

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2017-527893

(P2017-527893A)

(43) 公表日 平成29年9月21日(2017.9.21)

(51) Int. Cl.	F I	テーマコード (参考)
G06F 11/20 (2006.01)	G06F 11/20 630	5B034
G06F 9/50 (2006.01)	G06F 9/46 465Z	5B042
G06F 11/07 (2006.01)	G06F 11/20 633	
	G06F 11/07 193	
	G06F 11/07 140A	

審査請求 未請求 予備審査請求 未請求 (全 31 頁) 最終頁に続く

(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)
 (85) 翻訳文提出日 平成29年3月30日 (2017. 3. 30)
 (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)
 (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 米国 (US)

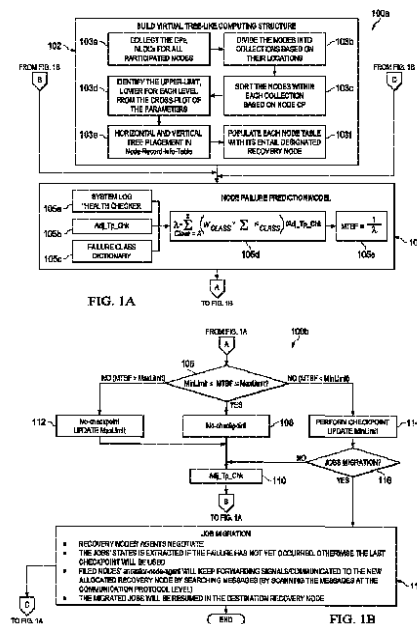
(71) 出願人 506018363
 サウジ アラビアン オイル カンパニー
 サウジアラビア国 31311 ダーラン
 、 イースタン アベニュー 1
 (74) 代理人 100097320
 弁理士 宮川 貞二
 (74) 代理人 100155192
 弁理士 金子 美代子
 (74) 代理人 100131820
 弁理士 金井 俊幸
 (74) 代理人 100100398
 弁理士 柴田 茂夫
 (72) 発明者 アル-ワハビ, ハリド エス.
 サウジアラビア王国 ダーラン 3131
 1, ピー. オー. ボックス 8166
 最終頁に続く

(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) 【要約】

本開示は、概して、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。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こと、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こと、計算したMTBFと最大及び最小閾値との間の比較に基づいて、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こと、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こと、および、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ことを含む。

【選択図】 図 1 A



【特許請求の範囲】**【請求項 1】**

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。

【請求項 2】

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、

請求項 1 の方法。

【請求項 3】

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、

請求項 2 の方法。

【請求項 4】

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、

請求項 3 の方法。

【請求項 5】

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、

請求項 1 の方法。

【請求項 6】

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、

請求項 1 の方法。

【請求項 7】

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；

10

20

30

40

50

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；

請求項 6 の方法。

【請求項 8】

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。

【請求項 9】

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、

請求項 8 の媒体。

【請求項 10】

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、

請求項 9 の媒体。

【請求項 11】

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、

請求項 10 の媒体。

【請求項 12】

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、

請求項 8 の媒体。

【請求項 13】

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

10

20

30

40

50

新する；命令をさらに含む、
請求項 8 の媒体。

【請求項 14】

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、
請求項 13 の媒体。

【請求項 15】

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。

【請求項 16】

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、

請求項 15 のシステム。

【請求項 17】

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、

請求項 16 のシステム。

【請求項 18】

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、

請求項 17 のシステム。

【請求項 19】

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、

請求項 15 のシステム。

【請求項 20】

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

10

20

30

40

50

トを作成する；

前記 M T B F 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、

請求項 15 のシステム。

【発明の詳細な説明】

【技術分野】

10

【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トリ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。

20

30

【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。

40

【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

50

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。

10

【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。

20

【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。

30

【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。

40

【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、特定の实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

50

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トニ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。

10

【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る。

【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。

20

【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。

30

【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。

40

【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トニ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トニ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

50

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。

10

【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、特定の实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。

20

【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一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トリングすることを可能にするかどうかの選択を伴う。

30

【0029】

チェックポイントリングは、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。チェックポイントリングにおいて、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、チェックポイントリングは、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、チェックポイントリ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。

40

【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

50

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、メモリアー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トリングの使用を介して、チェックポイントリング待ち時間（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。

10

【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には高性能、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、障害発生の場合に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。

20

30

【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回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-サーバの回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)-サーバの回答が不正確;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。

40

【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;. . .

50

；N n)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

10

20

30

40

50

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ノ又はバージョン、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】

103 a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）、及びノ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。103 aから、方法100 aは103 bに進む。

【0044】

103 b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。103 bから、方法100 aは103 cに進む。

【0045】

103 c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。103 cから、方法100 aは103 dに進む。

【0046】

103 d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一つに過ぎない。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

10

20

30

40

50

た左又は右 - 図 3 下部において、ノード 3 0 4 b 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。1 0 3 d から、方法 1 0 0 a は 1 0 3 e に進む。

【 0 0 4 9 】

1 0 3 e 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。1 0 3 e から、方法 1 0 0 a は 1 0 3 f に進む。

【 0 0 5 0 】

1 0 3 f 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。1 0 3 f から、方法 1 0 0 a は 1 0 4 (ノード障害予測モデル) に進む。

【 0 0 5 1 】

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 1 0 3 a ~ 1 0 3 f 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 3 0 0 を示している。上記の 1 0 3 b で説明したように、3 0 2 a、3 0 2 b、・・・、3 0 2 n は、位置 (N L O C)、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 1 0 3 c 及び 1 0 3 d で説明したように、3 0 4 a、3 0 4 b、・・・、3 0 4 n は、C P 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 3 0 4 a は、ノード 3 0 4 n よりも高い C P パラメータ値を有していてもよい。更に、ノード (例えば、3 0 6 a 及び 3 0 6 b) 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 I D) 値例が、いくつかのノードの内部 (例えば、3 0 4 n はレベル 1、ノード 1 のために「N 1 (1)」を表示し、ノード 3 0 6 b はレベル 2、ノード 1 3 のために「N 2 (1 3)」を表示する) 等に (N 2 の下のノード I D 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。

【 0 0 5 2 】

仮想ツリー状構造のトポロジ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 C P 値は変化し、ノードはより高い C P / メモリモデル等のためにスワップされ、その N L O C 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 N L O C 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。

【 0 0 5 3 】

図 4 A に目を向けると、図 4 A は、実装に従って M T B F の計算に用いられるノード性

10

20

30

40

50

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。

10

【0055】

図 1B において、プロセス状態をチェックポイントニングするか、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（Min Limit）及び最大閾値（Max Limit）と比較される。最初に、Min Limit 及び Max Limit は、いくつかの所定の時間値に設定される。Min Limit 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。Max Limit 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。

20

【0057】

106 において、MTBF が Min Limit と Max Limit との間（例えば、Min Limit 以上で、Max Limit 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。

30

【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か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。

40

【0059】

仮に 106 において、MTBF が Max Limit 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、Max Limit 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 Max Limit は、Max Limit 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 Min Limit よりも小さい場合、方法 100b は

50

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。

10

【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ことを確実にするためである。

20

【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。

30

【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。

40

【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。

50

【 0 0 6 8 】

1 1 9 e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。1 1 9 eから、方法1 0 0 cは1 1 9 fに進む。

【 0 0 6 9 】

1 1 9 f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。1 1 9 fから、方法1 0 0 cは1 1 9 gに進む。

10

【 0 0 7 0 】

1 1 9 g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。1 1 9 gから、方法1 0 0 cは停止する。図1 Bに戻ると、1 1 8から特定のダウンノードのために方法1 1 8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1 A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）。

20

【 0 0 7 1 】

図1 Aに戻ると、1 0 4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3 0 4 bが存在するレベルのいずれかのノードもノード3 0 4 b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。

30

【 0 0 7 2 】

1 0 4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1 0 5 a（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1 0 5 b（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1 0 5 b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1 0 5 c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）。

40

【 0 0 7 3 】

1 0 5 d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

50

、それらの障害種類頻度は、特定の期間 t_p 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。105 e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。

10

【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に、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に、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。

20

【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であってもよい-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に、(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。

30

40

【0077】

50

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ト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ト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ト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ト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ト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、チェックポイント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トデータストレージノードの故障が多額のチェックポイントデータの損失の原因とならないように、チェックポイント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ト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。

10

20

30

40

50

【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場合、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ト活動を実行する。

【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 9 A 及び 9 B を参照すると、図 9 A 及び 9 B は、実装により、それぞれ独立及び従属プロセス 900 a 及び 900 b に関するチェックポイントリクエストを示している。図 9 A 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 9 B のような従属プロセス N2 902 b の場合において、一旦チェックポイントリクエストを受信されると、プロセス N2 902 b 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 904 b に渡す。従属プロセス 904 B 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 906 b 及び N46 908 b）、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 904 b）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 906 b 及び N46 908 b）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ト操作が完了する時を通知できる。

10

【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、EDCS 1000 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。

20

【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、キーボード、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。

30

【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。

40

【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

50

、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回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。

10

【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JAVAX(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。

20

30

【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の特定の实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。

40

【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、

50

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。

10

【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、チェックポイントアプリケーション、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。

20

【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か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。

30

【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、機械生成される電氣的、光学的、又は電磁的符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。

40

【0095】

50

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、読み出し専用メモリ（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。

10

20

【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利用されてもよく；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回答して、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。

30

【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。

40

【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

50

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。

10

20

30

40

50

【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、特定の発明の特定の实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征も、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

【 図 1 C 】

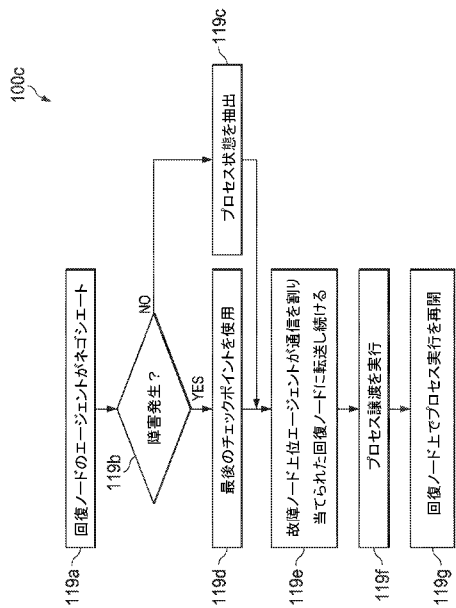


図 1C

【 図 2 】

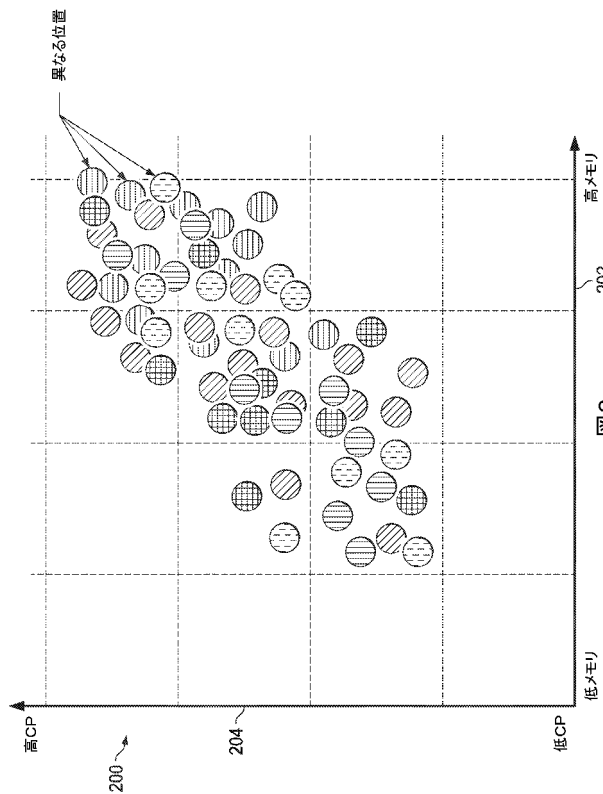


図 2

【 図 3 】

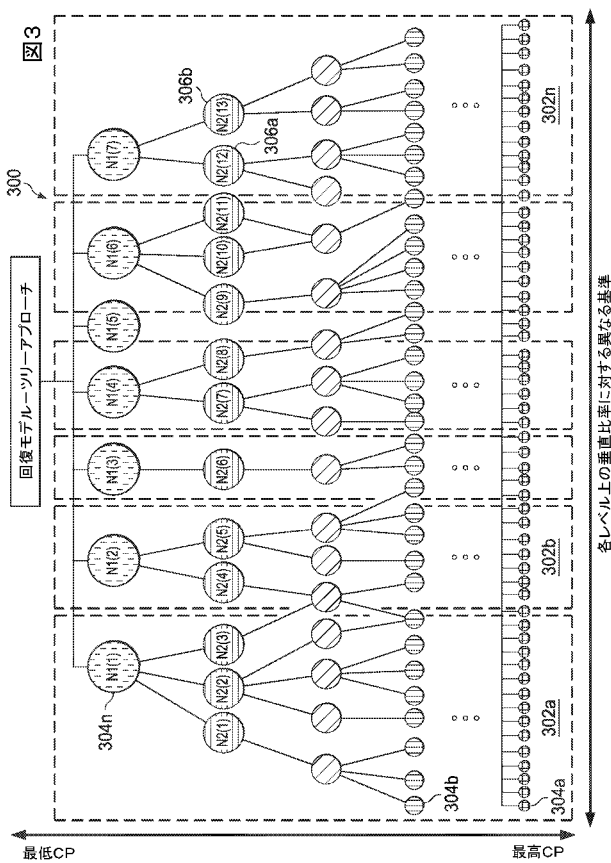


図 3

【 図 4 A 】

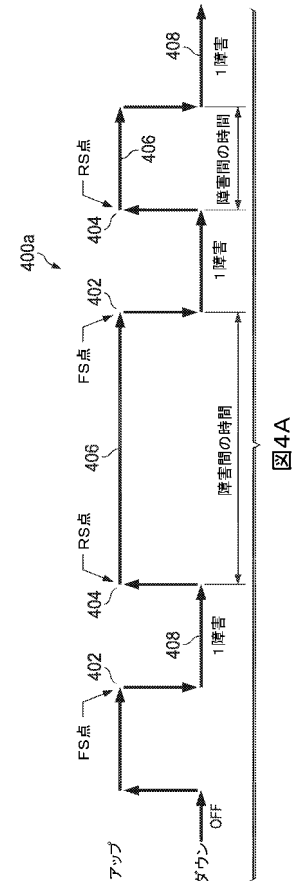
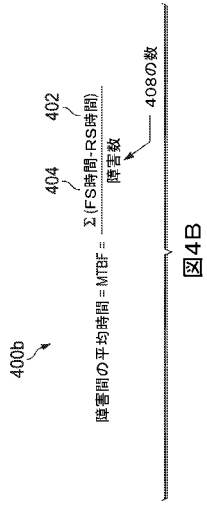
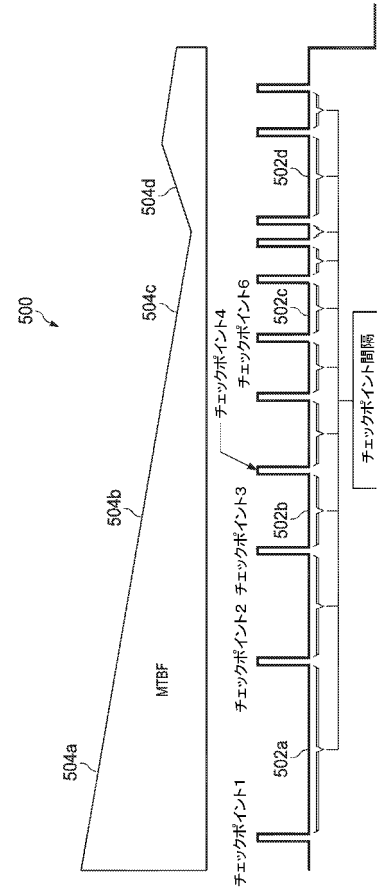


図 4A

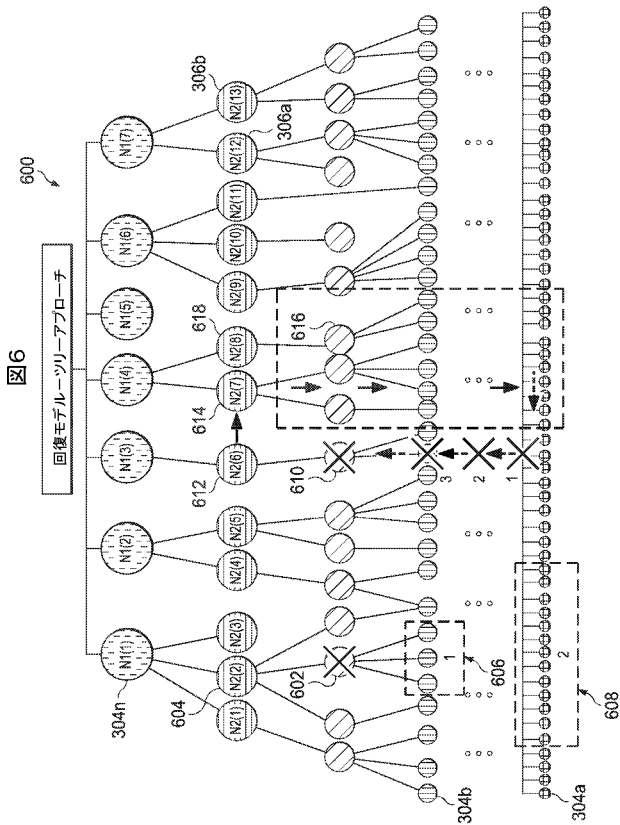
【 図 4 B 】



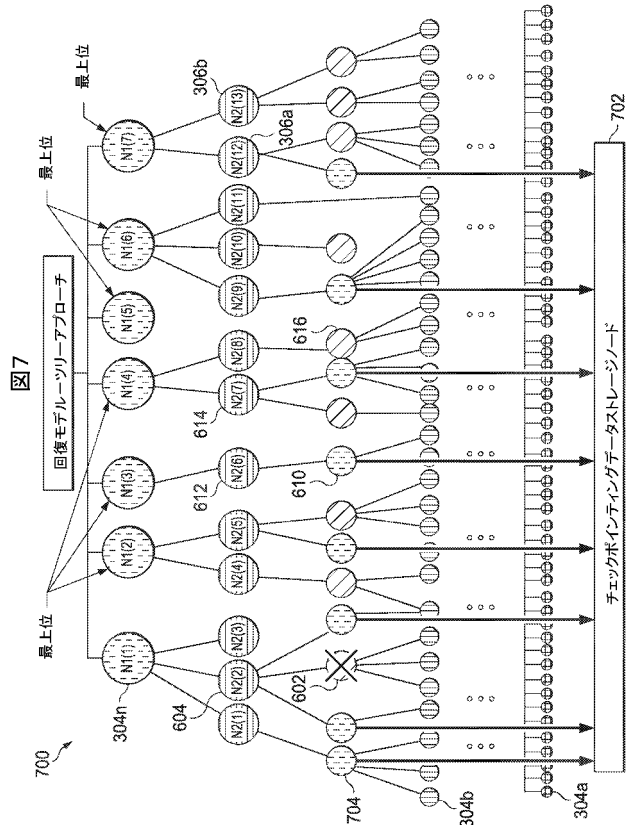
【 図 5 】



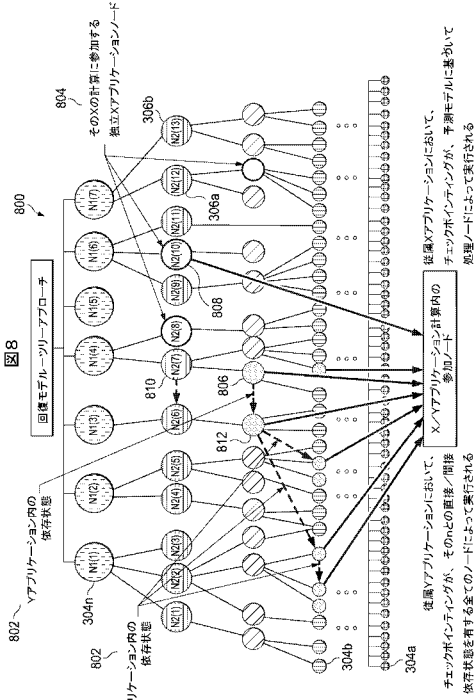
【 図 6 】



【 図 7 】



【 図 8 】



【 図 10 】

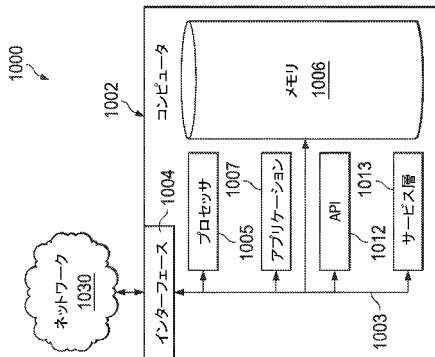


図10

【 図 9 】

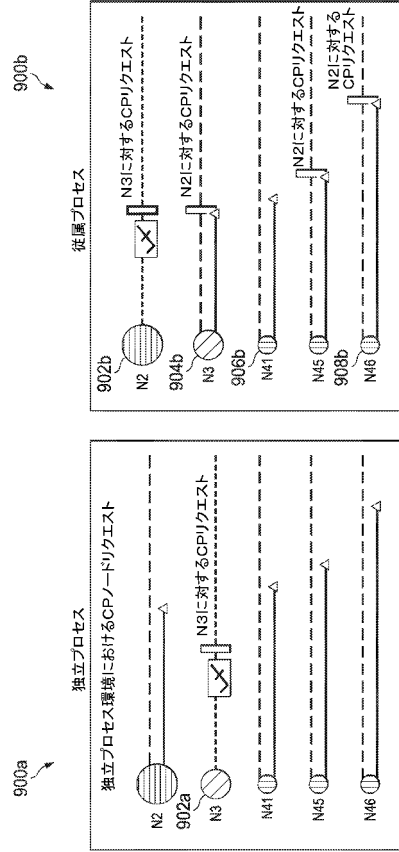


図9B

図9A

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/041121

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F11/14 ADD.		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EP0-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2002/087913 A1 (HARPER RICHARD EDWIN [US] ET AL) 4 July 2002 (2002-07-04) abstract; figure 4 paragraphs [0013], [0025], [0027] - [0030], [0035], [0037], [0039] - [0041] -----	1-20
X	US 2010/011254 A1 (VOTTA LAWRENCE G [US] ET AL) 14 January 2010 (2010-01-14) abstract paragraphs [0008], [0010], [0011], [0017], [0039], [0040], [0049], [0056], [0062] - [0066], [0080], [0085], [0088] -----	1-20
A	WO 2011/080910 A2 (NEC CORP [JP]; SUN WEI [JP]) 7 July 2011 (2011-07-07) abstract ----- -/--	1-20
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C.		<input checked="" type="checkbox"/> See patent family annex.
* Special categories of cited documents :		
<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>		<p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>
Date of the actual completion of the international search 2 November 2015		Date of mailing of the international search report 10/11/2015
Name and mailing address of the ISA/ European Patent Office, P.B. 6818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Leuridan, Koen

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/041121

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2011/296241 A1 (ELNOZAHY ELMOOTAZBELLAH NABIL [US]) 1 December 2011 (2011-12-01) abstract -----	1-20

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2015/041121

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2002087913 A1	04-07-2002	NONE	
US 2010011254 A1	14-01-2010	NONE	
WO 2011080910 A2	07-07-2011	JP 5672304 B2 JP 2013516665 A US 2012303998 A1 WO 2011080910 A2	18-02-2015 13-05-2013 29-11-2012 07-07-2011
US 2011296241 A1	01-12-2011	US 2011296241 A1 US 2012226939 A1	01-12-2011 06-09-2012

フロントページの続き

(51) Int.Cl. F I テーマコード(参考)
G 0 6 F 11/07 1 6 0

(81) 指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US

(特許庁注：以下のものは登録商標)

1 . L i n u x

Fターム(参考) 5B034 BB05 CC01 DD02
5B042 GA12 GA34 GA39 JJ06 JJ15 JJ29 KK01 KK06 LA08 MA08
MC22 MC30