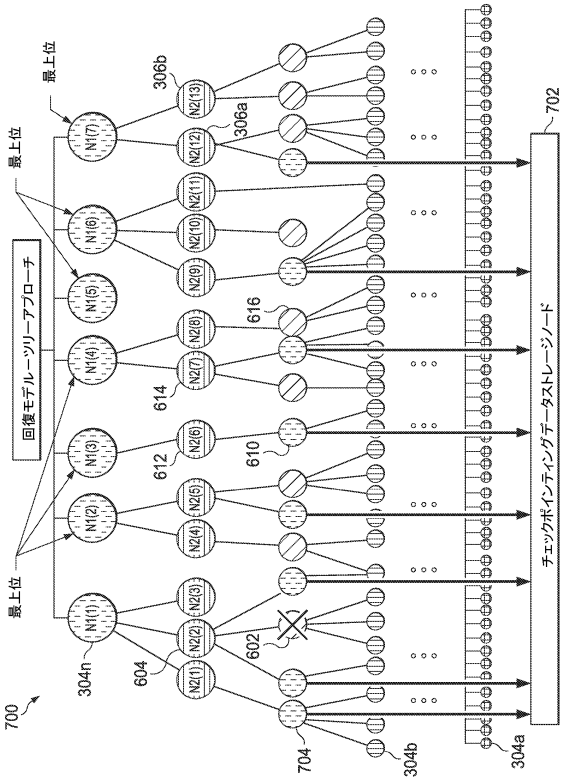
【図５】


500
504a
504b
504c
504d
MTBF
チェックポイント1
チェックポイント2
チェックポイント3
チェックポイント4
チェックポイント6
502a
502b
502c
502d
チェックポイント間隔
図5

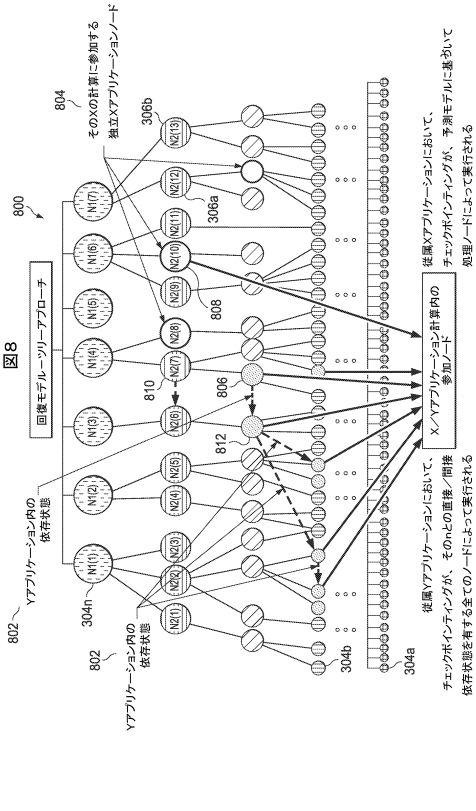
【図６】



【図７】



【図８】



【図９】

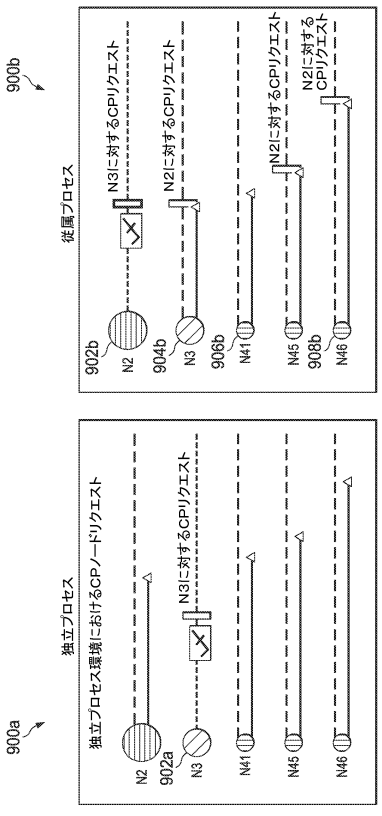


図9A

図9B

【図１０】

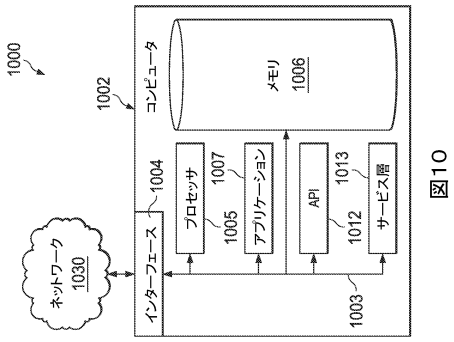

1000
1002
コンピュータ
メモリ
1006
1004
インターフェース
プロセッサ
1005
1007
アプリケーション
API
1012
1013
サービス層
1003
ネットワーク
1030

図10

フロントページの続き

審査官　清木　泰

(56)参考文献　米国特許出願公開第２０１０／０３１８８３７（ＵＳ，Ａ１）
米国特許出願公開第２０１０／００８８４９４（ＵＳ，Ａ１）
特開平０５－２１６８４５（ＪＰ，Ａ）
特開２００７－２１３６７０（ＪＰ，Ａ）
特開２００６－２５１９９９（ＪＰ，Ａ）
特開２００６－１７２０６５（ＪＰ，Ａ）
米国特許出願公開第２０１０／００１１２５４（ＵＳ，Ａ１）
米国特許出願公開第２００２／００８７９１３（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ１１／１６　－１１／２０
Ｇ０６Ｆ１１／１４
Ｇ０６Ｆ１１／０７
Ｇ０６Ｆ１１／２８　－１１／３６
Ｇ０６Ｆ１５／１６　－１５／１７７
Ｇ０６Ｆ　９／４５５－　９／５４