

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5486687号
(P5486687)

(45) 発行日 平成26年5月7日(2014.5.7)

(24) 登録日 平成26年2月28日(2014.2.28)

(51) Int.Cl. F I
G 0 6 F 9 / 5 0 (2 0 0 6 . 0 1) G O 6 F 9 / 4 6 4 6 2 A

請求項の数 15 (全 61 頁)

(21) 出願番号	特願2012-531096 (P2012-531096)	(73) 特許権者	506329306
(86) (22) 出願日	平成22年9月27日 (2010. 9. 27)		アマゾン テクノロジーズ インコーポレ イテッド
(65) 公表番号	特表2013-505519 (P2013-505519A)		アメリカ合衆国 89507 ネバダ州
(43) 公表日	平成25年2月14日 (2013. 2. 14)		レノ ビーオー ボックス 8102
(86) 国際出願番号	PCT/US2010/050351	(74) 代理人	110001243
(87) 国際公開番号	W02011/041253		特許業務法人 谷・阿部特許事務所
(87) 国際公開日	平成23年4月7日 (2011. 4. 7)	(72) 発明者	アレックス マクリノフスキー
審査請求日	平成24年3月22日 (2012. 3. 22)		アメリカ合衆国 98109 ワシントン 州 シアトル テリー アベニュー ノー ス 410
(31) 優先権主張番号	12/569, 744	(72) 発明者	ブレイク メイケ
(32) 優先日	平成21年9月29日 (2009. 9. 29)		アメリカ合衆国 98109 ワシントン 州 シアトル テリー アベニュー ノー ス 410
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	12/569, 723		
(32) 優先日	平成21年9月29日 (2009. 9. 29)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 因果関係のプログラム実行容量修正への結論付け、およびプログラム実行容量の動的な修正

(57) 【特許請求の範囲】

【請求項1】

ユーザに提供されるプログラム実行容量における動的な修正の因果関係 (c a u s a l i t y) を判別するためにコンピュータで実装される方法であって、前記方法は、

複数のユーザによって使用されるためのプログラム実行サービスを提供するように構成される1つ以上のコンピューティングシステムの制御下であって、前記プログラム実行サービスの前記ユーザのプログラムを実行するように構成可能な複数のコンピューティングノードを前記プログラム実行サービスが有する、当該制御下において、

前記プログラム実行サービスの第1のユーザに提供されているプログラム実行容量に対する自動化された修正の開始において使用するための1つ以上の容量修正トリガを判別することであって、前記容量修正トリガの各々は、前記容量修正トリガが満たされるかどうかの判別において使用するための1つ以上の基準を含み、前記容量修正トリガが満たされる場合にリクエストされるプログラム実行容量修正の指定された種類を含む、判別することと、

10

前記第1のユーザの初期の望ましいプログラム実行容量の提供において使用するための、多様な前記複数のコンピューティングノードの第1の群を自動的に判別することと、前記第1の群の前記コンピューティングノードを、前記第1のユーザについて、各々1つ以上のソフトウェアプログラムを実行するために、第1の時刻において利用可能にすることと、

前記第1の時刻とその後の第2の時刻との間の期間において生じ、かつ、各々、前記第

20

1の群の前記複数のコンピューティングノードによって提供される前記プログラム実行容量の修正を生じさせることができる、複数の独立したイベントを自動的に識別することであって、前記複数のイベントは、前記第1の群によって提供される前記プログラム実行容量の指定された修正を要求するために、満たされると判別される前記容量修正トリガのうちの少なくとも1つ、および前記第1の時刻の後に前記第1のユーザによって動的に指定される少なくとも1つの容量修正命令を含む、識別することと、

前記第2の時刻において、前記第1の群の前記コンピューティングノードから前記第1のユーザに対して利用可能な実際のプログラム実行容量を自動的に判別することであって、前記第2の時刻における前記実際のプログラム実行容量は、前記第1の時刻と前記第2の時刻との間の期間において生じた前記第1の群の1つ以上のコンピューティングノード 10
の利用可能性に対する1つ以上の変更に、少なくとも一部が基づき、前記第1の時刻において前記第1のユーザに提供された前記望ましいプログラム実行容量とは別である、判別することと、

前記第1の群の前記1つ以上のコンピューティングノードの前記利用可能性の変更の各々について、前記識別されたイベントのうちの1つ以上を、前記利用可能性の変更の原因として自動的に結論付けることであって、前記利用可能性の変更のうちの少なくとも1つは、互いに独立している複数のイベントの組み合わせであり、かつ、満たされていると判別される、少なくとも1つの指定された容量修正トリガのうちの1つ以上を含む、原因を有する、結論付けることと、及び

その結論付けられる原因に対応して1つ以上の追加の動作が実行されることを可能にするように、前記利用可能性の変更のうちの1つ以上の各々について、前記結論付けられる原因の表示を提供することと

を含む方法。

【請求項2】

前記少なくとも1つの利用可能性の変更のうちの1つの前記原因として結論付けられる複数のイベントの前記組み合わせは、前記少なくとも1つの指定された容量修正命令のうちの1つ以上を含む、請求項1に記載の方法。

【請求項3】

前記実際のプログラム実行容量を前記自動的に判別することは、生じている複数の利用可能性の変更を判別することを含み、前記利用可能性の変更の各々の前記原因として前記1つ以上のイベントを自動的に結論付けることは、各々、前記複数の利用可能性の変更のうちの1つの単一の原因である、前記複数のイベントのうちの1つ以上を識別することと、単一の原因を有さない前記複数の利用可能性の変更の互いの前記組み合わせの一部として、前記複数の利用可能性の変更のうちの前記1つの単一の原因ではない前記複数のイベントの全てを総計することと、を含む、請求項1に記載の方法。

【請求項4】

前記第2の時刻後に生じる1つ以上の追加の期間の各々について、追加の期間において生じる利用可能性の変更の原因として、1つ以上の追加の識別されたイベントを自動的に結論付ける (a t t r i b u t i n g) ことをさらに含み、1つ以上の追加の期間のうちの1つにおいて生じた利用可能性の変更のうちの1つの原因である追加の識別されたイベントのうちの1つについて、前記1つの追加の識別されたイベントは、前記第1の時刻と前記第2の時刻との間の期間において生じた利用可能性の変更のうちの1つの結果である、請求項1に記載の方法。

【請求項5】

ユーザのためのプログラム実行容量内の動的修正の因果関係 (c a u s a l i t y) を判別するように構成されるコンピューティングシステムであって、

1つ以上のプロセッサと、

前記1つ以上のプロセッサのうちの少なくとも1つによって実行されると、前記複数のユーザの各々について、

前記ユーザの代わりに、1つ以上のソフトウェアプログラムを実行するために使用する

10

20

30

40

50

ために提供される初期のプログラム実行容量を判別することと、前記ユーザに提供されている前記プログラム実行容量へ、自動化された修正を後で開始するために使用するための1つ以上の容量修正トリガを判別することと、

前記ユーザへ前記初期の望ましいプログラム実行容量を提供するために使用するために、第1の時刻において前記ユーザに複数の利用可能なコンピューティングノードの群を自動的に関連付けることと、

前記第1の時刻後に、

前記第1の時刻と後の第2の時刻との間で生じ、かつ、各々、前記群の前記プログラム実行容量の関連付けられた望ましい修正を有する、複数の独立したイベントを自動的に識別することと、前記複数のイベントは、満たされると判別されている前記容量修正トリガのうちの少なくとも1つを含む、自動的に識別することと、

前記群によって提供された前記プログラム実行容量に影響する前記群の1つ以上のコンピューティングノードの利用可能性に対して、1つ以上の変更を自動的に判別することと、

前記群の前記判別された利用可能性の変更の各々に、1つ以上の原因を自動的に結論付けることと、前記利用可能性の変更のうちの少なくとも1つに結論付けられる前記1つ以上の原因は、前記識別された独立したイベントのうちの複数の組み合わせであり、前記利用可能性の変更のうちの少なくとも1つの他の変更に結論付けられる前記1つ以上の原因は、前記識別された独立したイベントのうちの単一の1つである、結論付けることと、

前記群の前記判別された利用可能性の変更のうちの1つ以上の前記結論付けられる1つ以上の原因の表示を提供することと、によってネットワークでアクセス可能なサービスの複数のユーザのプログラム実行容量を管理するように構成されるシステムマネージャーモジュールと

を備えるコンピューティングシステム。

【請求項6】

前記複数のユーザのうちの1人のための前記複数の独立したイベントは、前記第1の時刻の後に、前記1人のユーザによって動的に指定される少なくとも1つの容量修正命令をさらに含み、前記システムマネージャーモジュールは、前記少なくとも1つの満たされた容量修正トリガと、前記1人のユーザによって指定された前記少なくとも1つの容量修正命令との前記望ましい修正の総計 (a g g r e g a t i n g) に基づいて、前記1人のユーザの前記群の前記プログラム実行容量を増加させるために、単一の総計された修正を自動的に判別し、かつ、前記第2の時刻の前に前記1人のユーザの前記群へ前記判別された単一の総計された修正を開始するようにさらに構成され、前記判別された単一の総計された修正は、前記1人のユーザのための前記少なくとも1つの利用可能性の変更のうちの1つであり、前記判別された単一の総計された修正の前記1つ以上の原因として結論付けられる前記複数の独立したイベントの前記組み合わせは、前記少なくとも1つの満たされた容量修正トリガと、前記1人のユーザによって指定された前記少なくとも1つの容量修正命令とを含む、請求項5に記載のコンピューティングシステム。

【請求項7】

前記1人のユーザのための前記判別された容量修正トリガは前記1人のユーザによって指定され、前記ネットワークでアクセス可能なサービスは、前記プログラム実行サービスのリモートユーザのプログラムを実行するように構成可能である複数のコンピューティングノードを提供するプログラム実行サービスであり、前記複数のユーザのうちの少なくとも1人の各々の前記群の前記複数のコンピューティングノードは、前記複数のコンピューティングノードのサブセットである、請求項6に記載のコンピューティングシステム。

【請求項8】

ユーザのプログラム実行容量を動的に修正するためのコンピュータで実装可能な方法であって、前記方法は、

複数のユーザが使用するためのプログラム実行サービスを提供するように構成される1

10

20

30

40

50

つ以上のコンピューティングシステムの制御下であって、前記プログラム実行サービスの前記ユーザのプログラムを実行するように構成可能な複数のコンピューティングノードを前記プログラム実行サービスが有する、当該制御下において、

前記プログラム実行サービスの第1のユーザの代わりに1つ以上のソフトウェアプログラムを実行するために使用するための初期の望ましいプログラム実行容量を指定し、かつ、前記望ましいプログラム実行容量に対する自動化された修正を後で開始するために使用するための1つ以上の容量修正トリガを指定する、情報を受信することであって、前記容量修正トリガの各々は、前記容量修正トリガが満たされているかどうかを判別するために使用するための1つ以上の基準を含み、かつ、前記容量修正トリガが満たされる場合にリクエストされる前記望ましいプログラム実行容量への指定された種類の修正を含む、受信

10

することと、
前記第1のユーザの前記初期の望ましいプログラム実行容量を提供するために使用するための前記複数のコンピューティングノードの複数の第1の群を自動的に判別することと、第1の時刻において利用可能な前記第1の群の前記コンピューティングノードを、前記第1のユーザのために1つ以上のソフトウェアプログラムを各々実行させることと、前記第1の時刻において前記第1のユーザに利用可能であると記録される公的プログラム実行容量として、前記初期の望ましいプログラム実行容量を指定することと、

前記第1の時刻の後、前記1つ以上のソフトウェアプログラムの前記実行することに関するこれらのコンピューティングノードの性能特性の1つ以上のタイプの現行の値を判別するために、前記第1の群の前記コンピューティングノードを監視することであって、前記監視することは、前記第1の時刻後に、後の第2の時刻において、前記第1の群の前記コンピューティングノードから前記第1のユーザへ利用可能な実際のプログラム実行容量を判別することを含む、監視することと、

20

前記第1の時刻と前記第2の時刻との間の期間の間に生じる複数のイベントに基づいて、前記第1のユーザの前記望ましいプログラム実行容量を修正するために自動的に判別することであって、前記複数のイベントは、前記1つ以上の性能特性タイプの前記判別された現行の値の一部に基づいて、満たされることが判別されている前記指定された容量修正トリガのうち少なくとも1つを含み、かつ、前記第1の時刻の後に前記第1のユーザによって動的に指定される少なくとも1つの容量修正命令を含み、前記少なくとも1つの指定された容量修正トリガおよび前記少なくとも1つの容量修正命令の各々は、前記望ましい

30

プログラム実行容量への指定された修正を示す、判別することと、及び
前記第2の時刻で前記判別された実際のプログラム実行容量と、前記修正された望ましいプログラム実行容量とに、前記第1のユーザに利用可能な前記記録された公的プログラム実行容量を自動的に調和させることであって、前記修正されたコンピューティングノードが、前記修正された望ましいプログラム実行容量を提供するために利用可能であり、かつ、前記記録された公的プログラム実行容量が、前記提供された修正された望ましいプログラム実行容量を反映するように更新されるように、前記記録された公的プログラム実行容量を前記調和させることは、前記判別された実際のプログラム実行容量および前記修正された望ましいプログラム実行容量に従い、前記第1の群の一部である前記コンピューティングノードを自動的に修正することを含み、さらに、前記第2の時刻における前記望ましいプログラム実行容量と一致するように前記第2の時刻における前記第1のユーザが利用可能であるとして記録される公的プログラム実行容量を更新することであって、前記第2の時刻における前記実際のプログラム実行容量が前記第2の時刻における前記公的プログラム実行容量とは別である場合に、前記第2の時刻における前記望ましいプログラム実行容量と一致するように前記第2の時刻における前記実際のプログラム実行容量を更新することを含む、調和させることと

40

を含む方法。

【請求項9】

前記初期の望ましいプログラム実行容量が初期の望ましいコンピューティングノードの数となるように、前記第1の群の前記プログラム実行容量は、前記第1の群の一部である

50

コンピューティングノードの数に基づいて計測され、前記第2の時刻の前記判別された実際のプログラム実行容量は、前記第2の時刻に前記第1の群の一部として利用可能である実際のコンピューティングノードの数になり、前記望ましいプログラム実行容量を修正するために前記自動的に判別することは、前記第1の群のうちのコンピューティングノードの修正された望ましい量として、前記修正された望ましいプログラム実行容量を判別することを含み、前記第1の群の一部である前記コンピューティングノードを前記自動的に修正することは、前記修正された望ましい量であるように前記第1の群の一部である前記コンピューティングノードの前記実際の数を変更させることを含む、請求項8に記載の方法。

【請求項10】

10

前記第1の群の前記プログラム実行容量は、前記第1の群の一部である前記コンピューティングノードから利用可能なコンピューティングリソースの複数のタイプの各々を総計することに基づいて計測され、前記第1の群の一部である前記コンピューティングノードを前記自動的に修正することは、前記コンピューティングノードを前記自動的に修正後に、前記第1の群の一部である前記コンピューティングノードから利用可能なコンピューティングリソースの前記複数のタイプのうちの少なくとも1つの数の総計量を変更させることを含む、請求項8に記載の方法。

【請求項11】

前記少なくとも1つの指定された容量修正トリガの各々の前記1つ以上の基準は、前記1つ以上の性能特性タイプの少なくとも1つのための1つ以上の指定された値に基づき、前記方法は、前記少なくとも1つの指定された容量修正トリガのうちの各々についての前記1つ以上の指定された値と、前記1つ以上の性能特性タイプの前記判別された現行の値との間の一致に少なくとも一部基づいて、前記少なくとも1つの指定された容量修正トリガが満たされることを自動的に判別することをさらに含む、請求項8に記載の方法。

20

【請求項12】

前記第1の群の前記修正された望ましいプログラム実行容量を前記判別することは、前記少なくとも1つの指定された容量修正トリガおよび前記少なくとも1つの容量修正命令の各々によって指定される前記望ましいプログラム実行容量への前記示された修正を総計することと、前記望ましいプログラム実行容量への前記総計された示された修正に基づいて、前記判別された修正された望ましいプログラム実行容量を指定することと、を含む、請求項8に記載の方法。

30

【請求項13】

ユーザのためにプログラム実行容量を動的に修正するように構成されたコンピューティングシステムであって、

1つ以上のプロセッサと、

前記1つ以上のプロセッサのうちの少なくとも1つによって実行されると、前記複数のユーザの各々について、

前記ユーザの代わりに、1つ以上のソフトウェアプログラムを実行するために使用するために望ましいプログラム実行容量を判別することと、前記ユーザについて、前記望ましいプログラム実行容量への自動化された修正を後で開始するために使用するための1つ以上の容量修正トリガを判別することと、

40

前記ユーザへ前記判別された望ましいプログラム実行容量を提供するために使用するために、第1の時刻において、前記ユーザに、複数の利用可能なコンピューティングノードの群を自動的に関連付けることと、

前記第1の時刻後に、

前記第1の時刻後の後の第2の時刻において、前記群の前記コンピューティングノードから前記ユーザへ利用可能な実際のプログラム実行容量を自動的に判別することであって、前記第2の時刻の前記実際のプログラム実行容量は、前記第1の時刻に前記ユーザへ提供された前記望ましいプログラム実行容量とは別である、判別することと、

各々、前記第1の時刻後に生じ、かつ、各々、前記望ましいプログラム実行容量の関連

50

付けられた修正を有する複数の独立したイベントに基づいて、前記第2の時刻に前記ユーザに提供される修正された望ましいプログラム実行容量を自動的に判別することであって、前記複数の独立したイベントは、満たされると判別されている前記容量修正トリガのうちの少なくとも1つを含み、かつ、前記第1の時刻の後に前記ユーザによって動的に指定される少なくとも1つの容量修正命令を含み、前記第2の時刻で前記修正された望ましいプログラム実行容量は、前記複数の独立したイベントの前記関連付けられた修正の総計に基づく、判別することと、及び

前記第2の時刻の前記修正された望ましいプログラム実行容量を、前記第2の時刻の前記判別された実際のプログラム実行容量と調和させるように、前記第2の時刻の前記群の一部である前記コンピューティングノードを自動的に修正することであって、前記修正されたコンピューティングノードは、前記ユーザへ前記修正された望ましいプログラム実行容量を提供するために使用するためのものである、修正することと、によってネットワークでアクセス可能なサービスの複数のユーザについてプログラム実行容量を管理するように構成されるシステムマネージャーモジュールと

を備えるコンピューティングシステム。

【請求項14】

前記複数のユーザのうちの1人について前記判別された望ましいプログラム実行容量は、前記第1の時刻に前記1人のユーザに対して利用可能であると記録される初期の公的プログラム実行容量として識別され、かつ、前記1人のユーザによって示され、前記第2の時刻に前記1人のユーザに利用可能な前記実際のプログラム実行容量の前記自動的に判別することは、1つ以上の変更を識別するように、前記第1の時刻から前記第2の時刻への期間の間、前記1人のユーザについて前記群の前記コンピューティングノードを監視することに少なくとも部分的に基づき、前記コンピューティングノードの前記監視することは、前記1つ以上の変更を反映するために、前記1人のユーザに利用可能な前記記録された公的プログラム実行容量を更新することを含み、前記1人のユーザに利用可能な前記記録された公的プログラム実行容量は、前記第2の時刻で前記修正された望ましいプログラム実行容量の前記自動的に判別することに基づいて更新されず、前記第2の時刻で前記群の一部である前記コンピューティングノードの前記修正することは、前記第2の時刻で前記記録された公的プログラム実行容量を、前記第2の時刻で前記修正された望ましいプログラム実行容量と調和させるようにさらに実行され、前記記録された公的プログラム実行容量の前記調和させることは、前記1人のユーザに利用可能なさらに更新された記録された公的プログラム実行容量として、前記第2の時刻で前記修正された望ましいプログラム実行容量を識別することを含む、請求項13に記載のコンピューティングシステム。

【請求項15】

前記1人のユーザの前記判別された容量修正トリガは、前記1人のユーザによって指定され、前記ネットワークでアクセス可能なサービスは、プログラム実行サービスのリモートユーザのプログラムを実行するように構成可能な複数のコンピューティングノードを提供するプログラム実行サービスであり、前記複数のユーザのうちの少なくとも1人の各々についての前記群の前記複数のコンピューティングノードは、前記複数のコンピューティングノードのサブセットである、請求項13に記載のコンピューティングシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、因果関係のプログラム実行容量修正への結論付け (attributing)、およびプログラム実行容量の動的な修正に関する。

【背景技術】

【0002】

多くの企業およびその他の組織が、(例えばローカルネットワークの一部として)同一場所に配置される、または代わりに(例えば1つ以上の非公開または公開の中間ネットワークを介して接続する)複数の別々の地形的位置に配置されるコンピューティングシステ

10

20

30

40

50

ムによって等、それらの操作を支援するように、多数のコンピューティングシステムを相互接続するコンピュータネットワークを運用している。例えば、単一の組織によって、および、単一の組織の代わりに運営される非公開データセンタ、およびコンピューティングリソースを顧客に提供するためのビジネスとしてのエンティティによって運用される公開データセンタ等の非常に多くの数の相互接続されたコンピューティングシステムを収容するデータセンタが一般的になってきている。いくつかの公開データセンタ運用者は、ネットワークアクセス、電力、様々な顧客によって所有されるハードウェアの安全な設置施設を提供する一方で、他の公開データセンタ運用者は、さらに、その顧客が利用するために利用可能にされたハードウェアリソースを含む「フルサービス」施設を提供する。しかしながら、典型的なデータセンタの規模および範囲が増加するにつれて、物理的コンピューティングリソースを供給、運営および管理するタスクは、急速に複雑になってきている。

10

【0003】

コモディティハードウェアの仮想化技法の出現により、多様なニーズを有する多くのユーザのための大規模なコンピューティングリソースの管理に関していくつかの利点をもたらされたことにより、多様なコンピューティングリソースを複数のユーザが効率的かつ安全に共有することが可能になった。例えば、VMWare、XEN、Linux（登録商標）のKVM（「Kernelベースの仮想マシン」）またはUser-Mode Linuxによって提供されるもの等の仮想化技法は、単一の物理的コンピューティングマシンによってホストされる1つ以上の仮想マシンを、各ユーザに提供することにより、複数のユーザの間で単一の物理的コンピューティングマシンを共有させることを可能にしてもよく、各々のこうした仮想マシンは、多様な仮想マシンの間のアプリケーション分離およびセキュリティを提供しながら、ユーザに、彼らが所与のハードウェアコンピューティングリソースの単独の運用者および管理者であるという幻想を抱かせる、個別の論理的なコンピューティングシステムとして作用するソフトウェアシミュレーションである。

20

【図面の簡単な説明】

【0004】

【図1A】プログラム実行サービスの複数のユーザに利用可能なプログラム実行容量を管理するための相互作用の例示的实施形態を例示するネットワーク図である。

【図1B】プログラム実行サービスの複数のユーザに利用可能なプログラム実行容量を管理するための相互作用の例示的实施形態を例示するネットワーク図である。

30

【図2A】多様なタイミングかつ多様な様式で利用可能なプログラム実行容量を動的に修正する場合等、ユーザのために複数のコンピューティングノードの群のプログラム実行容量を管理する例を例示する。

【図2B】多様なタイミングかつ多様な様式で利用可能なプログラム実行容量を動的に修正する場合等、ユーザのために複数のコンピューティングノードの群のプログラム実行容量を管理する例を例示する。

【図3】複数のユーザに提供されるプログラム実行容量を管理するためのコンピューティングシステムの例示的实施形態を例示する模式図である。

【図4】プログラム実行サービスシステムマネージャルーチンの例示的实施形態のフロー図を例示する。

40

【図5】再帰的容量調和ルーチンの例示的实施形態のフロー図を例示する。

【図6】容量修正帰属ルーチンの例示的实施形態のフロー図を例示する。

【図7】プログラム実行サービス容量保守マネージャルーチンの例示的实施形態のフロー図を例示する。

【発明を実施するための形態】

【0005】

1人以上のユーザのためにプログラムを実行するために使用されるプログラム実行容量を管理するための技法を説明する。少なくともいくつかの実施形態において、管理されるプログラム実行容量は、1つ以上のプログラムを実行する際に、ユーザによる使用のために提供される1つ以上のコンピューティングノードの群を含む。加えて、ユーザに関連す

50

るコンピューティングノードの群は、群のうちのコンピューティングノードからユーザに利用可能なプログラム実行容量の量を管理するために使用中である間に、動的に修正される場合がある。ユーザに関連するコンピューティングノードの群に対する修正は、多様な実施形態において多様な形態（例えば、コンピューティングノードを動的に追加および/または削除すること等によって、群の中のコンピューティングノードの数を変更するために）を有する場合があり、多様な実施形態において多様な様式において（例えば、ユーザによって指定される動的命令に基づいて、ユーザによって以前に定義されたトリガが満たされたという自動判定に基づいて、群のうちのコンピューティングノードを提供しているサービスの自動操作に基づいて等）開始される場合がある。コンピューティングノードの群から利用可能なプログラム実行容量の動的修正に関する追加の詳細を以下に示す。加えて、少なくともいくつかの実施形態において、技法は、リモートユーザのプログラムを実行するために複数のコンピューティングノード（例えば、1つ以上の物理的コンピューティングシステム上でホストされる仮想マシンおよび/または複数の物理的コンピューティングシステム）を提供するネットワークからアクセス可能なプログラム実行サービス等、サービスの複数の顧客または他のユーザの代わりに複数のプログラムを実行するプログラム実行サービス（以下「PES」）と併用される場合がある。技法のいくつかまたは全てはまた、以下に詳細を記載するように、プログラム実行サービスシステムマネージャモジュールおよび/またはプログラム実行サービス容量保守マネージャモジュールによって自動的に実施される場合もある。

10

【0006】

20

上述のように、ユーザのために1つ以上のプログラムを実行している使用中のコンピューティングノードに対して実施される動的修正は、多様な形態を有する場合があり、多様な実施形態において多様な様式で開始される場合がある。一例として、コンピューティングノードの群のプログラム実行容量は、群の一部であるコンピューティングノードの数によって、少なくとも部分的に測定される場合があり、群のコンピューティングノード数を変更することによって（例えば、コンピューティングノード数を増加することによって、プログラム実行容量を増加するように、およびコンピューティングノード数を減少することによって、プログラム実行容量を減少するように）、修正される場合がある。そのようなコンピューティングノード数の修正は、例えば、コンピューティングノード数における所与の増減率が、群のコンピューティングリソースおよびプログラム実行容量の合計における同一または同様な増減率に対応する等、群の中のコンピューティングノードのいくつか、または全てが同一、または同様な量のコンピューティングリソース（例えば、メモリの量、ハードドライブ容量、CPU実行サイクル、ネットワーク帯域幅等）に対するアクセスを提供または有する状況において、使用される場合がある。他の実施形態において、群のうちのコンピューティングノードのいくつかまたは全ては、それらがアクセスを有するコンピューティングリソースの量に関して（例えば、2つ以上の別個の種類コンピューティングノード構成が使用される場合、各コンピューティングノードが他のコンピューティングノードとは独立的に構成される場合等）、またはそうでなければ、それらが提供するプログラム実行容量の種類に関して（例えば、専用のハードウェア、ソフトウェアプログラムの種類に基づいて等）、1つ以上の顕著な様式において異なる場合があるが、それにもかかわらず、群のための動的プログラム実行容量修正は、少なくとも部分的に、群のコンピューティングノードの数を修正することに基づく場合、またはその代わりに、他の部分に詳細を記載するように、他の様式（例えば、ある種類のコンピューティングノード構成から別の種類へ変更する）に基づく場合がある。

30

40

【0007】

さらに、少なくともいくつかの実施形態および状況においては、1つ以上のコンピューティングノードの群のプログラム実行容量は、群によって提供される1つ以上の種類のコンピューティングリソースの総計量（例えば、メモリの量、ハードドライブ容量、CPU実行サイクル、ネットワーク帯域幅等）に基づく等、コンピューティングノードの数以外の様式で測定され、修正される場合がある。加えて、少なくともいくつかの実施形態にお

50

いて、コンピューティングノードの群のプログラム実行容量を測定し、特定する際に、コンピューティングノードのいくつかまたは全ての地理的場所、コンピューティングノードのいくつかまたは全ての間の相互関係（例えば、多くても最大の地理的距離または少なくとも最小の地理的距離によって分類される、多くても最大のネットワークレイテンシまたは少なくとも最小のネットワークレイテンシによって分類される、同時に障害が発生する可能性が少ない、2つ以上の独立したデータセンタまたは他のコンピューティングノード集合体に分類される等）、専門のハードウェア性能および/またはソフトウェア性能の利用可能性等、追加の因子が考慮される場合がある。プログラム実行容量を測定および特定することに関して追加の詳細を以下に示す。

【0008】

ユーザのために1つ以上のプログラムを実行している使用中の群のコンピューティングノードを管理しているPESまたは他のシステムは、多様な様式において、コンピューティングノード群のための動的プログラム実行容量修正をどのように、いつ行うかを自動的に判定する場合がある。例えば、少なくともいくつかの実施形態および状況において、PESまたは他のシステムは、即時様式でいくつかの種類のプログラム実行容量修正を行う場合があり、一方で、他の種類のプログラム実行容量修正が、定期的またはそうでなければ再帰的様式（例えば、1つ以上の総計された修正の先行の実施以降等、期間中に行われるようにリクエスト、またはそうでなければ判定された複数の修正を延期し、総計するように）で実施される場合がある。複数のプログラム実行容量修正の判定が、期間にわたって総計される場合、総計された修正の判定に関する情報は、多様な様式において、プログラム実行容量修正の性能を向上させるために使用される場合がある。例えば、2つの判定されたプログラム実行容量修正が、反対の種類の修正（例えば、コンピューティングノードの数を増加/減少するように、利用可能な総計メモリを増加/減少するように等）に対応する場合、2つの修正は、相互を部分的にまたは完全に相殺するように選択されることによって、またはその代わりに、2つの修正のうちより高い優先度を実施し、他の修正を実施しないように選択することによって等、多様な様式で総計される場合がある。加えて、2つ以上の判定されたプログラム実行容量修正が、同様または補完的な種類の修正（例えば、コンピューティングノードの数を指定された量全て増加する、利用可能な総計メモリを指定された量全て増加する等）に対応する場合、それらの判定された修正は、多様な様式において、同様に総計される（例えば、最大、最小、最高の優先度を有するもの、最初に判定されたもの、最後に判定されたもの等、何らかの指定された条件を満たす単一の判定された修正を選択するように、多様な判定された修正を集積し、集積された修正量を使用する等）場合がある。多様な種類のプログラム実行容量修正をどのように、いつ行うかを判定することに関する追加の詳細を以下に示す。

【0009】

加えて、PESまたは他のシステムが、ユーザの代わりに1つ以上のプログラムを実行しているコンピュータノードの群に対してプログラム実行容量修正を動的に実施する時、PESまたは他のシステムは、特定のプログラム実行容量修正に対する因果関係情報または他の責任を結論付けるための多様な操作をさらに実施する場合がある。責任の帰属は、例えば、各々が動的プログラム実行容量修正を生じさせることが可能である、関連期間中に発生していた特定のイベントを識別すること、およびイベントのうちの一つ以上を、その期間中またはその後実施される動的プログラム実行容量修正のうちの一つかまたは全てに結論付けることを含む場合がある。例えば、いくつかの動的プログラム実行容量修正は各々、少なくともいくつかの実施形態および状況において単一の特定のイベントによって開始される場合がある（例えば、群のコンピューティングノードが障害が発生、またはそうでなければ利用不可能になり、システムが群のための置換コンピュータノードの提供を自動的に即時に開始し、コンピューティングノードが利用不可能であることが、自動化されたシステム動作に置換コンピューティングノードの提供を発生させる単一のイベントである場合）。他の動的プログラム実行容量修正は各々、少なくともいくつかの実施形態および状況において、各々容量修正の原因であった、または原因であったかもしれない

10

20

30

40

50

複数のイベントの組み合わせに結論付けられる場合がある（例えば、複数の独立したイベントが各々期間中に、群のコンピューティングノード数の増加をリクエストまたは指示し、期間の最後に単一のコンピューティングノード数の増加を実施するために総計し、多様な独立したイベントがその後合わせて、自動化されたシステム動作にコンピューティングノード数の増加を間接的に実施させる複数のイベントである場合）。

【 0 0 1 0 】

ユーザに関連するコンピューティングノード群に対する特定の動的プログラム実行容量修正の責任の帰属は、例えば、ユーザがプログラム実行容量修正のうちの少なくともいくつかに対して課金される時（例えば、P E Sまたは他のシステムが、群の各コンピューティングノードを提供することに対して、プログラム実行容量の量の各々の他の測定に対して、および/または他の原理に基づいて、ユーザ顧客に課金する有料システムである場合）のような状況において、特定の利益を有する場合がある。そのような状況において、ユーザは、特定の動的プログラム実行容量修正に対応する課金に関する情報だけではなく、ユーザがそれらの動的プログラム実行容量修正の原因および適切性を確認することを可能にするように関連の責任帰属情報も受信し、検討することが可能な場合がある。少なくともいくつかのそのような実施形態において、責任帰属情報は、ユーザが、人間可読情報の中に含まれる、P E Sまたは他のシステムによって多様な自動化された動作がなぜ行われたかの説明を理解することを可能にするように、ユーザに表示するための人間可読形式で生成される場合がある。そのような責任帰属情報はまた、P E Sまたは他のシステムによって他の操作を自動的に開始することを含めて、他の実施形態において、多様な他の様式で使用される場合がある。加えて、少なくともいくつかの実施形態において、責任帰属情報は、どのイベントが特定の指示されたプログラム実行容量修正、または群の1つ以上のコンピューティングノードの利用可能性における他の変更の原因であるか、および/またはどのプログラム実行容量修正または他のコンピューティングノード群の利用可能性の変更が1つ以上の指示されたイベントによって生じたかを識別するためのリクエスト等、ユーザまたは他の情報源から受信される多様な種類のクエリに対する応答において生成および/または使用される場合がある。動的プログラム実行容量修正に対する責任帰属情報を判定し、使用することに関する追加の詳細を以下に示す。

【 0 0 1 1 】

動的プログラム実行容量修正を自動的に管理するための前述の技法は、多様な状況において多様な利益を提供することができる。例えば、考慮される複数のリクエストまたは判定された動的プログラム実行容量修正を合わせて総計することによって、P E Sまたは他のシステムは、総計された修正がどのように実施されるかを最適化するとともに、追加の変更のためにコンピューティングノードのうちの一つかまたは全ての一時的な利用不可能の期間を生じる場合があるコンピュータノードに対して反復変更を最小限にすることが可能な場合があり、加えて、ユーザは、コンピューティングノードの群の性能特性に基づく多様な種類のトリガを予め定義する場合があり、プログラム実行容量は、特定のトリガが満たされるに応じて適切に自動的に増加または減少される（例えば、コンピューティングノード群のコンピューティング負荷の一時的増加に対処するために、コンピューティングノード群のプログラム実行容量を応答的に増加する、プログラム実行容量使用の反復パターンを示す特定の性能特性および/または履歴データにおける経時的なトレンドに基づいて等、追加のプログラム実行容量に対して予測される近い将来の必要性および/または既存のプログラム実行容量に対して予測される近い将来の不要性に対処するために、コンピューティングノード群のプログラム実行容量を積極的に増加または減少する等）。代替として、ユーザは、特定レベル（例えば、コンピューティングノードの特定の所望の定数）で、またはこの近辺でコンピューティングノードの群に対するプログラム実行容量を維持することを望む場合があり、多様な修正は、その特定レベルで利用可能なプログラム実行容量を維持するように（例えば、特定の所望数とは異なっていた実際のコンピューティングノード数を、その特定の所望数に戻すように）、コンピューティングノードの群に自動的に行われる場合がある。そのような技法は、例えば、群のコンピューティングノード

10

20

30

40

50

の各々が、同じプログラムの別個のコピーを実行し（例えば、群全体の総計コンピューティング負荷に対する代替として機能する）、コンピューティングノードの数が各コンピューティングノードによって処理される作業量を管理するように修正される時、あるいはそうでなければ、群の多様なコンピューティングノードが各々、同一プログラムの別個のコピーを実行しない状況において（例えば、いくつかのコンピューティングノードにアプリケーションサーバプログラムを実行させ、他のコンピューティングノードに関連のデータベースサーバプログラムを実行させる等、群の別個のサブセットのコンピューティングノードが各々異なるプログラムのコピーを実行する場合、コンピューティングノードのいくつかまたは全てが、分散様式のように、単一のプログラムの異なる部分を実施する場合等）、有用な場合がある。さらに、群に追加のコンピューティングノードを追加する時、PESまたは他のシステムは、1つ以上のプログラムを実行するために準備状態にある追加されたコンピューティングノードをプロビジョンする、または追加のコンピューティングノード上で1つ以上のプログラムの実行を自動的にさらに開始する等、少なくともいくつかの状況において、任意選択的に、他の動作をさらに行う場合がある。

【0012】

前述のように、ユーザは、コンピューティングノードの群のプログラム実行容量を動的に修正することに関する多様な種類のトリガを予め定義する場合があり、それらのトリガは、ユーザのコンピューティングノード群のための対応する自動化された動的プログラム実行容量修正を開始するために後で使用される場合がある。一例として、対応するコンピューティングノード群を所望のコンピューティングノード数または所望の総計コンピューティングリソース量に自動的に維持する、あるいは代わりに、指定された条件が満たされる場合に、指定されたコンピューティングノード数または総計コンピューティングリソース量に変更するために使用される場合がある等、コンピューティングノードの特定の所望数または1つ以上のコンピューティングリソースの所望の総計量を指定する、トリガが定義される場合がある。他の状況において、コンピューティングノード群の1つ以上の指示された性能特性が指定された閾値に到達する場合、またはそうでなければ指定された条件（例えば、指定された期間に指定された範囲内に留まる、指定された期間にわたって指定された量の変更、または特定の「加速」または変更率等、特定のトレンドを示す、プログラム実行容量使用の反復パターンを示す履歴データから等、1つ以上の指定されたパターンに一致またはその他対応する、AND、NOT、OR等の論理演算子を使用および/または他の様式で組み合わせられる場合がある等、複数の性能特性の値の指定された論理的または他の組み合わせを満たす等）を満たす場合にトリガされる場合がある等、コンピューティングノード数あるいは1つ以上のコンピューティングリソースの総計量の具体的な絶対的または相対的変更を指定するトリガが定義される場合がある。他の実施形態および状況において、予め定義されたトリガは、そうでなければ、コンピューティングノード群の性能特性の部分ではない情報に基づいて（例えば、特定のトリガのための条件の一部または全てである1つ以上の指定された時間に一致する現在の時間に基づいて、例えば、待ち行列に入っている、またはそうでなければ1つ以上のプログラムによって実施されることがわかっているまたは予測されている作業の量によって示される現在のコンピューティング負荷等、性能特性以外の様式で測定される現在のコンピューティングノード群上で実行されている1つ以上のプログラムのステータス情報に基づいて、例えば、他の実行中のプログラム上のコンピューティング負荷が増加または減少するにつれて、現在のコンピューティングノード群のプログラム実行容量を増加するように、現在のコンピューティングノード群が他の実行中プログラムと相互作用またはそうでなければサポートしている場合等、現在のコンピューティングノード群によって実行されていない1つ以上の他の実行中のプログラムの性能特性または他のステータス情報に基づいて等）満たされる場合がある。そのような性能特性は、単一のコンピューティングノードの1つ以上のコンピューティングリソースの使用の絶対的または相対的量（例えば、使用されている利用可能なメモリまたはCPUサイクル利用のパーセント量、使用されているネットワーク帯域幅の絶対量等）、群のコンピューティングノードの全ての1つ以上のコンピューティングリソースの総

10

20

30

40

50

計使用の絶対的または相対的、外部のコンピューティングシステムからの通信に応答する際のレーテンシまたは他の遅延の絶対的または相対的、1つ以上の所望の動作を実施する際のコンピューティングノードの障害の絶対的または相対的等の非排他的リストを含む、群のコンピューティングノードのうち1つ以上の測定可能な属性またはその操作に対応する他のメトリックに基づく場合がある。さらに、少なくともいくつかの実施形態および状況において、満たされる時に発生することになる特定のプログラム実行容量修正をトリガに直接指定させるのではなく、トリガが満たされると、指定されたシステムまたはモジュールが通知されるようにする場合があり、そのシステムまたはモジュールが、特定のプログラム実行容量修正をリクエストする場合がある（例えば、変動しない予め定義された容量修正、その時の現状に基づく等、通知時に動的に判定される容量修正等）。加えて、PESまたは他のシステムは、いくつかの実施形態において、群に関連するトリガの性能特性のいくつかまたは全てを判定するために、コンピューティングノードの群を監視するための多様な操作を実施する場合、またはそうでなければ、別の情報源から（例えば、コンピューティングノードを監視する第三者のソフトウェアから、そのコンピューティングノードを監視し、監視された情報を任意選択的に報告するように、コンピューティングノード上で実行しているソフトウェアから等）、そのような監視された性能特性を取得する場合がある。さらに、いくつかの実施形態において、PESまたは他のシステムは、トリガ条件が満たされることが指示された時に動的プログラム実行容量修正を開始するシステム定義トリガを有する場合、またはそうでなければ、指定された状況においてコンピューティングノード群にいくつかの種類の変更を行うことを自動的に判定する場合がある。

10

20

【0013】

定義されたトリガが、コンピューティングノード数または1つ以上のコンピューティングリソースの総計量における特定の絶対的または相対的変更を指定し、定義されたトリガの1つ以上の指定された条件が、コンピューティングノードの対応する群に対して、現在の状況に基づいて満たされると自動的に判定される時、PESまたは他のシステムは、トリガに指定されたプログラム実行容量修正を実施するかどうか、どのように実施するかを自動的に判定する場合がある。例えば、いくつかの種類指定されたプログラム実行容量修正は、即時に実施される場合があるが（例えば、1つ以上のコンピューティングノードの使用を終了するリクエスト、ユーザが即時実施に指定したトリガに基づくリクエスト等）、一方で、他の種類指定されたプログラム実行容量修正は、その期間中にリクエストまたはそうでなければ判定された複数のプログラム実行容量修正の総計の一部として考慮するために総計期間の最後まで延期される場合がある。同様に、ユーザによって（例えば、PESまたは他のシステムのGUIすなわちグラフィカルユーザインターフェースを介して、PESまたは他のシステムの定義されたAPIすなわちアプリケーションプログラムインターフェースを介してユーザのプログラムによって等）動的にリクエストされるプログラム実行容量修正は、動的プログラム実行容量修正の種類に基づいて、即時または延期実施に対するユーザからの明示的なリクエストに基づいて等、即時に実施される、および/または一時的に延期され、同様な様式で総計されると判定される場合がある。さらに、複数の判定されたプログラム実行容量修正の組み合わせをどのように管理するかを判定する時、いくつかの状況においては、異なる優先度が、異なる判定された修正に関連付けられる場合がある。その場合、動的に指定されたユーザリクエストまたは他のユーザ命令には満たされたトリガとは異なる優先権が与えられる（例えば、より高いまたはより低い優先度）、異なる種類の判定された修正には異なる優先度が与えられる（例えば、プログラム実行容量を減少するリクエストには、プログラム実行容量を増加するリクエストよりも高い優先権が与えられる）等、そのような優先度は、多様な様式で評価される場合がある。ユーザ定義トリガを使用し、性能特性を判定することに関する追加の詳細を以下に示す。

30

40

【0014】

さらに、少なくともいくつかの実施形態において、PESまたは他のシステムは、対応

50

する群のコンピューティングノードの複数の属性を追跡し、使用することに基づいて、またはそうでなければ、群の1つ以上のコンピューティングノードに各々関連する制御可能なパラメータに関して、複数の判定されたプログラム実行容量修正の定期的またはそうでなければ再帰的総計を管理する場合がある。一例として、コンピューティングノード群のプログラム実行容量に対応するメトリックとして、コンピューティングノード群のコンピューティングノード数を使用して、PESまたは他のシステムは、コンピューティングノード群に対して、コンピューティングノード群が開始された時に関連ユーザによって初期に設定される場合がある、トリガが満たされること、および/または動的に指定されたユーザリクエストによって修正される場合がある等、コンピューティングノード群の所望のコンピューティングノード数、群の現在利用可能なコンピューティングノードの実際のコンピューティングノード数（例えば、群のコンピューティングノードの継続的または反復監視によって判定される）、および群の現在利用可能なコンピューティングノードの公式記録のコンピューティングノード数（例えば、1つ以上の動的プログラム実行容量修正が開始される最後の時点に判定される、継続的または反復的な監視によって時々更新される際に任意選択的に判定される）等、少なくとも3つの相互関連測定値を維持し、使用する場合がある。コンピューティングノード群のそのような複数の属性は、実際の数を継続的または反復的に測定し、公式記録数をそれに従い更新する（例えば、群のコンピューティングノードの監視の実施に基づいて）、および現在の所望の数に一致するように、最新の公式記録数を更新するように定期的に試みる（例えば、複数の総計され判定された修正の組み合わせをどのように実施するか、実施するかどうかを考慮する時）等、多様な様式で使用される場合がある。

【0015】

前述のように、PESまたは他のシステムはさらに、行われた特定の動的プログラム実行容量修正に対する因果関係情報または他の責任を結論付けるために多様な操作を実施する場合がある。例えば、前述のように、コンピューティングノード群の動的プログラム実行容量修正に対するリクエストに対応するイベントは、受信される動的なユーザ指定の命令、および満たされることが自動的に判定される予め定義されたトリガ、ならびにいくつかの実施形態においては、PESまたは他のシステムによって自動的に行われる動作を含めて、追跡される場合がある（例えば、コンピューティングノードがフリーズ、またはそうでなければ所望の動作を実施するために利用不可能になっていること、その現在のコンピューティングノード群の一部としてコンピューティングノードの使用を自動的にシャットダウンまたはそうでなければ終了すると判定される場合等、実施される監視に基づく）。同様に、発生するイベントおよび/または他の容量変更に対応する変更等のコンピューティングノード群のプログラム実行容量における実際の変更は、同様に追跡される場合がある（例えば、障害が発生またはそうでなければ利用不可能になっているコンピューティングノードのインスタンス）。一例として、多様なイベントに関する情報は、第1のデータベーステーブルに記憶される場合があり、容量の変更または他の利用可能性の変更に関する多様な情報は、第2のデータベーステーブルに記憶される場合がある。特定のイベントと、特定の利用可能性の変更との間の関係が識別される場合（例えば、特定のイベントが、特定の利用可能性の変更を即時に実施させる）、そのイベントと利用可能性の変更との間の対応する結合は、同一の結合に関する識別子を、そのイベントおよびその利用可能性の変更に対する他の情報と共に記憶することによって、追跡される場合がある。複数のイベントが個別にまたは組み合わせられて、特定の利用可能性の変更を生じるが、因果関係が単一のイベントに結論付けられない他の状況において、単一の結合に関する識別子は、その利用可能性の変更およびそれらのイベントの各々に対する、ならびにそれらが同様にそれらの同じ複数のイベントに個別にまたは組み合わせられて結論付けられる場合がある場合はさらに1つ以上の他の利用可能性の変更に対する、他の情報と共に記憶される場合がある。このため、例えば、所与のコンピューティングノード群に対する複数のイベントが単一の総計期間中に発生し、1つ以上のプログラム実行容量の変更がその同じ総計期間中またはその直後に発生する場合（例えば、総計期間の最後に実施される調和アクティビティ

10

20

30

40

50

の一部として)、かつそれらのイベントのどれもが1つ以上の容量の変更のうちのどれにも直接帰属可能ではない場合、複数のイベントの全ての組み合わせは、1つ以上の容量の変更の各々に結論付けられる場合がある。そのような因果関係帰属に関する詳細は、図2Bに関する記載を含めて、以下に示す。

【0016】

加えて、同様な様式において、複数のそのような所望の属性、実際の属性、および公式の属性は、第1のセットの総計平均CPUサイクル利用の所望の量、実際の量、および公式の量、第2のセットの総計平均ネットワーク帯域幅利用の所望の量、実際の量、および公式の量等、コンピューティングノード群のプログラム実行容量に対応する1つ以上の他の制御可能なパラメータのために追跡され、使用される場合がある。さらに、コンピューティングノード群のプログラム実行容量に対応する複数のパラメータが同時に追跡され、使用される時、PESまたは他のシステムは、所望の総計平均CPUサイクル利用および所望の総計平均ネットワーク帯域幅利用を同時に達成するために、コンピューティングノード群を修正する等、パラメータの全てを管理しようとする場合がある。別の例として、コンピューティングノード群の複数のパラメータは、コンピューティングノードの数および群の多様なコンピューティングノードの指定された地理的場所(例えば、第1のデータセンタに群のコンピューティングノードの15~20パーセント、および1つ以上の他のデータセンタに残りの群のコンピューティングノード)の両方を含む場合があり、PESまたは他のシステムは、群のコンピューティングノード数およびコンピューティングノードの地理的場所の両方を同時に管理しようとする。1つ以上のプログラム実行容量パラメータを追跡し、管理するために複数の属性を使用することに関する追加の詳細を以下に示す。

【0017】

加えて、PESまたは他のシステムは、多様な実施形態において、多様な様式でコンピューティングノードへのアクセスをユーザに提供することができる。例えば、いくつかの実施形態において、プログラムを実行するためにPESから利用可能なコンピューティングノードのうち少なくともいくつかは、ユーザの各々が他のユーザよりも優先的アクセスを有してコンピューティングノードを使用するように、1人以上のユーザに対して、それらのユーザによる優先使用のために割り当てられる場合がある。1つのそのような実施形態において、ユーザの優先的アクセスは、リースに類似の様式等、指定された期間、それらのコンピューティングノードを使用するために専用または排他的アクセスを有するユーザの各々に基づく場合がある。加えて、いくつかの実施形態において、専用または他の優先的使用のために1人以上のユーザに割り当てられるコンピューティングノードのうち少なくともいくつかは、コンピューティングノードが割り当てられているユーザによって、コンピューティングノードが使用中でない時、および/またはコンピューティングノードが割り当てられているユーザが割り当てられたコンピューティングノードを他のユーザによって使用するために明示的に利用可能にする時等、場合によっては他のユーザのために余分のプログラム実行容量として使用される場合がある。この様式において、第1の群のユーザに割り当てられている少なくともいくつかのプログラム実行容量は、非保証ベース等で(例えば、1人以上の他のユーザによる優先または予約使用等、そのプログラム実行容量が他の目的のために望ましい場合は、余分のプログラム実行容量へのアクセスは無効になる場合がある)、時によっては、他のユーザの代わりに一時的にプログラムを実行するために利用可能になる場合がある。さらに、いくつかの実施形態において、PESは、リクエストされた実行を満たすに十分なコンピューティングノードが、そのようなリクエストが満たされるという保証はないが、リクエストされた時間に(または近辺で)利用可能であれば、そのようなリクエストによって指示された1つ以上のプログラムが実行される場合があるように、(例えば、それらのユーザのリクエスト時に即時に、指示された将来の時間に、指示された将来の期間中のある時間に等)プログラムを実行するという動的に受信されたユーザのリクエストを満たすために利用可能なオンデマンドのコンピューティングノードを含む場合がある。加えて、いくつかの実施形態において、即時(また

10

20

30

40

50

は予定の) 実行に対するそのようなオンデマンドリクエストが満たされ、それぞれのユーザの代わりに1つ以上のプログラムの実行を成功裏に開始した後、オンデマンドのコンピューティングノードの進行中の使用は、それぞれのユーザの選択する時間等、ある将来の時点まで継続することが保証され、任意選択的に所定の制限の対象となる(例えば、PESは、他の目的のための使用を先取りしないことは保証されるが、プログラムを実行しているコンピューティングノードの障害に対して保証されない等)。いくつかの実施形態において、オンデマンドの変動可能なプログラム実行容量を提供するために使用されるコンピューティングノードは、専用プログラム実行容量を提供するために使用されるコンピューティングノードとは、および/または余分のプログラム実行容量を提供するために使用されるコンピューティングノードとは区別される場合があり、このため、オンデマンドの変動可能なプログラム実行容量を提供するために使用されるコンピューティングノードのうちの一つが使用中でない場合、いくつかの実施形態において、それらは、オンデマンドの変動可能なプログラム実行容量リクエストが受信されるまで、余分のプログラム実行容量を提供するために使用される場合があり、一方で、他の実施形態において、それらは、余分のプログラム実行容量を提供するためには使用されない場合がある。他の実施形態において、単一の種類のプログラム実行容量だけが提供される場合があり、および/または他の種類のプログラム実行容量が提供される場合がある。

10

【0018】

図1Aは、複数のユーザのためにプログラムを実行するプログラム実行容量を提供する際に使用するために利用可能なコンピューティングノードを管理する、プログラム実行サービスの例を例示するネットワーク図である。図示の目的のために、特定の種類のプログラム実行容量が特定の様式で提供され、管理される、いくつかの例および実施形態を以下に記載する。加えて、後述する例および実施形態のうちの一つにおいては、コンピューティングノードの群によって提供されるプログラム実行容量は、特定の様式で測定される場合(例えば、コンピューティングノードの数に基づいて)、特定の様式で管理される場合(例えば、1つ以上のプログラム実行容量メトリックに関して、所望の属性、実際の属性、および公式の属性の使用を追跡することによって)、多様な様式で関連のユーザによって制御される場合(例えば、予め定義されたトリガおよび/または動的に指定される命令の使用に基づいて)、特定の様式で修正される場合(例えば、期間中に判定され、延期される少なくともいくつかのプログラム実行容量修正を総計し、次いで、群の中のコンピューティングノードの数を修正する等、期間の最後に1つ以上の対応する総計された修正を実施することによって)等がある。これらの例は、例示の目的で提供され、簡潔性のために簡素化されており、本発明の技法は、広範の多様な他の状況において使用される場合があり、以下にはそのうちの一つだけを記載することが理解される。

20

30

【0019】

図1Aの例において、多様なユーザ(図示せず)は、ネットワーク100上から、プログラム実行サービスプロバイダエンティティ105によって提供されるPESと相互作用するために、多様なクライアントコンピューティングシステム130を使用しており、PESの機能のうちの一つは、この例において、プログラム実行サービスシステムマネージャ(以下、「PES SM」)モジュール110の例示の実施形態によって提供され、他の機能は、この例において、プログラム実行サービス容量保守マネージャ(以下、「PES CMM」)モジュール115の例示の実施形態によって提供される。PES SMモジュール110は、例えば、群のために初期の所望のコンピューティングノード数を指定すること、およびコンピューティングノード数に対して後で自動的に動的修正を行う際に使用するためのトリガを指定することを含めて、ユーザのためにプログラムを実行するために使用されるコンピューティングノードの群を構成する際に特定のユーザを支援する場合がある。この例において、PESは、ユーザのプログラムを実行するために、多様なコンピューティングノード120を利用可能にするが、他の実施形態において、群のうち少なくともいくつかのために使用されるコンピューティングノードのうちの一つは、他の様式で提供される場合がある(例えば、ユーザ、および/または外部コンピ

40

50

ューティングシステム 140 等の第三者によって利用可能にされるが、PES によって管理される)。加えて、PES CMM モジュール 115 は、使用中のコンピューティングノードを継続的にまたは反復して監視すること、および任意選択的に、プログラム実行容量を以前に判定されたレベルで維持するように、障害を発生またはそうでなければ利用不可能になる任意のコンピューティングノードを置換するようにすることを支援する場合がある。

【0020】

ネットワーク 100 は、例えば、連結されたネットワークのうちの公的にアクセス可能なネットワークの場合があり、可能性として、インターネット等の多様な別個の当事者によって運用される場合がある。他の実施形態において、ネットワーク 100 は、例えば、非特権ユーザには全体的または部分的にアクセス不可能な企業または大学のネットワーク等、非公開ネットワークの場合がある。さらに他の実施形態において、ネットワーク 100 は、インターネットへのアクセス、および/またはインターネットからのアクセスを有する 1 つ以上の非公開ネットワークを含む場合がある。図示された実施形態において、PES SM モジュール 110 および PES CMM モジュール 115 は各々、1 つ以上のコンピューティングシステム（図示せず）上で実行するソフトウェア命令を含む場合がある。加えて、モジュール 110 および 115 ならびに多様なコンピューティングノード 120 は、単一のデータセンタで、またはそうでなければ、共存するコンピューティングシステムを使用する等、あるいはその代わりに、多様な別個の地理的場所において多様なコンピューティングシステムを使用する分散様式等、多様な様式で提供される場合がある。

【0021】

いくつかの実施形態において、例示されたコンピューティングノード 120 は、複数の物理的コンピューティングシステムおよび/または 1 つ以上の物理的コンピューティングシステム上でホストされる複数の仮想マシン（図 1B を参照して詳細に記載する）を含む場合がある。コンピューティングノード 120 の各々は、例えば、処理容量（例えば、処理ユニットの数および/またはサイズ）、メモリ容量、記憶容量、ネットワーク帯域幅容量等のうちの 1 つ以上の組み合わせによって測定される場合があるような、特定量のプログラム実行容量を提供する等、1 つ以上のプログラムを実行するために利用可能な何らかの量のコンピューティングリソースを有する。いくつかの実施形態において、PES プロバイダ 105 は、予め構成されたコンピューティングノードを提供する場合があり、各々の予め構成されたコンピューティングノードは、ユーザの代わりにプログラムを実行するために利用可能なリソースの量と同等、そうでなければ同様な量を有し、一方で、他の実施形態において、PES プロバイダ 105 は、各選択肢が変動する量および/または種類（例えば、処理ユニットのサイズ、速度および/または種類、処理ユニットの数、メモリおよび/または記憶の量、32 ビットまたは 64 ビット等のプラットフォームの構成等）のコンピューティングリソースを有する等、ユーザの代わりにプログラムを実行するためにユーザが選択する場合がある多様な異なるコンピューティングノードの選択肢を提供する場合がある。

【0022】

この例示された実施形態において、プログラム実行サービスは、複数のユーザの各々のために、1 つ以上のコンピューティングノード 120 の群を管理するための機能を提供する。他の部分に詳細を記載するように、多様なユーザは、ユーザの代わりにプログラムを実行するためのコンピューティングノードの群の使用を開始するというリクエストを指定するために、PES SM モジュール 110 と相互作用する場合がある。多様な実施形態において、そのようなリソースは、ユーザの代わりにコンピューティングノードの群上で 1 つ以上のプログラムを実行するリクエストの時、および/またはユーザが PES のサービスを使用するために最初に登録および/または加入する時等、1 つ以上の他の時間に指定される場合がある。いくつかの実施形態において、PES SM モジュール 110 は、ユーザがユーザの代わりに実行する 1 つ以上のプログラムに関する情報（例えば、プログラム、ソースコード、1 つ以上のプログラムのアドレス指定可能な場所等）、アカウント情報（

10

20

30

40

50

例えば、ユーザ名、請求情報等)、使用条件等を指定することができるように、1人以上のユーザに加入および/または登録サービスを提供する場合がある。いくつかの実施形態において、ユーザがPESSMモジュール110と相互作用してサービスに加入および/または登録した後、ユーザには、ユーザに関連する1つ以上の識別子(例えば、キー、トークン、ユーザ名等)が発行される場合があり、ユーザの代わりにプログラムを実行する際に関連して使用される。他の実施形態において、PESSMモジュール110以外のモジュール(図示せず)は、PESSの加入および/または登録サービスに関する多様な操作を実施するために提供される場合がある。

【0023】

1つ以上のコンピューティングノードを使用するためにユーザからリクエストが受信された後、PESSM1モジュール10は、リクエストを満たすために利用可能なリソースを備える十分な数のコンピューティングノード120が存在するかどうかを判定する場合があり、存在する場合、PESSM1モジュール10は、ユーザの代わりに適当な量のコンピューティングノード上でリクエストに対する1つ以上のプログラムの実行を開始する場合がある。ユーザが1つ以上のコンピューティングノードの群上で1つ以上のプログラムをさらに実行するリクエストを予定している場合、PESSMモジュール110は、1つ以上の将来の時間に1つ以上のプログラムを実行するための適切な量のコンピューティングノードを即時に予約しようとする場合、および/または後の時間まで(例えば、1つ以上の将来の時間が発生する時)、実行のためにどのコンピューティングノードを使用するかの判定を遅延する場合がある。例示される実施形態において、PESSMモジュール110が、ユーザリクエストのためにコンピューティングノードを割り当てることが不可能な場合、リクエストは、プログラムが実行されない等、失敗する場合がある。そのような場合、ユーザは、後で実行するために、失敗したリクエストを再送信する場合がある。前述のように、いくつかの実施形態において、ユーザは、使用されるコンピューティングノードの数、使用されるコンピューティングノードの種類、コンピューティングノードが使用される期間、コンピューティングノードが実施する特定の操作(例えば、データ転送および/または記憶)等に基づいて等、PESSの使用に関連して多様な料金が課金される場合がある。

【0024】

1つ以上のコンピューティングノードの群がユーザの代わりに1つ以上のプログラムを実行する際に使用するために提供された後、コンピューティングノード群は、多様な様式で管理される場合がある。例えば、前述のように、PESSCMMモジュール115は、群のコンピューティングノードによって提供されている実際のコンピューティングノード数または実際のプログラム実行容量の他の測定値を含めて、コンピューティングノードのいくつかまたは全ての性能特性を判定するため等、群のコンピューティングノードを監視する場合がある。群の実際のプログラム実行容量は、例えば、コンピューティングノードの1つ以上が障害を発生またはそうでなければ利用不可能になると変化する場合があり、少なくともいくつかの実施形態において、モジュール115は、そのような状況においてコンピューティングノード群のプログラム実行容量を維持するために多様な操作を実施する場合がある(例えば、利用不可能なコンピューティングノードの代わりに、群に置換コンピューティングノードの追加を開始することによって)。加えて、モジュール115は、実際のコンピューティングノード数における変更を検出時、定期的等のように、コンピューティングノード群の公式記録のコンピューティングノード数を更新するために、コンピューティングノード群の判定された実際のコンピューティングノード数についての情報を使用する場合がある。

【0025】

加えて、PESSMモジュール110はまた、多様な様式でコンピューティングノード群を管理する際に支援する場合もある。例えば、前述のように、群に関連するユーザは、多様な指定された条件が満たされた場合に、群に対して行われるコンピューティングノード数の変更の種類を指定する、1つ以上の数修正トリガを以前に指定していた場合がある

10

20

30

40

50

。その上、少なくともいくつかの実施形態において、群に関連するユーザは、群のコンピューティングノードの数を変更することを含めて、多様な時点で関連するコンピューティングノード群の操作に対する変更を動的に指定する場合がある。コンピューティングノード群を管理することの一部として、モジュール110は、コンピューティングノード群の所望のコンピューティングノード数を追跡し、コンピューティングノード群の公式記録のコンピューティングノード数と、コンピューティングノード群の所望のコンピューティングノード数に定期的を調和する場合がある。そのような定期的調和は、予め定義されたユーザ指定トリガのうちの1つ以上を満たすことから、および/または1つ以上の動的に指定されたユーザ命令から等、総計期間中に発生するコンピューティングノード数へのリクエストされた修正を追跡し、総計することを含む場合がある。期間の最後に、モジュール110は、次いで、総計されたコンピューティングノード数修正を実施するかどうか、どのように実施するかを判定する場合があり、コンピューティングノード数修正の総計に基づいて、所望のコンピューティングノード数を更新する場合がある。モジュール110は次いで、総計されたコンピューティングノード数の修正を開始し、総計されたコンピューティングノード数の修正からもたらされる新しい所望のコンピューティングノード数に一致するように公式記録のコンピューティングノード数を更新することによって、所望コンピューティングノード数および公式記録のコンピューティングノード数の調和を完了する場合がある。さらに、期間の最後に現在の実際のコンピューティングノード数が公式記録のコンピューティングノード数と異なる場合、実際の実際のコンピューティングノード数を考慮して、現在の実際のコンピューティングノード数を現在所望のコンピューティングノードに更新するように総計されたコンピューティングノード数の修正の開始が実施される等、所望のコンピューティングノードおよび公式記録のコンピューティングノードの調和がさらに実施される場合がある。モジュール110は、指定の期間が経過する都度、最低量のコンピューティングノード数の変更がリクエストされる都度等、反復してそのような定期的調和アクティビティをさらに実施する場合がある。

【0026】

さらに、PESSMモジュール110は、特定のコンピューティングノード群に関して発生するイベントおよび特定のプログラム実行容量変更を追跡するために他のアクティビティを実施する場合がある。プログラム実行容量変更情報のうちの少なくともいくつかは、定期的または再帰的調和アクティビティの一部として、モジュール110によって開始される動的なプログラム実行容量修正に対応する場合があり、任意選択的にいくつかの状況において、PESMMモジュール115によって実施される監視からのプログラム実行容量変更情報（例えば、障害を発生またはそうでなければ利用不可能になるコンピューティングノード、またはコンピューティングノード群のプログラム実行容量における他の種類の検出された変更に対応する）をさらに含む場合がある。モジュール110は、実施される定期的または他の再帰調和アクティビティに一致して、または代わりに他の時間に等、特定の容量変更に対する因果関係または他の責任を特定のイベントに結論付けるために、多様な時点で自動判定をさらに行う場合がある。

【0027】

図1Aの前述の例示的な実施形態は、多様なユーザのために多様な種類の機能を提供するPESに関して記載されるが、多様な他の実施形態が存在する場合があることが理解される。例えば、少なくともいくつかの実施形態において、コンピューティングノード120のうちの単一の未使用部分（例えば、未使用の処理ユニットクロックサイクル、メモリの未使用部分等）は、第1のユーザの1つ以上のプログラムが、1人以上の他のユーザのプログラムと単一のコンピューティングノードのリソースを共有している場合がある等、1人以上のユーザによって使用するために利用可能になる場合がある。加えて、いくつかの実施形態は、プログラム実行サービスおよびプログラム実行容量に関して記載されるが、記載の技法は、多様な他の群のコンピューティングノードまたは他の種類のコンピューティング関連リソースに対するアクセスを管理するために使用される場合があることが理解される。複数のユーザによって使用されるために管理される場合がある他の種類のコン

10

20

30

40

50

ピューティング関連リソースの例の非排他的リストとして、持続的データ記憶機能（例えば、ハードディスクドライブ等の不揮発性記憶装置）、一時的データ記憶機能（例えば、RAM等の揮発性メモリ）、メッセージ待ち行列および/または渡し機能、他の種類の通信機能（例えば、ネットワークソケット、仮想通信回路等）、データベース管理機能、専用帯域幅または他のネットワーク関連リソース、入力装置機能、出力装置機能、CPUサイクルまたは他の命令実行機能等が挙げられる場合がある。

【0028】

図1Bは、複数の物理的コンピューティングシステムを含む1つ以上のデータセンタを使用するプログラム実行サービスが提供される場合がある実施形態を例示する。具体的には、図1Bは、プログラム実行サービスのPESSMモジュール180およびPESCMモジュール160が、1つ以上のデータセンタにある多様なコンピューティングシステムを使用して、ユーザの代わりに1つ以上のプログラムの実行を管理する、例示的な実施形態を例示するネットワーク図である。例示された例は、PESによって使用されるデータセンタ170を含み、これは、データセンタ170に対して外部のインターネット196に接続される。この例において、インターネット196は、非公開ネットワーク194を介するコンピューティングシステム190、および直接アクセス可能なコンピューティングシステム192等、多様な外部のコンピューティングシステムへのアクセスを提供する。非公開ネットワーク194は、例えば、非公開ネットワーク194に対して外部の非特権コンピューティングシステムから全面的または部分的にアクセス不可能な企業ネットワークの場合がある。コンピューティングシステム192は、例えば、（例えば、電話またはケーブルモデム、デジタル加入者回線（以下「DSL」）等を介して）インターネットに直接接続する家庭用コンピューティングシステムを含む場合がある。加えて、少なくともいくつかの実施形態においてPESによってさらに使用される場合があるように、インターネット196を介してデータセンタ170に接続される、1つ以上の他のデータセンタ198が例示される。

【0029】

例示的なデータセンタ170は、いくつかの物理的なホストコンピューティングシステム175、物理的なコンピューティングシステム182、PESのPESSMモジュール180、およびPESのPESCMモジュール160を含む。この例において、ホストコンピューティングシステム175は各々、複数の仮想マシン177aおよびVMマネージャ構成要素179aを備えるホストコンピューティングシステム175aに関して例示されるように、複数の仮想マシンを提供し、それらの仮想マシン（例えば、ハイパーバイザまたは他の仮想マシンモニタ）を管理するための仮想マシン（以下「VM」）マネージャ構成要素を有する。他のホストコンピューティングシステム175bは、そのような構成要素を同様に含む場合があるが、それらの他の構成要素は簡潔性のために例示されず、コンピューティングシステム182のいくつかまたは全ては、任意選択的に、1つ以上のそのような仮想マシンおよび/またはVMマネージャ構成要素（図示せず）を同様に有する場合がある。ホストコンピューティングシステムによって提供される仮想マシンの各々は、ホストコンピューティングシステム上の第1の仮想マシンコンピューティングノードを第1のユーザのための第1のコンピューティングノード群の一部にし、その同じホストコンピューティングシステム上の第2の仮想マシンコンピューティングノードを第2のユーザのための第2のコンピューティングノード群の一部にする等、PESのための別個のコンピューティングノードとして使用される場合がある。代替として、他の実施形態において、データセンタにある物理的ホストコンピューティングシステムのいくつかまたは全ては、そうではなく、PESのエンドユーザ顧客の代わりに1つ以上のプログラムを実行するコンピューティングノードとして直接機能する等、仮想マシンを一切提供しない場合がある。さらに、いくつかの実施形態において、多様なコンピューティングシステム175および182は、異なる機能を有する場合、異なる使用関連料金を有する場合、異なる種類のユーザプログラム（例えば、異なるサイズの仮想マシンソフトウェアの画像インスタンス、または異なるパターンのI/Oおよびメモリのアクセスならびにネットワーク使

10

20

30

40

50

用量等の、異なる種類のリソース条件および/またはコンピューティングリソース使用量を有するプログラム)をサポートする場合等がある。そうである場合、特定のユーザおよび/またはそれらのプログラムは、1つ以上のそのような因子に従って(例えば自動的に)グループ化される場合があり、特定のプログラムを実行するためにどのコンピューティングシステムを選択するかに関して、制約条件および/または優先条件としてさらに使用される場合がある。例示的データセンタ170はさらに、スイッチ、エッジルータ、コアルータ等の複数のネットワークングデバイス(図示せず)を含む場合がある内部ネットワーク172を含み、コンピューティングシステム175および182、PES CMMモジュール160、ならびにPES SMモジュール180は、内部ネットワーク172に接続される。多様なホストコンピューティングシステム175および他のコンピューティングシステム182は、共通のバックプレーンまたは他の相互接続媒体を共有するラックの中にグループ化されることを含む、多様な様式で配置される場合がある。さらに、モジュール160および180の各々は、1つ以上のコンピューティングシステム(図示せず)を使用して実行される場合がある。

10

【0030】

例示されるPES SMモジュール180およびPES CMMモジュール160は、他の部分に詳細を記載するように、コンピューティングシステム175および182を使用して提供されるコンピューティングノードの群上でのプログラムの実行を管理するために、記載の技法の少なくともいくつかを実施する。特定のコンピューティングノードがユーザの1つ以上のプログラムを実行するために選択される時、PES SMモジュールは、いくつかの実施形態において、その選択されたコンピューティングノードに対してプログラムの実行を制御する、VMマネージャ構成要素または他のマネージャ構成要素と相互作用することによって、それらのプログラムの実行を開始する場合、または代替として、選択されたコンピューティングノード上でプログラムを直接実行する場合がある。PESのユーザは、PES SMモジュール180と相互作用するために、コンピューティングシステム190または192、あるいは他のデータセンタ198のうちの1つにあるコンピューティングシステム等、多様なコンピューティングシステムを使用する場合がある。

20

【0031】

図1Bのデータセンタは、例示の目的のみに提供され、プログラム実行サービスまたは他のソフトウェア実行サービスは、他の実施形態において他の様式で提供される場合があることが理解される。例えば、PES SMモジュール180および/またはPES CMMモジュール160は、その代わりに、コンピューティングシステム190、192またはデータセンタ198で等、データセンタ170に対して外部にある1つ以上の他のコンピューティングシステムを使用して提供される場合がある。

30

【0032】

図2Aは、PES SMモジュールおよび/またはPES CMMモジュールの実施形態によって自動的に実施される場合がある技法等、例示的なユーザの1つ以上のプログラムを実行するために提供される、コンピューティングノードの例示的群を管理するための技法の例を例示する。具体的には、この例において、特定のユーザ(以下ユーザUUUと称す、図2Aには図示せず)は、実行中プログラムによって提供されるサービスによって受信されたリクエストを処理するために、代替のコンピューティングノードとして機能する等(例えば、群の複数のコンピューティングノード全体でサービスのコンピューティング負荷をバランスするために)、ユーザの代わりに指示されたプログラムのコピーを各々実行する複数のコンピューティングノードの群の使用を開始していた。時系列グラフ210に例示されるように、ユーザは、時間T1に、コンピューティングノード群が提供されることをリクエストしており、群の初期の所望のコンピューティングノード数215aを8つのコンピューティングノードに指定している。情報250は、初期のリクエスト時等、ユーザがコンピューティングノード群に指定していた多様なユーザ定義のトリガを示す。加えて、時系列グラフ205は、コンピューティングノード群のために追跡され、トリガ250が満たされるかどうかを判定するために使用される2つの例示的な種類の性能特性に

40

50

ついでの情報为例示し、これは、この例においては、コンピューティングノード群の総計平均CPU利用205aおよびコンピューティングノード群の総計平均ネットワーク帯域幅利用205bを含む。

【0033】

コンピューティングノード群の提供を開始するという受信されたユーザリクエストに対応する、時間T1でのイベントE1に応じて、例示的PES（図示せず）は、初期の所望のコンピューティングノード数に従って、初期に8つのコンピューティングノードを含む、ユーザのためのコンピューティングノード群を開始する。加えて、コンピューティングノード群の公式記録のコンピューティングノード数225は、同様に、8つのコンピューティングノードに設定される。しかしながら、時系列グラフ210に示されるように、コンピューティングノードの各々上でPESまたはユーザによってプログラムコピーを実行させることを含め、コンピューティングノード群の一部として使用するためにコンピューティングノードをプロビジョンし、利用可能にするためには多少の時間がかかるので、8つのコンピューティングノードは実際には即時に利用可能ではない。具体的には、時間T1後に初期時間が経過した後、コンピューティングノード群のプログラム実行容量における変更C1255が発生し、これは、時間T1で開始させられた後、およそ時間T2で完了し、8つのコンピューティングノードのうちの最初の4つが利用可能になることに対応する。したがって、コンピューティングノード群に追跡される実際のコンピューティングノード数220は、その時に0から4に増加する。加えて、同時にまたはその直後に、時系列グラフ205は、総計平均CPU利用205aおよび総計平均ネットワーク帯域幅利用205bの性能特性は、群の利用可能なコンピューティングノードの操作に基づいて追跡されることが開始することを示す。

【0034】

この例において、PES（または他の第三者システム）は、性能特性205aおよび205bならびに実際のコンピューティングノード数220情報が最新様式に維持されるように、実質的に継続様式で群のコンピューティングノードの監視を実施する。しかしながら、この例において、コンピューティングノード群のコンピューティングノード数に対する少なくともいくつかの種類の動的修正は、総計期間260中に少なくともいくつかの種類のリクエストされた修正を総計し、それぞれ所望のコンピューティングノード数、実際のコンピューティングノード数、および公式記録のコンピューティングノード数215、220および225間で、総計期間の最後に調和アクティビティを実施する等、定期的のみ実施される。したがって、第1の総計期間260a中、コンピューティングノード群のコンピューティングノード数に追加の変更255が発生する。例えば、最初の4つのコンピューティングノードを利用可能にする変更C1の後、およそ時間T3に完了される次の変更C2は、コンピューティングノード群の一部として使用するために利用可能になる3つの追加のコンピューティングノードに対応する。

【0035】

しかしながら、この例において、初期にリクエストされた第8のコンピューティングノードは、初期に選択されたコンピューティングノードに伴うハードウェア障害に起因して等、適時に利用可能にならない（または、代わりに他の状況において、コンピューティングノード群で使用されるために時間T1では7つのコンピューティングノードだけが利用可能であることに起因して、第8のコンピューティングノードは、後で利用可能になると直ちに動的に追加される）。したがって、イベントE2265は、およそ時間T4で、変更C2の後直ちに発生することが示され、PESは、必要に応じて（例えば、中間状態においてフリーズした場合）本来の第8のコンピューティングノードを終了し、置換の第8のコンピューティングノードの提供を開始する。図2Bに関して詳細を記載するように、この例において、PESは、初期の第8のコンピューティングノードを追加することに対応する、変更C10aを時間T1に開始しているが、その変更C10aは、追加され、利用可能になる実際の第8のコンピューティングノードの完了に失敗し、そのために、変更C10aは、時系列グラフ210の実際のコンピューティングノード数220に反映さ

10

20

30

40

50

れず、図 2 A にも示されない。そうではなく、この例において、イベント E 2 は、同時またはほぼ同時に発生する、2つの別個のイベント E 2 a および E 2 b (それぞれ図示せず) に分割される。具体的には、P E S は、この例において、(例えば、指定期限内に) 正しく開始することに失敗している第 8 のコンピューティングノードに対応する、イベント E 2 a を時間 T 4 に記録し、P E S は、正しく開始することに失敗した初期の第 8 のコンピューティングノードを終了するように、時間 T 4 で変更 C 1 0 b (図示せず) を自動的に開始する。さらに、この例において、イベント E 2 a の結果として発生する、初期の第 8 のコンピューティングノードの実際の終了は、それ自体、置換の第 8 のコンピューティングノードの即時提供を自動的に開始する、別のイベント E 2 b として時間 T 4 で処理される。そのような一連のイベントおよび対応する変更の結果は、第 1 のイベントの変更からの結果がそれ自体、別の変更を生じさせる第 2 のイベントとして処理される場合があるように、他の部分に詳細を記載するように、イベントと対応する変更結果との間の相互関係を追跡する際に多様な利益を提供する場合がある。その後、イベント E 2 b に基づいて時間 T 4 に開始し、時間 T 5 に完了する、変更 C 3 2 5 5 は、置換の第 8 のコンピューティングノードがコンピューティングノード群の一部として使用するために利用可能になるので、実際のコンピューティングノード数を 7 から 8 にもたらず。

【 0 0 3 6 】

コンピューティングノード群の第 8 のコンピューティングノードが総計期間 2 6 0 a の部分で意図した様式で操作した後、コンピューティングノードのうちの一つが失敗している、あるいは利用不可能になっていることによって促されて、実際のコンピューティングノード数 2 2 0 における別の変更 C 4 2 5 5 は、およそ時間 T 7 で発生する。対応するイベント E 3 2 6 5 は、変更 C 4 とほぼ同時に発生し、その中で、P E S は、必要に応じて利用不可能なコンピューティングノードを終了し、任意選択的に、置換コンピューティングノードの提供を自動的に開始する。具体的には、イベント E 2 に関して上述した様式と同様な様式で、イベント E 3 は、この例において、同時またはほぼ同時に発生する、イベント E 3 a および E 3 b (独立して図示せず) に分割される。このように、P E S は、この例において、利用不可能になったことが検出されているコンピューティングノードに対応するイベント E 3 a を時間 T 7 に記録し、P E S は、変更 C 4 を直接発生させる等、利用不可能なコンピューティングノードを終了するためのアクティビティを時間 T 7 で自動的に開始する。さらに、この例において、イベント E 3 a の結果として発生する、利用不可能なコンピューティングノードの終了は、それ自体、およそ時間 T 7 で、置換のコンピューティングノードの即時提供を自動的に開始する別個のイベント E 3 b として処理されるが、他の実施形態および状況において、置換コンピューティングノードの任意のそのような提供は、そうではなく、総計期間 2 6 0 a の最後に実施される調和アクティビティ H 1 まで延期される。その後、この例において、イベント E 3 b によって直接発生させられる、実際のコンピューティングノード数を 8 に戻す、変更 C 5 2 5 5 は、置換コンピューティングノードがコンピューティングノード群の一部として使用するために利用可能になると、およそ時間 T 9 で完了される。この例において、公式記録のコンピューティングノード数 2 2 5 は、変更 C 4 および C 5 にも、以前の変更 C 1 ~ 3 にも対応するように一時的に更新されないが、他の実施形態において、公式記録のコンピューティングノード数 2 2 5 は、更新された実際のコンピューティングノード数 2 2 0 と一致する状態に公式記録のコンピューティングノード数 2 2 5 を継続または反復して維持するために、いくつかまたは全てのそのような変更を反映するように更新される場合がある。指定されたトリガ 2 5 0 は全く満たされない、および動的に指定されたユーザ指示は全く受信されないことを含めて、他のコンピューティングノード容量の利用可能性の変更またはイベントは、この例において、総計期間 2 6 0 a の残りの間は全く発生しない。したがって、期間 2 6 0 a の最後の時間 T 1 1 で、調和アクティビティの第 1 のセット H 1 が考慮されるが、それぞれ現在所望のコンピューティングノード、実際のコンピューティングノード、および公式記録のコンピューティングノード 2 1 5、2 2 0、および 2 2 5 は、全て 8 つのコンピューティングノードに一致するため、アクティビティは全く必要とされない。利用不

10

20

30

40

50

可能なコンピューティングノードを置換する変更C5が、この例に例示されるように総計期間260a中に実施されなかった場合、そうではなく、(例えば、総計期間260a中にリクエストまたは開始された任意の他の動的な利用可能性の変更に伴って)その時間に利用不可能なコンピューティングノードを置換するために、総計期間後に調和アクティビティH1の一部として開始される。

【0037】

しかしながら、第2の総計期間260b中、追加のイベントは確かに発生する。具体的には、イベントE5 265は、時系列グラフ205に平均総計CPU利用205aの線上の最初の黒い楕円で図に例示されるように、トリガTR-1 250aが増加する総計平均CPU利用205aによって満たされたというPESによる自動決定に対応して、およそ時間T16で最初に発生する。トリガTR-1が満たされると、12の合計のリクエストされた所望のコンピューティングノード数215cに対して、所望のコンピューティングノード数215を4つのコンピューティングノードだけ増加するというリクエストを開始する(または代替として、現在所望のコンピューティングノード数に関係なく、更新された所望のコンピューティングノード数が12に設定されるようにリクエストすることができる)。同様に、さらに追加の時間が経過した後、イベントE6 265は、次に、時系列グラフ205に総計平均ネットワーク帯域幅利用205b線上の黒い楕円で図に例示されるように、トリガTR-N 250cが増加総計平均ネットワーク帯域幅利用205bによって満たされたという判定に対応して、およそ時間T18に発生する。トリガTR-Nが満たされると、合計10のリクエストされた所望のコンピューティングノード数215dに対して、所望のコンピューティングノード数215を2つのコンピューティングノードだけ増加するというリクエストを開始する(または代替として、現在所望のコンピューティングノード数に関係なく、更新された所望のコンピューティングノード数が10に設定されるようにリクエストすることができる)。短い追加時間が経過した後、イベントE4 265は、およそ時間T19で発生し、その中で、コンピューティングノード群に関連するユーザは、合計で11のリクエストされた所望のコンピューティングノード数215bに対して、所望のコンピューティングノード数215を3つのコンピューティングノードだけ増加するように動的に指定されたリクエストを提供する(または、代替として、現在所望のコンピューティングノード数に関係なく、更新された所望のコンピューティングノード数は11に設定されるようにリクエストすることができる)。このリクエストは、例えば、総計平均CPU利用205aが高い、コンピューティングノード群上の合計のコンピューティング負荷が増加している等というユーザ通知に基づいて行われる場合がある。

【0038】

最後に、総計期間260bの終了直前に、障害が発生している、またはそうでなければ利用不可能になっている群のコンピューティングノードのうちの1つによって促され、追加の変更C6は、およそ時間T20に発生する。対応するイベントE9は、変更C6とほぼ同時に発生し、その中で、PESは、必要に応じて利用不可能なコンピューティングノードを終了する。この例において、しかしながら、置換コンピューティングノードの提供を即時に開始するのではなく、コンピューティングノード群の中のコンピューティングノードの数は、それらの調和アクティビティに基づいて削減され、置換コンピューティングノードは必要がなくなるため、PESは、近い将来の第2のセットH2の調和アクティビティが時間T21で開始されるまで待機することを判定する(例えば、時間T21までの残りのわずかな量の時間に基づいて、置換コンピューティングノードを提供することのそのような延期は、使用中に利用不可能になる任意のコンピューティングノードのデフォルト動作であることに基づいて等)。他の実施形態において、PESは、そうではなく、置換コンピューティングノードの提供を即時に開始する場合がある(例えば、任意の状況において利用不可能なコンピューティングノードの置換を延期しないことに基づいて、所望のコンピューティングノード数が、調和アクティビティH2の一部として、減少するのではなく増加する可能性が高いことに起因して、この状況において利用不可能なコンピュー

10

20

30

40

50

ティングノードの置換を延期しないことに基づいて、ユーザが後の時間T3まで8つのコンピューティングノードに対して既に支払を済ませていること等の他の因子に起因して、この状況において利用不可能なコンピューティングノードの置換を延期しないことに基づいて等)。この例において、イベントE2およびE3に関して上述の様式に同様な様式で、イベントE9は、同時またはほぼ同時に発生する、2つの別個のイベントE9aおよびE9b(別々に図示せず)に分割される。このように、PESは、この例において、利用不可能になったことが検出されているコンピューティングノードに対応するイベントE9aを時間T20で記録し、PESは、変更C6を直接発生させる等、利用不可能なコンピューティングノードを終了するためのアクティビティを時間T20で自動的に開始する。さらに、この例において、イベントE9aの結果として発生する、利用不可能なコンピューティングノードの終了はそれ自体、所望のコンピューティングノード数215を8つのコンピューティングノードの所望の数215aに維持するように等、置換コンピューティングノードとして使用する群の追加のコンピューティングノードのリクエストを開始するための別のイベントE9bとしておよそ時間T20で処理される。この例において、イベントE9bのリクエストは、イベントE4~E6と同様な様式で処理され、現在の総計期間260bの最後で実施される調和アクティビティH2まで延期されるが、他の実施形態および状況においては、置換コンピューティングノードのそのような提供は、そうではなく、即時に開始される場合がある。加えて、この例において、公式記録のコンピューティングノード数225は、変更C4およびC5に関して前述した様式に同様な様式で、変更C6に対応するようには更新されないが、他の実施形態において、公式記録のコンピューティングノード数225は、そのような変更を反映するように更新される場合がある。

【0039】

したがって、第2の総計期間260bの最後に、調和アクティビティH2が開始され、この場合、コンピューティングノード群への動的修正となる。具体的には、この例のPESは、イベントE4、E5、E6およびE9bに対する多様なリクエストされた所望のコンピューティングノード数215b、215c、215dおよび215aに対応するリクエストされた修正を総計し、先行の所望の8つのコンピューティングノード数215aに動的総計数修正を行うと判定し、この例の動的総計数修正は、更新された現在所望の12のコンピューティングノード数215eに対応するように、4つの追加のコンピューティングノードの増加(リクエストされた数修正の最大をとることに基づいて)である。他の実施形態において、動的総計数修正は、イベントE9bに従い先行の所望のコンピューティングノード数215aを維持するように動的総計数修正を選択するように(例えば、現在の実際のコンピューティングノード数から1つのコンピューティングノードの増加となる)、イベントE4~E6のリクエストされた数修正の最小をとるように(例えば、現在所望のコンピューティングノード数から2つのコンピューティングノードの増加となる)、イベントE4~E6のリクエストされた数修正の平均をとるように(例えば、現在所望のコンピューティングノードから3つのコンピューティングノードの増加となる)、イベントE4~E6のリクエストされた数修正の集積をとる(例えば、現在所望のコンピューティングノード数から9つのコンピューティングノードの増加となる)、イベントE4~E6のリクエストされた数修正の最高優先度をとる(例えば、例えば、E4のユーザ命令がイベントE5およびE6のトリガを満たすことよりも高い優先度であるとみなされる場合、現在所望のコンピューティングノード数から3つのコンピューティングノードの増加となる)イベントE4~E6のリクエストされた数修正の第1のリクエストまたは最後のリクエストをとる(例えば、現在所望のコンピューティングノード数から、それぞれ、4つまたは3つのコンピューティングノードの増加となる)ように等、他の様式で判定される場合がある。ユーザ命令イベントがトリガを満たすイベントよりも高い優先度を持つ(例えば、ユーザ命令は、トリガを満たすイベントからのどのリクエストされた動的修正も必ずオーバーライドする場合)いくつかの実施形態および状況において、PESはさらに、この例において、ユーザ命令がイベントE4のために受信された後、トリガを満たすイベントをすべて無視する(または決して判定しない)ように、ユーザ命令イベントが受信

10

20

30

40

50

された後、総計期間中に追加のトリガを満たすイベントが発生することを阻止する場合がある。さらに、この例において、現在の実際のコンピューティングノード数および公式記録のコンピューティングノード数220および225は、調和アクティビティH2の時点では、それぞれ7つおよび8つのコンピューティングノードと異なる。したがって、調和アクティビティH2の一部として、PESは、7つの現在の実際のコンピューティングノード数から、12の新しい更新された所望のコンピューティングノード数に上昇するように、コンピューティングノード群のために5つの追加のコンピューティングノードを提供することを開始し、さらに、12の新しい更新された所望のコンピューティングノード数に一致するように、公式記録のコンピューティングノード数225を更新する。このため、この例において、5つのコンピューティングノードの容量の利用可能性の変更は、イベントE9bに従い、変更C7に関して利用不可能になった1つのコンピューティングノードを置換することによって、ならびに、イベントE4、E5およびE6から判定された総計数修正に対応する4つの追加のコンピューティングノードの増加によって、間接的に発生する。

10

【0040】

第3の総計期間260c中、追加のイベントおよびコンピューティングノード容量の利用可能性の変更がさらに発生する。具体的には、変更C7は、およそ時間T22で完了し、この中で、現在利用可能なコンピューティングノードの数は、時間T21で利用可能性が開始された5つの追加のコンピューティングノードを反映するように、5つ増加して合計で12のコンピューティングノードになり、実際のコンピューティングノード数220はこれに応じて更新される。時系列グラフ205に示されるように、総計平均CPU利用205aおよび総計平均ネットワーク帯域幅利用205bの両方は、変更C7後に減少し、総計平均CPU利用205aは急速に低下し、具体的には、この例において、総計平均CPU利用205aは最終的に、トリガ250bに指定された条件に対応する20%の閾値未満に低下し、トリガTR-3 250bが満たされたという判定を含む、イベントE8を発生させ、時系列グラフ205において、総計平均CPU利用205a線上の第2の黒い楕円で図に例示される。トリガが満たされると、合計10のリクエストされた所望のコンピューティングノード数215fに対して、所望のコンピューティングノード数215を2つのコンピューティングノードだけ減少するというリクエストを開始する（または代替として、現在所望のコンピューティングノード数に関係なく、更新された所望のコンピューティングノード数が10に設定されるようにリクエストすることができる）。

20

30

【0041】

最終的に、追加の変更C8は、およそ時間T28の総計期間260cの終了直前に、障害が発生または利用不可能になっている群のコンピューティングノードのうちの1つによって促され、発生する。対応するイベントE10は、変更C8とほぼ同時に発生し、その中で、PESは、必要に応じて利用不可能なコンピューティングノードを終了する。変更C6の様式に同様な様式で、PESは、この例において、利用不可能なコンピューティングノードの置換を即時に開始するのではなく、第3のセットH3の調和アクティビティが時間T31で開始されるまで待機しようとするが（例えば、イベントE8は、その時点でコンピューティングノード群のコンピューティングノードの所望の数が削減されることを可能にする、またはその可能性を高くするため）、他の実施形態において、PESはそうではなく、置換コンピューティングノードの提供を即時に開始する場合がある。具体的には、イベントE2、E3およびE9に関して上述の様式に同様な様式で、イベントE10は、この例において、同時またはほぼ同時に発生する、2つの別個のイベントE10aおよびE10b（別々に図示せず）に分割される。このように、PESは、この例において、利用不可能になったことが検出されているコンピューティングノードに対応するイベントE10aを時間T28で記録し、PESは、変更C8を直接発生させる等、利用不可能なコンピューティングノードを終了するためのアクティビティを時間T28で自動的に開始する。さらに、この例において、イベントE10aの結果として発生する、利用不可能なコンピューティングノードの終了はそれ自体、所望のコンピューティングノード数21

40

50

5を12のコンピューティングノードの所望の数215eに維持するように等、置換コンピューティングノードとして使用するために群の追加のコンピューティングノードのリクエストを開始する、別のイベントE10bとしておよそ時間T28で処理される。この例において、イベントE10bのリクエストは、イベントE9bと同様な様式において処理され、現在の総計期間260cの最後で実施される調和アクティビティH3まで延期されるが、他の実施形態および状況においては、置換コンピューティングノードのそのような提供は、そうではなく、即時に開始される場合がある。加えて、この例において、公式記録のコンピューティングノード数225は、変更C4~C6に関して前述した様式に同様な様式において、変更C8に対応するようには更新されないが、他の実施形態において、公式記録のコンピューティングノード数225は、そのような変更を反映するように更新

10

【0042】

したがって、第3の総計期間260cの最後に、調和アクティビティH3が開始され、コンピューティングノード群への変更となる。具体的には、この例のPESは、この例においてイベントE8およびE10bだけである、期間260b中に発生していたイベントに対応するリクエストされた修正を総計し、更新された現在所望の10のコンピューティングノード数215gに対応するように、この例においてコンピューティングノードを2つ減少することになる、先行の所望の12のコンピューティングノード数215eに動的総計数修正を行うと判定する。さらに、現在の実際のコンピューティングノード数および公式記録のコンピューティングノード数220および225は、調和アクティビティH3

の時点で、それぞれ11のコンピューティングノードおよび12のコンピューティングノードと異なる。したがって、調和アクティビティH3の一部として、PESは、現在の実際のコンピューティングノード限界が11であることを考慮して、コンピューティングノード群から既存のコンピューティングノードのうち1つを削除するように、現在所望の実際の10のコンピューティングノード数に到達するよう、現在公式の12のコンピューティングノード数の変更を開始する（例えば、任意選択的に既に開始したいいくつかまたは全ての動作が完了後に、削除されたコンピューティングノード上のプログラムの実行を終了し、そのコンピューティングノードを他のユーザによる他の将来の使用のために利用可能にする）。このように、この例において、時間T31で既存のコンピューティングノードのうち1つを削除するという容量の利用可能性の変更は、イベントE10bに関して

1つの群コンピューティングノードの終了によって、ならびにイベントE8に対応する2つのコンピューティングノードのリクエストされた減少によって、間接的に発生する。PESはさらに、新しい更新された所望の10のコンピューティングノード数に一致するように、公式記録のコンピューティングノード数225を更新する。T31の直後に、最後の変更C9は、およそ時間T32で完了し、この中で、現在利用可能なコンピューティングノードの数は、時間T31でコンピューティングノード群からの削除が開始したコンピューティングノードを反映するように、合計で10のコンピューティングノードになるように1つ削減され、実際のコンピューティングノード数220はこれに応じて更新される。他の実施形態において、変更C9は、削除されるコンピューティングノードがコンピューティングノード群のさらなるアクティビティから即時に排除される場合等、変更を行う

という判定後の実質的に即時の時間T31に発生する場合があるが、それでも一方で、削除されたコンピューティングノードは、一時的に実行中かつ操作の実施を継続するために利用可能である。

20

30

40

【0043】

加えて、調和アクティビティH1、H2、およびH3はこの例において時間の単一点で発生すると例示されているが、いくつかまたは全ての調和アクティビティは、一定期間にわたって実際に発生する場合があり、さらに一定期間にわたって続行する他の効果を有する場合がある。例えば、少なくともいくつかの実施形態において、PESによって開始されるコンピューティングノード群のプログラム実行容量への何らかまたは全ての変更は、一時的なロックダウン期間になる場合があり、少なくともいくつかの他の種類のイベント

50

または変更が可能ではない場合がある。そのようなロックダウンを発生させる場合がある、P E S が指示するプログラム実行容量変更として、例えば、調和アクティビティの一部として開始されるプログラム実行容量増加および/または減少（例えば、新しいコンピューティングノードを追加、既存のコンピューティングノードを削減する等）、ならびに/あるいはP E S 判定に回答して即時に開始されるプログラム実行容量変更（例えば、コンピューティングノード障害、ユーザ命令の受信、ユーザ指定のトリガが満たされること等）を挙げることができる。そのようなロックダウンは、多様な種類の期間を有する場合があり（例えば、追加されているコンピューティングノードが利用可能になるまで等、指定の結果または発生まで、追加されているコンピューティングノードが利用可能になるまでの平均または予測時間等の指定された期間等）、いくつかの実施形態においては、P E S が開始した変更の種類に基づいて変動する場合がある。そのようなロックダウン中、少なくともいくつかの種類のイベントまたは変更は、ロックダウン期間中、ユーザ指定のトリガが満たされることを可能にしない（または任意のそのようなトリガが満たされたことを無視することによって）、および/またはロックダウン期間中、ユーザ命令が受容されることを可能にしない（または任意のそのようなユーザ命令を無視することによって）等、可能ではない場合がある。さらに、いくつかの実施形態において、コンピューティングノード群に関連するユーザは同様に、ロックダウンの後に続く、またはそうではなく他の時間に有効になる、指定された時間量の冷却期間等、ロックダウンの様式と同様の様式において操作する冷却期間を同様に指定する場合がある。ロックダウンと同様に、冷却期間中、少なくともいくつかの種類のイベントまたは変更は、冷却期間中にユーザ指定のトリガが満たされることを可能にしない（または任意のそのようなトリガが満たされたことを無視することによって）等、可能ではない場合がある。ユーザおよび/またはP E S は、他の実施形態において他の様式でコンピューティングノード群への修正を制御する場合があることが理解される。

10

20

【 0 0 4 4 】

このため、この様式において、P E S は、多様な状況に基づいて、コンピューティングノード群に対する多様な動的修正を行うことを含め、コンピューティングノードの群によってユーザに提供されるプログラム実行容量を管理するように操作する。図 2 A に例示されるイベントおよび変更は、例示の目的のために提供され、その実際のイベント、変更およびP E S の他の操作は、他の実施形態および状況において異なる場合があることが理解される。

30

【 0 0 4 5 】

図 2 B は、P E S S M モジュールの実施形態によって自動的に実施される場合がある技法等、図 2 A に関して前述した動的コンピューティングノード数修正に因果関係情報を自動的に結論付けるための技法の例を例示する。具体的には、図 2 B は、図 2 A に例示される情報のうちのいくつかに対応する情報を記憶する 2 つの例示的なデータベーステーブルデータ構造を例示し、図 2 B の例示的テーブル 2 8 0 は、図 2 A の例示的なコンピューティングノード群（図 2 B において「群 1」と呼ばれる）に発生した例示的なコンピューティングノード容量の利用可能性の変更に関する多様な情報を記憶し、図 2 B の例示的なテーブル 2 9 0 は、図 2 A の例示的なコンピューティングノード群「群 1」に発生した例示的なイベントに関する多様な情報を記憶する。

40

【 0 0 4 6 】

例示的なテーブル 2 8 0 は、図 2 A に関して記載される例示的な変更 C 1 ~ C 1 0 のうちの 1 つに各々対応する、多様な行またはエントリ 2 8 5 a ~ 2 8 5 k を含み、多様なフィールドまたは列 2 8 0 a ~ 2 8 0 x は各行に例示される。具体的には、この例において、各行は、固有の識別子（以下「ID」）2 8 0 a、変更に対応する適用可能なコンピューティングノード群の ID 2 8 0 b、プログラム実行容量変更の種類 2 8 0 c、プログラム実行容量変更の意図された結果の指示 2 8 0 d、変更についての多様な詳細 2 8 0 e、変更の開始および終了時間 2 8 0 f および 2 8 0 g、変更が発生またはそうでなければ変更に関連する総計期間 2 8 0 h（この例において、総計期間の最後に実施される調和アク

50

ティビティに基づいて呼ばれる)、変更イベント結合ID 280x、および任意選択的に多様な他の情報を含む。1つの例証例として、行285aは、群1の変更C1に対応し、これは、総計期間H1中の時間T02に完了される4つのコンピューティングノードの増加に参与する、群1のコンピューティングノード数における変更に対応する。他の行は同様な情報を含む。

【0047】

例示的なテーブル290は、図2Aに関して記載の例示的なイベントE1~E10のうちの1つに各々対応する、多様な行またはエントリ295a~295mを含み、多様なフィールドまたは列290a~290xは各行に例示される。具体的には、この例において、各行は、固有の識別子(以下「ID」)290a、イベントに対応する適用可能なコンピューティングノード群のID290b、イベントの種類290c、イベントの発生源についての情報290d、イベントの時間290e、イベントが発生またはイベントが関連する総計期間290f(この例において、総計期間の最後に実施される調和アクティビティに基づいて称される)、変更イベント結合ID290x、および任意選択的に多様な他の情報(例えば、受信されたユーザ命令に関連する特定のユーザ)を含む。1つの例証例として、行295hは、群1のイベントE4に対応し、それは、群1の所望のコンピューティングノード数を3つのコンピューティングノードだけ増加することをリクエストする、総計期間H2中の時間T19に受信されたユーザ命令に対応する。他の行は同様な情報を含む。

【0048】

図2Bの例において、テーブル280および290の変更イベント結合ID情報280xおよび290xは、変更とイベントとの間の関係を反映するように結論付けられる因果関係情報を反映する。具体的には、この例において、テーブル280および290の行が変更イベント結合ID情報280xおよび290xの共通の値を共有する場合、対応するイベントと変更との間の結論付けられた因果関係を反映する。一例として、図2Aに関して前述したように、イベントE3bは、利用不可能になったコンピューティングノードの置換コンピューティングノードを提供することに対応し、これは、結果として群へのコンピューティングノードの追加に対応する変更C5に直接帰属可能である。したがって、行295e(イベントE3bに対応する)および列285g(変更C5に対応する)は、共通の結合識別子を共有する。

【0049】

これとは対照的に、調和アクティビティH2の間に開始される5つのコンピューティングノードの追加に対応する、変更C7は、その変更に対して直接帰属可能な単一のイベントを有さない。具体的には、イベントE4、E5、およびE6は各々、群1に対して動的修正をリクエストし、それらは、変更C7のための5つのコンピューティングノードのうちの4つの追加を促すように、総計され、組み合わせられて使用された。このように、行295f~295h(イベントE4~E6に対応する)および行285i(変更C7に対応する)は全て共通の結合識別子を共有する。さらに、図2Aの例において、変更C7の一部として追加された5つのコンピューティングノードのうちの1つは、変更C6に関して利用不可能になったコンピューティングノードの置換ノードであり、置換コンピューティングノードの開始は、イベントE9bに対応する。したがって、行295j(イベントE9bに対応する)および列285i(変更C7に対応する)も、同一の共通の結合識別子を有する。他の実施形態において、置換コンピューティングノードの提供が、利用不可能なコンピューティングノードを即時に置換するため等、イベントE4~E6に対応する4つの追加のコンピューティングノードの提供とは別にイベントE9bにおいて開始された場合、それに対応する変更は、期間260bの終了および結果の調和アクティビティH2の前に発生し、テーブル280において別々に示され、イベントE9bおよびその追加の変更は、別個の対応する結合識別子を共有する。加えて、変更C7のために追加される5つのコンピューティングノードは、テーブル280の単一の行に総計された様式で示されるが、他の実施形態においては、追加される各コンピューティングノードは、テーブル2

10

20

30

40

50

80の独立した行285に表される場合があり、その場合、各々は、総計された変更Cの行285 iだけに現在例示されている、同じ結合識別子N7を共有する。

【0050】

さらに、この例において、動的ユーザ命令および/または満たされたユーザトリガに基づくだけでなく、コンピューティングノード数に対する不注意の変更に基づいて（例えば、コンピューティングノードの障害発生または利用不可能になることに起因して）、コンピューティングノード数に対する動的修正に対応する変更がテーブル280内で追跡される。そのような変更は、アクティビティの監視に基づいて検出され、追加の変更をトリガするイベントとして例示されるが（例えば、置換コンピューティングノードの提供）、他の実施形態においては、そのような変更を追跡しない、および/またはシステム起動の置換アクティビティをイベントとして処理しない等、他の様式で処理される場合がある。さらに、テーブル280および290の変更イベント結合情報は、1つ以上の対応するイベントに直接帰属可能な変更（例えば、それらの変更を即時に開始させるイベント）と、1つ以上の対応するイベントに間接的に帰属可能な変更（例えば、調和アクティビティの間等、共に総計され、組み合わせられたイベントが、それらの変更を開始させる）との間を区別しないが、他の実施形態においては、そのような情報はさらに追跡される場合がある。

【0051】

前述のように、コンピューティングノード群に対する特定の動的プログラム実行容量修正に対する責任の帰属は、なぜ変更が発生したかの説明をコンピューティングノード群に関連するユーザに提供する場合を含めて、多様な利益を提供する場合がある。そのような責任帰属情報は、どのイベントが群の1つ以上のコンピューティングノードの利用可能性における特定の指示されたプログラム実行容量修正または他の変更の原因であるか、および/またはどのプログラム実行容量修正または他のコンピューティングノード群の利用可能性の変更が1つ以上の指示されたイベントによって発生するかを識別するリクエスト等、ユーザまたは他の情報源から受信される多様な種類のクエリに回答して、さらに、生成および/または使用される場合がある。図2Bに関する記載のように、テーブル280および290に例示される結合情報は、責任帰属情報を追跡し、提供するための1つの機構を提供する。例えば、図2Aの変更C7に対応する5つのコンピューティングノードの追加に関して、ユーザは、5つのコンピューティングノードがなぜ追加されたかを知りたい場合がある（例えば、特に、3つのコンピューティングノードを追加するというイベントE4のユーザ命令を鑑みて）。図2Bに例示される結合情報を使用することによって、PESまたは他のシステムは、人間可読説明を容易に自動的に生成することができる。例えば、図2Aの変更C1に対応する4つのコンピューティングノードを追加する原因に関するユーザリクエストに回答して、PESまたは他のシステムは、例えば、「変更C01は、時間T1にイベントE01によって直接発生した。イベントE01は、8つのコンピューティングノードの群を開始するというユーザUUUからのリクエストだった」ということを示すことができる。この情報は、イベントE1に対応するテーブル290の行295aを識別するために、変更C1に対応する行285aからの結合情報280xを使用すること、および所望の様式で行285aおよび295aから情報を抽出し、フォーマットすることに基づいて（例えば、ユーザリクエストまたは先行のユーザの希望設定に基づいた形式、PESデフォルトに基づいた形式等）生成される場合がある。同様に、8つのコンピューティングノードの群を開始するユーザ命令E1の効果としてユーザリクエストに回答して、PESまたは他のシステムは、例えば、「時間T1のイベントE01は、変更C01、C02およびC03を直接発生させる。変更C01は、T1で開始し、時間T2で終了した4つのコンピューティングノードの追加だった。変更C02は、T1で開始し、時間T3で終了した3つのコンピューティングノードの追加だった。変更C03は、時間T1で開始し、完了しなかった1つのコンピューティングノードの追加だった。」と示す場合がある。別の例として、図2Aの変更C7に対応する5つのコンピューティングノードを追加する原因としてユーザリクエストに回答して、PESまたは他のシステムは、例えば、「変更C07は、T11からT21の期間中に発生したイベントE05、E06、E

10

20

30

40

50

04、およびE09bによって間接的に発生した。イベントE05は、トリガTR-1が満たされたことに基づいて、4つの追加のコンピューティングノードのリクエストだった。イベントE06は、トリガTR-Nが満たされたことに基づいて、2つの追加のコンピューティングノードのリクエストだった。イベントE04は、ユーザUUUからの動的ユーザ入力命令に基づく3つの追加のコンピューティングノードのリクエストだった。イベントE09bは、利用不可能になった群のコンピューティングノードの自動終了に基づいて、1つの置換コンピューティングノードのリクエストだった。」と示す場合がある。そのようなテキスト文字列に加えて、またはこの代わりに、責任帰属情報はさらに、自動報告書(例えば、テーブル、図等)等の多様な他の形式で定期的または他の予定様式で生成される場合がある。そのような責任帰属情報は、他の実施形態において多様な他の様式で生成され、使用される場合があることが理解される。

10

【0052】

図2Bに例示されるイベントおよび変更に関する情報は、例示目的のために提供され、記憶される情報およびそのような情報の記憶は、他の実施形態において多様な他の様式で実施される場合があることが理解される。さらに、PESは、ユーザ定義トリガについて、監視された性能測定情報について、ユーザアカウントについて等の情報を記憶する追加のテーブルを有する等、他の実施形態においてユーザおよびコンピューティングノード群についての多様な追加の種類情報を記憶する場合がある。

【0053】

加えて、図2Aおよび2Bの前述の例は、例示目的に提供され、他の実施形態は多様な方式で例とは異なる場合がある。例えば、プログラム実行容量は、多様な利用可能なコンピューティングノードが同等として処理される場合等(例えば、同等のコンピューティングリソースを有する)、これらの例ではコンピューティングノードの数に基づいて測定され、修正されるが、多様な利用可能なコンピューティングノードは、変動する特性を有する異なる種類である場合(例えば、異なる量の処理容量、メモリ、プラットフォーム仕様等)、および/またはプログラム実行容量がコンピューティングノード数以外の様式で追跡される、他の実施形態が提供される。いくつかのそのような実施形態において、多様なリクエストは、リクエストに関連するプログラムを実行するために選択されたコンピューティングノードの群において使用するための1つ以上の特定の種類のコンピューティングノードの指示を含む場合があり、それらのリクエストは、対応する指定の種類コンピューティングノード上でのみ達成される場合がある。

20

30

【0054】

加えて、少なくともいくつかの実施形態において、PESまたは他のシステムはさらに、コンピューティングノードの群を管理する一部として、他の種類のアクティビティを実施する場合がある。例えば、前述のように、PESまたは他のシステムは、多様な時点でコンピューティングノード群からコンピューティングノードを追加または削除するように判定する場合があり、期間中に満たされた1つ以上のトリガおよび/または期間中に受信されたユーザ命令に回答して総計期間の最後の調和アクティビティ中、即時に効果を有することが指定または判定されたユーザ命令またはトリガを満たすことに回答して任意の時間に、障害が発生したコンピューティングノードを置換および/または進行中の操作が禁止されているコンピューティングノードを終了して削除するように等、PESまたは他のシステムの自動アクティビティに回答して任意の時点で等を含む。少なくともいくつかの実施形態において、PESまたは他のシステムはさらに、少なくともいくつかのそのようなコンピューティングノード群修正のための正確なタイミングを判定する時に、他の因子を考慮する場合がある。例えば、ユーザが所与の期間の所与の数のコンピューティングノードの使用に対して責任を負わされる状況において(例えば、次の時間X個のコンピューティングノードに対して既に課金されている)、そのような情報は、コンピューティングノード群に対する少なくともいくつかの種類修正に関するタイミングを判定するために使用される場合がある。PESまたは他のシステムが、群のコンピューティングノードの数をX未満に削減すると判定した場合、PESまたは他のシステムは、コンピューティン

40

50

グノード数を X 未満に実際に削減する前に、所与の期間の最後近辺または最後（例えば、ユーザが既に課金されている時間の最後）まで待機すると判定する場合がある。同様に、群のコンピューティングノードが、新しい調和アクティビティが実施される時点近辺（例えば、リクエストされたコンピューティングノード数修正が総計されている期間の最後近辺）に障害を発生した場合、PES または他のシステムは、障害を反映するように実際のノード数および任意選択的に公式記録のノード数を更新するが、置換コンピューティングノードの追加を即時に開始しないと判定する場合があり、この様式では、調和アクティビティ中のリクエストされたコンピューティングノード数修正の総計が、群のコンピューティングノード数を 1 つ以上のコンピューティングノードだけ削減すると決定した場合、群の既存のコンピューティングノードを終了するのではなく、障害を発生したコンピューティングノードを削減する 1 つとして使用する場合がある。

10

【 0 0 5 5 】

いくつかの実施形態において、PES または他のシステムは、コンピューティングノード群の複数の別個の種類を同時に管理する場合がある。一例として、コンピューティングノード群は、単一の場所にあるコンピューティングノードを使用して現在所望のコンピューティングノード数に管理されている場合があるが、ユーザは、群のコンピューティングノードのうちの指定された所望のサブセットが、別個の第 2 の場所に配置されるように、コンピューティングノード群に対する第 2 の種類のメトリックを指定すると判断する場合がある。その場合、PES または他のシステムは、第 2 のメトリックの所望の値に到達するまで、第 2 の場所にある群に任意の新しいコンピューティングノードを増分的に追加し、第 1 の場所にある群から任意の既存のコンピューティングノードを増分的に削除することによって、または代替として、置換コンピューティングノードが第 2 の場所にある群に追加されることを可能にするように、第 1 の場所にある群から既存のコンピューティングノードを即時に終了することによって、を含む、多様な様式において、第 2 のメトリックの所望の値を満たすように操作する場合がある。そのような増分の追加および/または削除は、利用不可能なコンピューティングノードを置換する、動的に指定されたユーザ命令および/または満たされたユーザ指定のトリガにตอบสนองしてコンピューティングノード数を修正する等を含む、他の部分に詳細を記載される様式のうちのいずれかにおいてトリガされる場合がある。加えて、コンピューティングノード数および場所は、この例において同時にバランスされるが、多様な他の種類の変更が同様な様式で（例えば、異なる関連量のコンピューティングリソースを有する異なる種類に基づいて等、既存のコンピューティングノードを第 1 の種類から第 2 の種類に変更するように）実施される場合がある。さらに、プログラム実行容量修正は、いくつかの例において、コンピューティングノード数を変更することによって行われるが、他の実施形態において、そのような容量修正は、群の 1 つ以上のコンピューティングノードのプログラム実行容量を変更することによって行われる場合がある（例えば、第 1 のコンピューティングノードを対象のコンピューティングリソースの事実上 1 つ以上の種類を有する第 2 のコンピューティングノードと置換することによって、既に群の一部であるコンピューティングノードのうちの 1 つに対して利用可能な 1 つ以上のコンピューティングリソースの量を修正することによって等）。

20

30

【 0 0 5 6 】

前述のように、多様な種類の機能は、多様な実施形態において PES によって提供され、使用される場合があり、機能は多様な方式で提供される場合がある。例えば、いくつかの実施形態において、PES から利用可能なプログラム実行容量は、コンピューティングマシンの間でデータを伝送することが可能な 1 つ以上のネットワークまたは他のデータ交換媒体を介して相互接続された複数の物理的コンピューティングマシンを介して等、ユーザの代わりにプログラムを実行するための複数のコンピューティングノードを含む場合がある。コンピューティングマシンのうちの少なくともいくつかは、いくつかの実施形態において、各々複数のプログラムを同時に実行するに十分なコンピューティング関連リソースを含み（例えば、十分な書き込み可能メモリ、不揮発性記憶装置、CPU サイクルまたは他の CPU 使用量測定値、ネットワーク帯域幅、スワップスペース等）、いくつかのそ

40

50

のような実施形態におけるコンピューティングマシンのうちの少なくともいくつかは各々、別個のユーザの代わりに1つ以上のプログラムを各々実行する場合がある複数の仮想マシンコンピューティングノードをホストする場合がある。さらに、多様な実施形態において、PESは、複数のユーザの代わりに、多様な種類のプログラムを実行する場合がある。例えば、ユーザの代わりに実行されるそのようなプログラムは、1つ以上のオペレーティングシステム、アプリケーション（例えば、サーバおよび/または他のソフトウェアアプリケーション）、ユーティリティ、ライブラリ等を含む場合があり、加えて、少なくともいくつかの実施形態において、そのようなプログラムは、1つ以上の仮想マシンコンピューティングノード上でブート可能またはロード可能で、各々がオペレーティングシステムソフトウェア、1つ以上のアプリケーションプログラムのためのソフトウェア、および/または構成情報等を含む場合がある、仮想マシンイメージ等、実行可能なソフトウェアイメージを含む場合がある。

10

【0057】

少なくともいくつかの実施形態において、PESによる1つ以上のコンピューティングノードの群上での1つ以上のプログラムの実行は、それらのプログラムの即時実行に対する現在の実行リクエストにตอบสนองして開始される場合がある。代替として、開始は、今現在のためにそれらのプログラムのその後将来の実行を予定またはそうでなければ予約する、以前に受信されたプログラム実行リクエストに基づく場合がある。プログラム実行リクエストは、ユーザから直接（例えば、プログラム実行サービスによって提供される相互作用型コンソールまたは他のGUIを介して）、または1つ以上の他のプログラムまたはそれ自体の他のインスタンスの実行を自動的に開始するユーザの実行中のプログラムから（例えば、ウェブサービスを使用するAPI等、プログラム実行サービスによって提供されるAPIを介して）のように、多様な方式で受信される場合がある。プログラム実行リクエストは、実行されるプログラムの実行可能または他のコピー、実行のために以前に登録または入力されたプログラムの指示、および同時に実行されるプログラムのいくつかのインスタンス（例えば、単一の所望のいくつかのインスタンスとして、最小および最大数の所望のインスタンスとして等）、ならびに1つ以上のプログラムの実行のための多様な他の種類の詳細設定および/または要件（例えば、リソース割り当て、実行の地理的および/または論理的場所、他のプログラムおよび/またはコンピューティングノードに対する実行の近似性、タイミング関係条件等）等、1つ以上のプログラムの実行の開始において使用される多様な情報を含む場合がある。

20

30

【0058】

指示された時間にプログラムの1つ以上のインスタンスを実行するリクエストを受信した後、PESは、プログラムインスタンスを実行するために群の中で使用する1つ以上のコンピューティングノードを判定することができる。いくつかの実施形態において、使用されるコンピューティングノードの判定は、将来の実行の場合であってもリクエストの時に実施される。他の実施形態において、1つ以上のプログラムインスタンスの将来の実行のために使用されるコンピューティングノードの判定は、その時点で利用可能な情報に基づく実行の将来の時間等、後の時間まで延期される場合がある。いくつかの実施形態において、ユーザの代わりに1つ以上のプログラムの実行のためにどのコンピューティングノードを使用するかを判定は、ユーザがPESを使用するために加入および/または登録する時、および/またはユーザのためにプログラムを実行するリクエストに先行する別の時間等、実行するリクエストに先行して行われる場合がある。例えば、いくつかのそのような実施形態において、1つ以上のコンピューティングノードは、期間中ユーザのためにソフトウェアを実行するリクエストが受信された任意の時間等、その期間中の任意の時間に関連するコンピューティングノード上でそのユーザの代わりに実行される場合がある等、一定期間ユーザに関連付けられる場合がある。加えて、いくつかの実施形態において、ユーザの代わりにプログラムを実行するためにどのコンピューティングノードを使用するかを判定は、例えば、コンピューティングノードが未使用および/またはそうでなければプログラムを実行するために利用可能である時に1つ以上のコンピューティングノード上で

40

50

1つ以上の保留中のリクエストのプログラムを実行する等、1つ以上のコンピューティングノードおよび/または1つ以上のコンピューティングノードのコンピューティングリソースがユーザのためにプログラムを実行するために利用可能になる時に行われる場合がある。

【0059】

各プログラムコピーまたはインスタンスの実行のためにどのコンピューティングノードを使用するかの判定は、リクエストに指定あるいはそうでなければプログラムおよび/または関連するユーザに指定された任意の詳細設定および/または要件に基づいて（例えば、登録時等）を含む、多様な方式で行われる場合がある。例えば、プログラムインスタンスの実行のために好ましいおよび/または必要なリソースの条件が判定される場合（例えば、メモリおよび/または記憶装置、CPUタイプ、サイクルまたは他の性能メトリック、ネットワーク容量、プラットフォームタイプ等）、プログラムインスタンスを実行するために適したコンピューティングノードの判定は、少なくとも部分的に、コンピューティングノードがそれらのリソース条件を満たすために利用可能な充分なリソースを有しているかどうかに基づく場合がある。プログラム実行サービス上でプログラムを実行および実行するように構成することに関する追加の詳細は、2006年3月31日付提出の係属中米国特許出願第11/395,463号、「Managing Execution of Programs by Multiple Computing Systems」に含まれ、参照によりその全体を本明細書に組み入れる。

10

【0060】

いくつかの実施形態において、PESがユーザによる1つ以上の料金の支払いの交換にそのユーザのためにプログラムを実行する場合があるように、料金は、PESの使用に関連付けられる場合がある。例えば、いくつかの実施形態において、ユーザのプログラムを実行するために割り当てられた、いくつかの処理ユニットのうちの1つ以上、メモリの量、記憶装置の量、ネットワークリソースの量等に基づいて等、ユーザの代わりに1つ以上のプログラムを実行するために割り当てられたプログラム実行容量の量および/または種類に基づいて、料金がユーザに課金される場合がある。いくつかの実施形態において、料金は、例えば、CPU能力または性能、プラットフォームタイプ（例えば、32ビット、64ビット等）等に基づく等、プログラムを実行するために使用されたコンピューティングリソースの多様な特性等の他の因子に基づく場合がある。いくつかの実施形態において、料金は、サービスの使用あたりの価格、そのコンピューティングサービスが使用された時間単位あたりの価格、使用された記憶装置あたりの価格、受信および/または送信されたデータあたりの価格等のように、多様な使用因子のベースで課金される場合がある。少なくともいくつかの実施形態において、以下に詳細を記載するように、料金は、プログラムの実行に関する多様な特性等（例えば、実行の継続性、耐障害性等）、多様な他の因子に基づく場合がある。少なくともいくつかの実施形態において、PESのプロバイダは、サービスの多様な料金帯、種類および/またはレベル、あるいは複数のユーザの代わりにプログラムを実行するための機能のうちの1つ以上を提示する場合があり、いくつかのそのような実施形態において、多様な料金は、サービスの多様な料金帯、種類および/またはレベルに関連付けられる場合がある。加えて、例えば、第1の量のプログラム実行容量機能に対して第1の料金帯で料金を課金し（例えば、使用されているコンピューティングノードの指定された第1の閾値まで）、第2の量のプログラム実行容量機能に対して第2の料金帯（例えば、より低い価格帯）で料金を課金する（例えば、指定された第1の閾値から使用されているコンピューティングノードの指定された第2の閾値まで）等、料金帯は、PESによって提供される特定の種類の機能に対して使用される場合がある。プログラム実行サービスに関連する多様な料金に関する追加の詳細は、2007年12月21日付提出の係属中米国特許出願第11/963,331号、「Providing Configurable Pricing for Execution of Software Images」に含まれ、参照によりその全体を本明細書に組み入れる。

20

30

40

【0061】

さらに、多様な他の種類の機能は、他の部分に詳細を記載するように、多様な実施形態においてPESによって提供され、使用される場合がある。

50

【 0 0 6 2 】

図3は、複数のユーザのためにコンピューティングノードの群を管理する技法を実施するために適したシステムの例示的实施形態を例示する模式図である。具体的には、図3は、プログラム実行サービスの少なくともいくつかの機能を提供するために適したサーバコンピューティングシステム300、ならびにプログラム実行サービスのユーザによって使用される場合がある多様なクライアントコンピューティングシステム350、プログラム実行サービスによって使用される場合があるコンピューティングノード360、および他のコンピューティングシステム380を例示する。例示された実施形態において、サーバコンピューティングシステム300は、CPU305、多様なI/O構成要素310、記憶装置320、およびメモリ330を含む、構成要素を有する。例示されたI/O構成要素は、ディスプレイ311、ネットワーク接続312、コンピュータ可読媒体ドライブ313、および他のI/Oデバイス315（例えば、キーボード、マウス、スピーカ等）を含む。加えて、例示されたユーザコンピューティングシステム350は、CPU351、I/O構成要素352、記憶装置354、およびメモリ357を含む、サーバコンピューティングシステム300の構成要素と同様な構成要素を有する。他のコンピューティングシステム380およびコンピューティングノード360も各々、サーバコンピューティングシステム300に関して例示された構成要素のうちのいくつかまたは全てと同様な構成要素を含む場合があるが、そのような構成要素は、簡潔性のためにこの例には例示されない。

10

【 0 0 6 3 】

プログラム実行サービスシステムマネージャモジュール340の実施形態は、メモリ330の中で実行中であり、ネットワーク390上から（例えば、インターネットおよび/またはワールドワイドウェブを介して、非公開セルラーネットワークを介して等）コンピューティングシステム350および380ならびにコンピューティングノード360と相互作用する。この例示的实施形態において、PESSM340は、PESSM340によって管理されるプログラム実行サービスと共に等、ユーザコンピューティングシステム350と相互作用している多様なユーザ（図示せず）による複数のコンピューティングノード360の使用を管理することに関する機能を含む。他のコンピューティングシステム350および380ならびにコンピューティングノード360は、PESSMとの相互作用の一部として多様なソフトウェアを実行している場合がある。例えば、ユーザコンピューティングシステム350は、他の部分に詳細に記載するように、多様な方式で1つ以上のコンピューティングノード360上でそれらのシステムのユーザの代わりにプログラムの実行を構成および/またはリクエストするため、ならびに多様な他の種類の動作を実施するため等、PESSM340と相互作用するために（例えば、ウェブブラウザまたは専門のクライアント側アプリケーションプログラムの一部として）メモリ357の中のソフトウェアを実行している場合がある。複数のユーザの代わりにプログラムを実行するための構成、実行および/または登録に関する情報322、コンピューティングノード群のプログラム実行容量修正に関する情報324（例えば、予め定義されたユーザ指定のトリガについての情報、ユーザからの動的に指定された容量修正命令についての情報、コンピューティングノード性能測定についての情報および指定されたトリガが満たされたかどうかの判定に関する他の情報、群に対する所望の、実際の、および公式記録のコンピューティングノード数の現在値等）、および特定のコンピューティングノード群の特定のプログラム実行容量修正の因果関係の帰属に関する情報326（例えば、発生する少なくともいくつかのイベントおよび群のコンピューティングノードに対する少なくともいくつかの変更をリストし、図2Bおよび他の部分に記載される様式に類似の様式等で、特定のイベントを特定の変更に関連付けることによって）等、PESSMモジュール340の機能に関する多様な情報は、記憶装置320に記憶される場合がある。

20

30

40

【 0 0 6 4 】

PESSMモジュール340が、1つ以上のコンピューティングノード360の群上で1つ以上のプログラムを実行するというリクエスト（または他の指示）を受信した後、P

50

ESSMモジュール340は、群の1つ以上のコンピューティングノードを選択し、それらのコンピューティングノード360上でそれらのプログラムの実行を開始する。加えて、PESSMモジュール340はさらに、コンピューティングノード上で開始されたプログラムの実行を後で終了する、プログラムのうちの1つ以上を1つ以上の他のコンピューティングノード360またはコンピューティングシステム380に移行する等のために、コンピューティングノード360と相互作用する場合がある。コンピューティングノード360は、いくつかの物理的コンピューティングシステムおよび/または1つ以上の物理的コンピューティングシステム上で実行中のいくつかの仮想マシンを含む等、多様な実施形態において多様な形態を有する場合がある。いくつかの実施形態において、サーバコンピューティングシステム300およびコンピューティングノード360は、データセンタまたは共同設置されたコンピューティングシステムの他の群の一部の場合、または非公開ネットワークのコンピューティングノードの場合がある。加えて、いくつかの実施形態において、PESSMモジュール340は、コンピューティングシステム380が、他のユーザによる使用のためにコンピューティングノードを提供している1つ以上の第三者参加者によって提供される場合等、それらのコンピューティングシステム上の1つ以上のプログラムの実行を開始または終了するために1つ以上の他のコンピューティングシステム380と相互作用する場合がある。いくつかの実施形態において、PESSMモジュール340はさらに、または代わりに、プログラム実行サービス以外の1つ以上の種類のコンピューティング関係リソースまたはサービスに対するアクセスを管理する場合がある(例えば、持続的または一時的なデータ保管サービス、メッセージングサービス、データベースサービス等)。

10

20

【0065】

加えて、プログラム実行サービス容量保守マネージャモジュール345の実施形態は、メモリ330の中で実行中であり、この実施形態においては、ネットワーク390上からコンピューティングノード360と相互作用する。具体的には、この例示的实施形態において、PESCMMモジュール345は、コンピューティングノード360のうちの1つ以上を監視またはこれらとの相互作用に関する機能を含み、コンピューティングノード群の現在の実際のプログラム実行容量を判定、および/またはコンピューティングノード群のいくつかまたは全てのコンピューティングノードに対応する現在の性能特性を判定する等、それらのコンピューティングノードの使用を追跡する。前述のように、そのような情報は、記憶装置320および/またはそれ以外の場所に記憶される場合があり、多様な実施形態においてモジュール340および345によって使用される場合がある。例えば、いくつかの実施形態において、モジュール345が、コンピューティングノードが障害を発生または利用不可能になったことを発見した場合(例えば、コンピューティングノードがコンピューティングノード群の一部として使用されていた後、コンピューティングノード群の一部として使用されるようにコンピューティングノードをプロビジョニングまたはそうでなければ開始している一部として等)、モジュール345は、利用不可能なコンピューティングノードを新しいコンピューティングノードと置換する動作を自動的に行う場合がある。他の実施形態において、そうではなく、モジュール345が代わりに別の情報源から、コンピューティングノード群の現在の実際のプログラム実行容量および/またはコンピューティングノード群のいくつかまたは全てのコンピューティングノードに対応する現在の性能特性についての情報を取得した場合、その情報を使用して、適切にコンピューティングノードのプログラム実行容量を維持する等、モジュール345はコンピューティングノードの監視のいくつかまたは全てを実施しない場合がある。

30

40

【0066】

コンピューティングシステム300、350および380ならびにコンピューティングノード360は、例示に過ぎず、本発明の範囲を限定することを意図するものではないことが理解される。コンピューティングシステムおよび/またはノードは各々、複数の相互作用しているコンピューティングシステムまたはデバイスを含み、コンピューティングシステム/ノードは、インターネット等の1つ以上のネットワークを通じて、ウェブを介し

50

て、または非公開ネットワークを介して（例えば、移動体通信ネットワーク等）を含む、例示されていない他のデバイスに接続される場合がある。より一般的には、コンピューティングノードまたは他のコンピューティングシステムは、デスクトップまたは他のコンピュータ、データベースサーバ、ネットワーク記憶装置および他のネットワークデバイス、PDA、携帯電話、無線電話、ポケットベル、電子手帳、インターネット機器、テレビベースのシステム（例えば、セットトップボックスおよび/またはパーソナル/デジタルビデオレコーダ）、ならびに適切な通信能力を含む多様な他の消費者製品を含むがこれらに限定されない、記載の種類機能と相互作用し、これらを実施する場合がある、ハードウェアまたはソフトウェアの任意の組み合わせを備える場合がある。加えて、例示されるモジュール340および/または345によって提供される機能は、いくつかの実施形態において、追加のモジュールに分散される場合がある。同様に、いくつかの実施形態において、モジュール340および/または345の機能のうちのいくつかは、提供されない場合がある、ならびに/もしくは、他の追加の機能が利用可能な場合がある。

10

【0067】

また、多様な項目は使用されている間はメモリまたは記憶装置に記憶されると例示されるが、これらの項目またはそれらの部分は、メモリの管理およびデータの整合性の目的で、メモリと他の記憶装置との間を転送される場合がある。代替として、他の実施形態において、ソフトウェアモジュールおよび/またはシステムのうちのいくつかまたは全ては、別のデバイス上のメモリの中で実行し、コンピュータ間の通信を介して例示のコンピューティングシステムと通信する場合がある。さらに、いくつかの実施形態において、システムおよび/またはモジュールのうちのいくつかまたは全ては、1つ以上のアプリケーション専用集積回路（ASIC）、標準集積回路、制御装置（例えば、適切な命令を実行することによって、マイクロコントローラおよび/または埋め込みコントローラを含む）、フィールドプログラマブルゲートアレイ（FPGA）、コンプレックスプログラマブルロジックデバイス（CPLD）等を含むがこれらに限定されない、少なくとも部分的にファームウェアおよび/またはハードウェア等、他の様式で実装または提供される場合がある。モジュール、システム、およびデータ構造のうちのいくつかまたは全てはまた、ハードディスク、メモリ、ネットワークまたは適切な接続を介して適切なドライブによって読み取られる携帯可能媒体品等のコンピュータ可読媒体上に記憶される場合がある（例えば、ソフトウェア命令または構造化されたデータとして）。システム、モジュール、およびデータ構造はまた、ワイヤレスベースおよびワイヤ/ケーブルベースの媒体を含む、多様なコンピュータ可読伝送媒体上で生成されたデータ信号として伝送される場合があり（例えば、搬送波、または他のアナログまたはデジタル伝播波の一部として）、多様な形態をとる場合がある（例えば、単一または多重アナログ信号の一部として、または複数の離散デジタルパケットまたはフレームとして）。そのようなコンピュータプログラム製品はまた、他の実施形態において他の形態をとる場合がある。したがって、本発明は、他のコンピュータシステム構成で実践される場合がある。

20

30

【0068】

図4は、プログラム実行サービスシステムマネージャルーチン400の例示の実施形態のフロー図である。ルーチンは、ユーザのためにコンピューティングノードの群の使用を管理する際に支援する、ならびにいくつかの状況において他の種類の管理操作を実施する等、例えば、それぞれ図1Aおよび1BのPESSMモジュール110および180、および/または図3のPESSMモジュール340の実行によって提供される場合がある。この例示された実施形態において、ルーチン400は、複数のユーザの代わりにプログラムを実行するためのプログラム実行容量を提供する、プログラム実行サービスの使用の多様な態様を管理する。

40

【0069】

例示された実施形態において、ルーチンはブロック405で開始し、情報またはリクエストが受信される。ルーチンはブロック410に続き、受信されたリクエストまたは情報が、ユーザからのリクエスト等、コンピューティングノードの群上で1つ以上のプログラ

50

ムの実行を開始することに関連するかを判定する。関連する場合、ルーチンはブロック 4 1 5 に続き、コンピューティングノード群のプログラム実行容量の所期の所望の量等（例えば、所望のコンピューティングノード数）、リクエストされたプログラム実行に関する情報、実行される 1 つ以上のプログラムに関する情報（任意選択）、および 1 つ以上のユーザ指定の容量修正トリガに関する情報（任意選択）を取得する。他の部分に記載するように、いくつかの実施形態において、ユーザは、多様な種類のコンピューティングノードのうちの一つ以上から選択する場合、および/または所望のコンピューティングリソースの多様な量および/または種類を指定する場合がある（例えば、処理ユニットタイプ/量、メモリ量、プラットフォーム仕様等）。ブロック 4 2 0 で、ルーチンは次いで、群で使用されるコンピューティングノードを選択し、ブロック 4 2 5 で、選択されたコンピューティングノードをプロビジョニングし、実行される 1 つ以上のプログラムの実行を任意選択的に開始することによって、それらの選択されたコンピューティングノードをユーザのために利用可能にすることを開始する。ルーチンはまた、群の選択されたコンピューティングノードの所望のプログラム実行容量を、群の初期の公式記録のプログラム実行容量として指定する。コンピューティングノードがユーザの代わりに使用するために利用可能である時、ユーザは、多様な様式で利用可能性が通知される場合、または他の実施形態において、コンピューティングノードは、ユーザによってさらに相互作用することなく、自動化された様式で操作する場合がある。ルーチンは次いでブロック 4 3 0 に続き、群に対する任意のユーザ指定のトリガを含む、コンピューティングノード群に関する情報を記憶する。

10

20

【 0 0 7 0 】

そうではなく、ブロック 4 1 0 でコンピューティングノードの群の実行を開始するというリクエストが受信されなかったと判定された場合、ルーチンは代わりにブロック 4 4 0 に続き、既存のコンピューティングノード群のプログラム実行容量を修正することに関するリクエストが受信されたかどうかを判定する。受信された場合、ルーチンはブロック 4 4 5 に続き、指示された既存のコンピューティングノード群に対してプログラム実行容量を修正することに関する動的に指定されたユーザ命令を受信し、記憶する。例示された実施形態において、ユーザ命令は、現在の総計期間中に発生する他の可能なプログラム実行容量修正リクエストと総計され、ブロック 4 6 5 ~ 4 7 7 に関して等、調和アクティビティが実施される次の時間中にさらに処理される場合があるが、他の実施形態において、少なくともいくつかのそのようなユーザ指定の修正リクエストは、代わりに即時に実施される場合がある。

30

【 0 0 7 1 】

その代わりにブロック 4 4 0 で、コンピューティングノードの群のプログラム実行容量を修正するというリクエストが受信されなかった場合、ルーチンは代わりにブロック 4 6 0 に続き、そのようなコンピューティングノード群のプログラム実行容量修正リクエストの総計の期間の最後等に、1 つ以上のコンピューティングノード群のプログラム実行容量に関する定期的または再帰的調和アクティビティを現在実施するかどうかを判定する。実施する場合、ルーチンはブロック 4 6 5 に続き、調和アクティビティを現在実施する 1 つ以上のコンピューティングノード群を判定し（例えば、総計期間が終了し、1 つ以上の動的プログラム実行容量修正が総計されているコンピューティングノード群）、ブロック 4 6 7 で、先頭から開始して、次の判定されたコンピューティングノード群を選択する。ルーチンは次いでブロック 4 7 0 に続き、再帰的容量調和アクティビティを実施し、そのようなルーチンの一例は、図 5 に関して詳細に記載する。ルーチンは次にブロック 4 7 5 に続き、ブロック 4 7 0 で実施されたプログラム実行容量修正に対する因果関係の帰属に関するアクティビティを実施するルーチンを実行し、そのようなルーチンの一例は、図 6 に関して詳細に記載する。ブロック 4 7 5 の後、ルーチンはブロック 4 7 7 に続き、処理することが判定されたコンピューティングノード群がさらに存在するかを判定し、存在する場合、ルーチンはブロック 4 6 7 に続き、次のそのように判定されたコンピューティングノード群を選択する。

40

50

【 0 0 7 2 】

その代わりに、ブロック 4 6 0 で、1 つ以上のコンピューティングノード群のプログラム実行容量に関して定期的または再帰的調和アクティビティを現在実施しないと判定される場合、ルーチンは代わりにブロック 4 8 0 に続き、1 つ以上の他の指示された操作を任意選択的に実施する。そのような操作は、例えば、単一のコンピューティングノード上で単一のプログラムを実行する等、他の種類のプログラム実行を実施することに関するユーザリクエスト（提供されるプログラム実行サービスがそのような他のプログラム実行の種類を提供する場合）、即時に実施されるユーザ指定のプログラム実行容量修正リクエスト（例えば、コンピューティングノードが正しく操作していない場合等、特定の指示されたコンピューティングノードの実行を終了するというユーザ命令）、追加のトリガを指定、
10
または指示されたコンピューティングノード群の構成情報を修正するというユーザリクエスト、再帰的調和アクティビティに追加またはその代わりに等、指示されたコンピューティングノード群に対して即時に調和アクティビティを実施するというユーザリクエスト（例えば、調和アクティビティがユーザリクエスト時のみに実施される場合）、ユーザが関連する 1 つ以上のコンピューティングノード群に関する多様なステータス情報を取得するというユーザリクエスト、加入、登録、または支払い操作のように、ユーザの管理に関するアクティビティを実施するというリクエスト等、のうちの 1 つ以上を含む場合がある。

【 0 0 7 3 】

ブロック 4 3 0、4 4 5、または 4 8 0 の後、または代わりにブロック 4 7 7 で追加の判定されたコンピューティングノード群が利用可能ではないと判定された場合、ルーチン
20
はブロック 4 8 5 に続き、ユーザに提供されたプログラム実行機能に基づいて 1 人以上のユーザに課金および/またはユーザから集金する等、ブロック 4 0 5 で受信された情報またはリクエストに基づいてまたはその他開始される（例えば、定期的）任意のユーザ請求（または払い戻し）アクティビティを任意選択的に実施する。ルーチンはさらに、必要に応じて、定期的にハウスキーピング操作を任意選択的に実施する場合がある。

【 0 0 7 4 】

ブロック 4 8 5 の後、ルーチンは、ブロック 4 9 5 に続き、ルーチンを終了する明示的な指示まで等、継続するかどうかを判定する。継続すると判定された場合、ルーチンは
30
ブロック 4 0 5 に戻り、そうでない場合はブロック 4 9 9 に続いて終了する。ユーザが特定のリクエストされた操作を実施することが認証されているかの判定、いくつかの種類のリクエストされた操作に対してユーザから支払いを即時に取得する等、追加の種類
のアクティビティは、いくつかの実施形態および状況において実施される場合があることが理解される。加えて、ユーザリクエストおよび他の操作は例示された実施形態において、特定のコンピューティングノード群および特定の関連ユーザに特定の様式で実施されるように指示されるが、他の実施形態において、いくつかまたは全てのそのような操作は、代わりに、単一のユーザに関連する複数のコンピューティングノード群へ、および/または 1 つ以上のコンピューティングノード群に関連する複数のユーザから等、より一般的に適用される場合がある。

【 0 0 7 5 】

図 5 は、再帰的容量調和ルーチン 5 0 0 の例示的实施形態のフロー図である。ルーチン
40
は、図 4 のルーチン 4 0 0 のブロック 4 7 0 から開始される場合があるように、例えば、それぞれ図 1 A および 1 B の P E S S M モジュール 1 1 0 および 1 8 0、ならびに/または図 3 の P E S S M モジュール 3 4 0 によって提供される場合がある。

【 0 0 7 6 】

例示された実施形態において、ルーチンはブロック 5 0 5 で開始し、容量調和アクティ
ビティが実施される予定の現在選択されたコンピューティングノード群の指示を取得する。
ブロック 5 1 5 で、ルーチンは次に、選択されたコンピューティングノード群に実施され
た先行の調和アクティビティに基づいて（または、これが実施される最初の調和アクテ
ィビティである場合は、選択されたコンピューティングノード群の初期の公式記録のプ
ログラム実行容量）、および/または P E S 容量保守マネージャモジュール（例えば、図 7
50

に関して詳細に記載のような)によって実施される場合がある、公式記録のプログラム実行容量に対する後続の修正に基づいて等、選択されたコンピューティングノード群の公式記録のプログラム実行容量についての情報を呼び出す。ルーチンは次いでブロック520および525に続き、それぞれ、選択されたコンピューティングノード群の現在の実際のプログラム実行容量および現在所望のプログラム実行容量を判定する。現在の実際のプログラム実行容量の判定は、例えば、選択されたコンピューティングノード群を監視することの一部として、PES容量保守マネージャモジュールによって以前に記憶された情報を呼び出すことを含む場合があるが、他の実施形態において、ルーチン500は、現在の実際の容量を動的に判定する場合がある(例えば、PES容量保守マネージャモジュールまたは他の監視している情報源にその情報を提供するように動的にリクエストすることによ

10

【0077】

ブロック525の現在所望のプログラム実行容量の判定は、例えば、調和アクティビティに対応する現在の総計期間中に選択されたコンピューティングノード群の関連ユーザから受信された任意の動的に指定された容量修正命令に関する情報(例えば、図4のブロック445に関する前述のように)、および現在の総計期間中に満たされていたことが以前に判定されていた選択されたコンピューティングノード群の任意のユーザ指定のトリガに関する情報を呼び出すことを含む場合がある。代替として、いくつかの実施形態において、ルーチンは代わりに、選択されたコンピューティングノード群の任意のユーザ指定のトリガが、現在の総計期間中に現在満たされている、および/または以前に満たされていたかどうかを現在判定するために、選択されたコンピューティングノード群に対して判定された性能特性情報を呼び出す場合がある(例えば、PES容量保守マネージャモジュールによって選択されたコンピューティングノード群を監視することの一部として以前に記憶されていた情報、またはその情報を提供するようにPES容量保守マネージャモジュールまたは他の監視している情報源にリクエストすることによって動的に取得された情報)。多様な情報が呼び出された後、現在所望のプログラム実行容量は、任意のそのような総計された容量修正が行われた後の結果の所望のプログラム実行容量を判定するために、他の部分に詳細を記載するように、1つ以上のリクエストされたプログラム実行容量修正を総計することによって判定される場合がある。

20

【0078】

ブロック525の後、ルーチンはブロック530に続き、現在公式の容量を現在所望の容量に調整するように、選択されたコンピューティングノード群の現在の実際のプログラム実行容量、所望のプログラム実行容量、および公式記録のプログラム実行容量を調和するために、選択されたコンピューティングノード群に行われる実際の変更を判定する。ルーチンは次いで、ブロック535に続き、修正された選択されたコンピューティングノード群によって提供される現在所望のプログラム実行容量を、選択されたコンピューティングノード群の更新された公式記録のプログラム実行容量として指定し、さらに、ブロック530で判定されたプログラム実行容量修正を開始する。ブロック535の後、ルーチンはブロック599に続き、戻る。

30

【0079】

図6は、容量修正帰属ルーチン600の例示的实施形態のフロー図である。ルーチンは、図4のルーチン400のブロック475から開始される場合がある等、例えば、それぞれ図1Aおよび1BのPESMモジュール110および180、ならびに/または図3のPESMモジュール340によって、提供される場合がある。

40

【0080】

例示された実施形態において、ルーチンはブロック605で開始し、容量修正帰属アクティビティが実施される現在選択されたコンピューティングノード群の指示を取得する。ブロック625で、ルーチンは次いで、選択されたコンピューティングノード群に対して実施された先行の調和アクティビティ等、先行の時間以降に発生していたプログラム実行容量の利用可能性の変更についての情報を識別する。変更情報の識別は、例えば、選択さ

50

れたコンピューティングノード群を監視することの一部として、PES容量保守マネージャモジュールによって以前に記憶された情報を呼び出すことを含む場合があるが、他の実施形態において、ルーチン600は、情報を動的に判定する場合がある（例えば、PES容量保守マネージャモジュールまたは他の監視している情報源にその情報を提供するように動的にリクエストすることによって）。

【0081】

ブロック625の後、ルーチンはブロック630に続き、任意のそのような容量の利用可能性の変更が発生していたかを判定し、発生していなかった場合はブロック699に続く。発生していた場合、ルーチンはブロック635に続き、選択されたコンピューティングノード群に対して実施された先行の調和アクティビティ等、先行の時間以降に発生していたプログラム実行容量修正リクエストイベントについての情報を識別する。イベント情報の識別は、例えば、図4および5に関して詳細を記載したように、選択されたコンピューティングノード群の機能を提供する一部として、PESシステムマネージャモジュールによって以前に記憶された情報を呼び出すことを含む場合がある。ブロック640で、ルーチンは次いで、イベントの種類に基づいて等、対応する利用可能性の変更を直接発生する識別されたイベントのうちのいずれかを判定する（例えば、利用不可能なコンピューティングノードを置換する自動化されたシステム操作、即時の対応応答を指定する受信されたユーザ命令等）。

10

【0082】

ブロック640の後、ルーチンはブロック645に続き、先頭から開始して、ブロック625で識別された次の利用可能性の変更を選択する。ブロック650で、ルーチンは次いで、この変更がブロック640で判定された個別のイベントのうちの1つに直接帰属可能であるかを判定し、可能である場合、選択された容量の利用可能性の変更の原因としてそのイベントを記録する。可能でない場合、ブロック650のルーチンは、選択された容量の利用可能性の変更の原因をブロック640で判定されていない他の識別されたイベントの組み合わせとして結論付ける。ブロック660において、ルーチンは次いで、容量の利用可能性の変更がさらに存在するかを判定し、存在する場合、ブロック645に戻り、次のそのような容量の利用可能性の変更を選択する。存在しない場合、ルーチンはブロック699に続いて、戻る。

20

【0083】

図7は、プログラム実行サービス容量保守マネージャルーチン700の例示的实施形態のフロー図である。ルーチンは、コンピューティングノード群から利用可能な実際のプログラム実行容量を判定することを含む、ユーザのためにコンピューティングノードの群の使用を監視する際に支援する等、例えば、それぞれ図1Aおよび1BのPES CMMモジュール115および160、ならびに/または図3のPES CMMモジュール345の実行によって提供される場合がある。この例示された実施形態において、ルーチン700は、複数のユーザの代わりにプログラムを実行するためのプログラム実行容量を提供するプログラム実行サービスと共に操作するが、他の実施形態において、ルーチン700の機能のいくつかまたは全ては、他の様式で提供される場合がある。

30

【0084】

例示された実施形態において、ルーチンはブロック705で開始し、継続的またはそうでなければ反復様式等、1つ以上のコンピューティングノード群のコンピューティングノードについての情報の収集を開始する指示が受信され、一方で本明細書には例示されないが、いくつかの実施形態において、他のモジュールおよび/またはルーチンは、対象の特定のコンピューティングノード群に関して等、対象の特定の情報を生成し、提供するように、ルーチン700に動的にリクエストする場合がある。ブロック705において、ルーチンは、コンピューティングノード群の総計情報を生成する際に後で使用するために、これらのコンピューティングノードが属するコンピューティングノード群の指示を含む、1つ以上のそのようなコンピューティングノードの性能特性情報を収集する。情報は、特定のコンピューティングノードまたは関連モジュール（例えば、仮想マシンコンピューティ

40

50

ングノードの関連VMマネージャ構成要素)に情報を提供するようにリクエストすることによって情報をプルすることによって、そのようなコンピューティングノードおよび/または関連構成要素が情報をルーチン700にプッシュすることによって(例えば、定期的、コンピューティングノードをシャットダウンまたは利用不可能にさせる検出されたエラー状態等の所定の種類のイベント時等)、特定のコンピューティングノードによるネットワークトラフィックおよび/または個別のリソース使用量を監視することによって等を含む、多様な様式で収集される場合がある。

【0085】

ルーチンは次いでブロック710に続き、1つ以上の選択されたコンピューティングノード群のために総計性能特性情報を判定し、全てのコンピューティングノード群、個別のコンピューティングノード情報がちょうど収集されたコンピューティングノード群、総計性能特性情報がリクエストされている、または最近生成されていないコンピューティングノード群、個別のコンピューティングノード情報がコンピューティングノード群のコンピューティングノードの各々に利用可能であるコンピューティングノード群等のために等、その判定された情報を後で使用するために記憶する。その総計性能特性情報は、群のコンピューティングノードに関して部分的情報だけが利用可能である時、利用可能ではない個別の性能特性情報を補間または推定すること等を含め、多様な様式で生成される場合がある。加えて、収集および/または総計される特定の性能特性情報は、所定の実施形態において全てのコンピューティングノード群のための所定の種類の情報を必ず収集する、特定のコンピューティングノード群の判定されたトリガに指定された条件に基づいて所定の種類の情報を収集する等、多様な様式で変動する場合がある。さらに、ブロック705~785は、この例において、順次様式で実施されると例示されるが、多様なブロックは、いくつかの実施形態においてそうではなく他の様式で実施されることが理解される。例えば、いくつかの実施形態において、ブロック705の情報収集アクティビティは、継続ベースまたはほぼ継続ベースで実施される場合があるが、ブロック710および/または他のブロックの総計情報生成は、定期的のみ実施される場合がある。

【0086】

ブロック710の後、ルーチンはブロック715に続き、現在利用可能な群のコンピューティングノードの現在数、および/またはコンピューティングノード群のプログラム実行容量の1つ以上の他の測定値を反映し、後で使用するために判定された情報を記憶する等、1つ以上のコンピューティングノード群の各々のために現在の実際のプログラム実行容量情報を判定し、記憶する。本明細書には例示されないが、いくつかの実施形態および状況において、ルーチンはさらに、コンピューティングノード群のために判定された現在の実際のプログラム実行容量を反映するように、コンピューティングノード群の対応する公式記録のプログラム実行容量を即時に更新する場合があるが、他の実施形態において、公式記録のプログラム実行容量を更新するには、次の対応する群の調和アクティビティまで待機する。

【0087】

ブロック720で、ルーチンは次いで任意選択的に、任意のコンピューティングノード群の満たされていた任意のコンピューティングノード群の任意の容量修正トリガを判定する(例えば、ブロック705で収集され、および/またはブロック710で総計された性能特性情報に基づいて、および/またはブロック715で判定された実際のプログラム実行容量情報に基づいて)が、他の実施形態において、そのようなトリガを満たしたという判定は、そうではなく、図5に関して詳細を説明する対応する調和アクティビティに関して等、総計期間の最後に実施される場合がある。任意のトリガが満たされたと判定された場合、そのような満たされたトリガについての情報は、後で使用するために記憶される。いくつかの実施形態および状況において、特定のトリガを満たすとさらに、関連のコンピューティングノード群の即時プログラム実行容量修正を開始する場合があり、そのような場合、そのようなプログラム実行容量修正動作が開始される場合があり、ルーチンはさらに、図6のルーチン600に関して後で使用するため等、その満たされたトリガをそのブ

10

20

30

40

50

ログラム実行容量修正動作に連結する因果関係情報を記録する場合がある。

【0088】

同様な様式において、ブロック725のルーチンは、任意のコンピューティングノード群のうちの任意のコンピューティングノードが、コンピューティングノード群のために利用可能になるように開始されたが、開始が失敗または指定期間内（例えば、10分）に完了しなかったかどうかを判定する。その場合、ルーチンの例示された実施形態は、任意のそのようなコンピューティングノードのための置換コンピューティングノードの即時提供を開始し、ルーチンはさらに、開始されたコンピューティングノードの利用不可能を開始された置換アクティビティの原因として連結する因果関係情報を記録する場合がある。他の実施形態において、そのような置換アクティビティは、代わりに即時様式で実施されない場合があり（例えば、代わりに、プログラム実行容量を修正する他のリクエストと総計される場合がある）、および/またはそうではなく図4のルーチン400によって実施される場合がある。加えて、そのような利用不可能なコンピューティングノードのための置換アクティビティを開始することの一部として、ルーチンはさらに、利用不可能なコンピューティングノードを終了するよう動作を行う場合がある（例えば、まだ実行中であるが、応答しない場合）。

10

【0089】

類似の様式において、ブロック730のルーチンは、コンピューティングノード群の一部として以前に使用中であったが、現在障害を発生またはそうでなければ利用不可能になっているコンピューティングノード群のうちの任意のコンピューティングノードに置換コンピューティングノードを即時に提供することを任意選択的に開始し、その場合、ルーチンは、開始された置換アクティビティの原因として、そのコンピューティングノードの利用不可能性を連結する因果関係情報をさらに記録する場合がある。他の実施形態において、そのような置換アクティビティは、そうではなく直後の様式において実施されない場合があり（例えば、そうではなく、プログラム実行容量を修正するための他のリクエストと共に総計される場合がある）、および/または図4のルーチン400によって実施される場合があり、および/または特定のコンピューティングノード群のプログラム実行容量に対する即時変更を自動的に開始することができる、他の種類の自動化された判定が実施される場合がある。加えて、利用不可能なコンピューティングノードの置換アクティビティを開始することの一部として、ルーチンはさらに、利用不可能なコンピューティングノードを終了する動作を行う場合がある（例えば、まだ稼動中であるが、応答しない場合）。さらに、他の部分に詳細を記載するように、ルーチンは、いくつかの実施形態において、ブロック725および/または730に関して、関連のユーザが課金される期間、置換アクティビティを実施する望ましさに影響を与える場合がある、次の調和アクティビティの実施が予定されているまでの時間量等、置換アクティビティを即時に実施するかどうかを判定する時に他の因子を検討する場合がある。

20

30

【0090】

ブロック730の後、ルーチンは、ブロック785に続き、必要に応じて記憶された情報を更新することを含む、任意のハウスキーピング操作を任意選択的に実施する。ブロック785の後、ルーチンは、ブロック795に続き、明示的な終了指示が受信されるまで等、継続するかどうかを判定する。継続すると判定された場合、ルーチンはブロック705に戻り、そうでない場合はブロック799に続いて終了する。ルーチン700に関して多様なアクティビティは、この実施形態において、ルーチン400を実施するモジュールとは異なるモジュールによって実施されると例示されるが、他の実施形態において、2つのルーチンの機能のいくつかまたは全ては、単一のモジュールによって実施される、またはそうでなければ組み合わせられる場合がある。

40

【0091】

いくつかの実施形態において、上述のルーチンによって提供される機能は、より多くのルーチンに分割、またはより少ないルーチンに統合等、代替の方式において提供される場合があることが理解される。同様に、いくつかの実施形態において、例示されたルーチン

50

は、他の例示されたルーチンが、その代わりにそれぞれそのような機能を欠落または含む時、または提供される機能の量に変更される時等、記載よりも多いまたは少ない機能を提供する場合がある。加えて、多様な操作は、特定の様式（例えば、順次または並列）において、および/または特定の順序で、実施されると例示される場合がある一方で、当業者は、他の実施形態において、操作は他の順序で、および他の様式において実施される場合があることを理解する。当業者はまた、上述のデータ構造は、単一のデータ構造を複数のデータ構造に分割させることによって、または複数のデータ構造を単一のデータ構造に統合させることによって等、異なる様式において構造化される場合があることも理解する。同様に、いくつかの実施形態において、例示されたデータ構造は、他の例示されたデータ構造が、その代わりにそれぞれそのような情報を欠落または含む時、または記憶される情報の量または種類が変更される時等、記載よりも多いまたは少ない情報を記憶する場合がある。

10

【0092】

第1節。ユーザに提供されるプログラム実行容量における動的な修正の因果関係を判定するための、プログラム実行サービスの構成されたコンピューティングシステムのための方法であって、

プログラム実行サービスの構成されたコンピューティングシステムの制御下であって、ユーザに課金された料金と引き換えに、かつ、多様なユーザの各々について、複数のリモートユーザのプログラムを実行するように構成可能な複数のコンピューティングノードをプログラム実行サービスが提供する、当該制御下において、

20

ユーザの指示されたプログラムの実行で使用するための初期の望ましいコンピューティングノードの数を指定し、かつ、ユーザが使用するために提供されているコンピューティングノードの数に対して自動化された修正を後で開始するために使用するための複数の数修正トリガを指定する、ユーザからの情報を受信することであって、数修正トリガの各々は、数修正トリガが満たされるかどうかを判別するために使用するための1つ以上の基準を含み、かつ、数修正トリガが満たされる場合にリクエストされる指定されたコンピューティングノードの数変更を含む、受信することと、

ユーザの指示されたプログラムを実行するために使用するための多様な複数のコンピューティングノードのユーザのために群を自動的に判別することであって、複数のコンピューティングノードは初期の望ましいコンピューティングノードの数であり、1回目で、ユーザが使用するための複数のコンピューティングノードを提供する、判別することと、

30

ユーザからの命令にตอบสนองして、群のうちのコンピューティングノードの各々において指示されたプログラムのコピーの実行を開始することと、

期間の間、これらのコンピューティングノードの操作から性能測定基準を判定するために、群のうちのコンピューティングノードを自動的に監視することと、ユーザによって指定された数修正トリガのうちの少なくとも1つが、判定された性能測定基準に基づいて満たされることを自動的に判別することと、

期間の間、各々、群のうちのコンピューティングノードの数変更について動的に指定されたリクエストを示す、ユーザからの1つ以上の数変更命令を受信することと、

ユーザのためにプログラムを実行するために、2回目で実際に利用可能な群のうちのコンピューティングノードの数を反映する2回目について実際のコンピューティングノード量を自動的に判別することであって、2回目の判定された実際のコンピューティングノード量は、期間の間の群のうちのコンピューティングノードの利用可能性に対する複数の変更の少なくとも部分的に基づく、1回目の初期の望ましいコンピューティングノードの数とは別であり、複数の利用可能性の変更の少なくともいくつかは、ユーザが課金される1つ以上の関連付けられた料金を有する、判別することと、

40

利用可能性に対する複数の変更のうちの少なくとも1つの各々について、満たされると判定される少なくとも1つの数修正トリガのうちの1つまたはユーザからの1つ以上の数変更命令のうちの1つのいずれかである利用可能性の変更の単一の原因を自動的に結論付けることと、

50

単一の原因が自動的に結論付けられていない複数の利用可能性の変更のうちの少なくとも1つの他の変更の各々について、この少なくとも1つの他の利用可能性の変更は、前記少なくとも1つの利用可能性の変更とは別であり、互いに独立しており、満たされると判定される少なくとも1つの数修正トリガのうちの1つ以上を含み、かつ、ユーザからの1つ以上の数変更命令のうちの少なくとも1つを含む、利用可能性の変更へ、複数の可能性のある原因の組み合わせを自動的に結論付けることと、

少なくともいくつかの利用可能性の変更のための関連付けられた料金が、ユーザに帰責されることを可能にするように、少なくとも1つの利用可能性の変更の各々について結論付けられる単一の原因と、少なくとも1つの他の利用可能性の変更の各々についての複数の原因の結論付けられる組み合わせとの指示を提供することと、によって、1回目から後の2回目への期間、群のうちのコンピューティングノードを自動的に管理することと、を含む、方法。

10

【0093】

第2節。複数のユーザのうちの1人について、2回目で判定された実際のコンピューティングノード数は、満たされると判定される少なくとも1つの数修正トリガのうちの1つ以上と、1人のユーザからの1つ以上の数変更命令のうちの少なくとも1つとの各々から、1人のユーザの群のためのコンピューティングノードの数のリクエストされた増加に部分的に基づいて、1回目について、初期の望ましいコンピューティングノード数よりも多く、1人のユーザの2回目の実際のコンピューティングノード数を自動的に判別することは、1つ以上の満たされた数修正トリガと、1人のユーザからの少なくとも1つの数変更命令との、リクエストされた増加の総計に基づいて、1人のユーザの群のコンピューティングノードの数を増加させるために行う単一の総計された修正を判別することを含み、1人のユーザの群のうちのコンピューティングノード数を増加させるための判定された単一の総計された修正は、1人のユーザの群の少なくとも1つの他の利用可能性の変更のうちの1つであり、1つ以上の満たされた数修正トリガおよび1人のユーザからの少なくとも1つの数変更命令のうちのどれも、1人のユーザの群について、少なくとも1つの利用可能性の変更のいずれの単一の原因としても結論付けられず、1人のユーザの群の少なくとも1つの他の利用可能性の変更の各々についての複数の可能性のある原因の組み合わせは、期間の間、1人のユーザのコンピューティングノード群における利用可能性の変更について識別され、かつ、少なくとも1つの利用可能性の変更のうちの1つの単一の原因として結論付けられない全ての可能性のある原因を含む、第1節に記載の方法。

20

30

【0094】

第3節。プログラム実行サービスは、リモートユーザの各々が、1つ以上のネットワークにおいてクライアントコンピューティングデバイスからの命令を提供するように、ネットワークでアクセス可能なサービスであり、1人のユーザの群の関連付けられた料金による少なくともいくつかの利用可能性の変更は、1人のユーザの群のうちのコンピューティングノードの数における1つ以上の増加を含み、1人のユーザは、1回目で1人のユーザが使用するための複数のコンピューティングノードを提供するための第1の料金を支払い、1人のユーザの群の1つ以上の数の増加の一部として提供された各追加のコンピューティングノードの1つ以上の個別の第2の料金を支払う、第2節に記載の方法。

40

【0095】

第4節。ユーザに提供されたプログラム実行容量における動的修正の因果関係を判別するためのコンピュータで実装可能な方法であって、方法は、

複数のユーザが使用するためのプログラム実行サービスを提供するように構成される1つ以上のコンピューティングシステムの制御下であって、プログラム実行サービスのユーザのプログラムを実行するように構成可能な複数のコンピューティングノードをプログラム実行サービスが有する、当該制御下において、

プログラム実行サービスの第1のユーザの代わりに、1つ以上のソフトウェアプログラムを実行するための初期の望ましいプログラム実行容量の指示を受信することと、

第1のユーザに提供されているプログラム実行容量に対する自動化された修正を後で開

50

始するために使用するための複数の容量修正トリガを判別することであって、容量修正トリガの各々は、容量修正トリガが満たされるかどうかを判別するために使用するための1つ以上の基準を含み、かつ、容量修正トリガが満たされる場合にリクエストされる指定された種類のプログラム実行容量修正を含む、判別することと、

第1のユーザの初期の望ましいプログラム実行容量を提供するために使用するための多様な複数のコンピューティングノードの複数の第1の群を自動的に判別することと、1回目で利用可能な第1の群のうちのコンピューティングノードに、第1のユーザのために1つ以上のソフトウェアプログラムを各々実行させることと、

1回目と後の2回目との間の期間において生じ、かつ、各々、第1の群の複数のコンピューティングノードによって提供されているプログラム実行容量の修正を生じさせることができる、複数の独立したイベントを自動的に識別することであって、複数のイベントは、満たされると判定される指定された容量修正トリガのうちの少なくとも1つを含む、識別することと、

10

2回目に第1の群のうちのコンピューティングノードから第1のユーザへ利用可能な実際のプログラム実行容量を自動的に判別することであって、2回目の実際のプログラム実行容量は、1回目と2回目との間の期間に生じた第1の群の1つ以上のコンピューティングノードの利用可能性に対する1つ以上の変更に少なくとも部分的に基づいて、1回目に第1のユーザに提供された望ましいプログラム実行容量とは別である、判別することと、

第1の群の1つ以上のコンピューティングノードの利用可能性の変更の各々について、利用可能性の変更の原因として識別されたイベントのうちの1つ以上を自動的に結論付けることであって、利用可能性の変更のうちの少なくとも1つは、互いに独立しており、かつ、満たされると判定される少なくとも1つの指定された容量修正トリガのうちの1つ以上を含む複数のイベントの組み合わせである原因を有する、結論付けることと、

20

1つ以上の追加の動作が、その結論付けられる原因に対応して実施されることを可能にするように、利用可能性の変更のうちの1つ以上の各々について、結論付けられる原因の指示を提供することと、を含む、方法。

【0096】

第5節。複数のイベントは、第1の群によって提供されているプログラム実行容量に対する指定された修正をリクエストするために、第1のユーザによって動的に指定される少なくとも1つの容量修正命令をさらに含み、少なくとも1つの利用可能性の変更のうちの1つの原因として結論付けられる複数のイベントの組み合わせは、少なくとも1つの指定された容量修正命令のうちの1つ以上を含む、第4節に記載の方法。

30

【0097】

第6節。利用可能性の変更のうちの少なくとも1つの他の変更は、複数のイベントのうちの単一の1つである結論付けられる原因を有し、この少なくとも1つの他の利用可能性の変更は、前記少なくとも1つの利用可能性の変更とは別である、第4節に記載の方法。

【0098】

第7節。単一の1つのイベントは、少なくとも1つの他の利用可能性の変更の原因の判定された種類であり、判定された原因の種類は、直接的原因または間接的原因のいずれかであり、単一の1つのイベントの指示を提供することは、少なくとも1つの他の利用可能性の変更の原因の判定された種類であることを自動的に判別することをさらに含む、第6節に記載の方法。

40

【0099】

第8節。実際のプログラム実行容量を自動的に判別することは、生じた複数の利用可能性の変更を判別することを含み、利用可能性の変更の各々の原因として1つ以上のイベントを自動的に結論付けることは、各々、複数の利用可能性の変更のうちの1つの単一の原因である複数のイベントのうちの1つ以上を識別することと、単一の原因を有しない複数の利用可能性の変更の互いの組み合わせの一部として、複数の利用可能性の変更のうちの1つの単一の原因ではない複数のイベントの全てを総計することと、を含む、第4節に記載の方法。

50

【 0 1 0 0 】

第9節。少なくとも1つの利用可能性の変更のうちの1つについて、1人の利用可能性の変更の原因としての複数のイベントの組み合わせを自動的に結論付けることは、複数のイベントの組み合わせが1つの利用可能性の変更の原因の判定された種類であることを判別することを含み、判定された種類の原因は、直接的原因または間接的原因のいずれかであり、1つ以上の利用可能性の変更の各々について結論付けられる原因の指示を提供することは、1つの利用可能性の変更の原因の判定された種類である複数のイベントの組み合わせの指示を提供することを含む、第4節に記載の方法。

【 0 1 0 1 】

第10節。複数の独立したイベントは、各々、第1の群によって提供されているプログラム実行容量の修正をリクエストし、方法は、複数のイベントのリクエストされた修正の総計に基づいて、第1の群のプログラム実行容量に対して行う単一の総計された修正を自動的に判別することと、2回目の前に第1の群に対する判定された単一の総計された修正を開始することと、をさらに含む、第4節に記載の方法。

10

【 0 1 0 2 】

第11節。複数のイベントは、第1の群によって提供されているプログラム実行容量に対する指定された修正をリクエストするために、第1のユーザによって動的に指定される少なくとも1つの容量修正命令をさらに含み、判定された複数の容量修正トリガは、1回目の前に第1のユーザによって指定され、少なくとも1つの利用可能性の変更のうちの1つの原因として結論付けられる複数のイベントの組み合わせは、少なくとも1つの指定された容量修正命令のうちの1つ以上を含む、第10節に記載の方法。

20

【 0 1 0 3 】

第12節。判定された単一の総計された修正は、1つ以上の指定された容量修正命令が原因として結論付けられる組み合わせに含まれる1つの利用可能性の変更である、第11節に記載の方法。

【 0 1 0 4 】

第13節。第1の群のプログラム実行容量は、初期の望ましいプログラム実行容量が初期の望ましいコンピューティングノードの数となり、かつ、2回目で判定された実際のプログラム実行容量が2回目で第1の群の一部として利用可能な実際のコンピューティングノードの数となるように、第1の群の一部であるコンピューティングノードの数に基づいて計測され、第1の群の1つ以上のコンピューティングノードの1つ以上の利用可能性の変更は、各々、利用不可能になる第1の群と、第1の群に追加される1つ以上の新しいコンピューティングノードとの一部である1つ以上のコンピューティングノードのうちの少なくとも1つを含み、第1の群に対する判定された単一の総計された修正は、第1の群のうちのコンピューティングノードの数の判定された変更である、第10節に記載の方法。

30

【 0 1 0 5 】

第14節。判定された単一の総計された修正は、1つ以上のコンピューティングノードの指定された数だけ、第1の群のうちのコンピューティングノードの数を増加させることであり、判定された単一の総計された修正は、結論付けられる原因の指示が提供される1つ以上の利用可能性の変更のうちの1つであり、方法は、開始された判定された単一の総計された修正に少なくとも部分的に基づいて、第1のユーザに対して1つ以上の金銭的な課金を開始することを含む1つ以上の追加の動作を実行することをさらに含む、第13節に記載の方法。

40

【 0 1 0 6 】

第15節。結論付けられる原因の指示が提供される1つ以上の利用可能性の変更のうちの1つは、1つの利用可能性の変更の原因の少なくとも一部として、複数のイベントの少なくとも最初を含み、1人の利用可能性の変更の結論付けられる原因の指示を提供することは、イベントが1つの利用可能性の変更を生じさせる、および第1のイベントによって利用可能性の変更が生じる、のうちの少なくとも1つのリクエストに応じて実行される、第4節に記載の方法。

50

【 0 1 0 7 】

第 1 6 節。ユーザからリクエストが受信され、1つの利用可能性の変更の結論付けられる原因の指示を提供することは、1人の利用可能性の変更の結論付けられた原因の人間可読説明を生成することと、ユーザに表示するための生成された人間可読説明を提供することを含む、第 1 5 節に記載の方法。

【 0 1 0 8 】

第 1 7 節。2回目後に生じる1つ以上の追加の期間の各々について、追加の期間の間に生じた利用可能性の変更の原因として1つ以上の追加の識別されたイベントを自動的に結論付けることをさらに含み、1つ以上の追加の期間のうちの1つの間に生じた利用可能性の変更のうちの1つの原因である追加の識別されたイベントのうちの1つについて、1つの追加の識別されたイベントは、1回目と2回目との間の期間の間に生じた利用可能性の変更のうちの1つの結果である、第 4 節に記載の方法。

10

【 0 1 0 9 】

第 1 8 節。第 1 のユーザのために1つ以上のソフトウェアプログラムを各々実行するために、1回目に第 1 の群のうちのコンピューティングノードを利用可能にすることは、第 1 の群のうちのコンピューティングノードの各々を自動的に供給することを含み、かつ、第 1 の群のうちのコンピューティングノードの各々上で1つ以上のソフトウェアプログラムの各々の実行を自動的に開始することを含む、第 4 節に記載の方法。

【 0 1 1 0 】

第 1 9 節。プログラム実行サービスは、第 1 のユーザからリモートである料金ベースのネットワークでアクセス可能なサービスであり、プログラム実行サービスは、1つ以上のネットワークにおいてプログラム実行サービスとプログラムの相互作用するためのリモートコンピューティングシステムの1つ以上のアプリケーションプログラミングインターフェース (API) と、1つ以上のネットワークにおいてリモートユーザが使用するためのグラフィカルユーザインターフェースとの少なくとも1つを提供し、第 1 のユーザは、利用可能性の変更のうちの1つ以上の結論付けられる原因に少なくとも一部基づいて、1つ以上の料金を支払う、第 4 節に記載の方法。

20

【 0 1 1 1 】

第 2 0 節。複数のコンピューティングノードが、複数の物理的コンピューティングシステムの各々について、各々少なくとも1つのプログラムを実行することができる物理的コンピューティングシステムによってホストされる複数の仮想マシンを含むように、プログラム実行サービスは、仮想技法を使用し、第 1 の群のうちのコンピューティングノードはホストされた仮想マシンであり、第 1 のユーザの1つ以上のプログラムは仮想マシンイメージの一部である、第 4 節に記載の方法。

30

【 0 1 1 2 】

第 2 1 節。

構成されたコンピューティングシステムの制御下において、第 1 のユーザの代わりに1つ以上のソフトウェアプログラムを実行するために使用するための初期のプログラム実行容量を提供するために、1回目で第 1 のユーザに関連付けられる複数のコンピューティングノードの第 1 の群の指示を受信することと、

40

1回目と後の2回目との間に生じ、かつ、各々、第 1 の群のプログラム実行容量の関連付けられた望ましい修正を有する複数の独立したイベントを自動的に識別することと、

2回目に第 1 の群のうちのコンピューティングノードから第 1 のユーザへ利用可能な実際のプログラム実行容量を自動的に判別することであって、2回目の実際のプログラム実行容量は、第 1 の群の1つ以上のコンピューティングノードの利用可能性に対する1つ以上の変更少なくとも部分的に基づく、1回目で提供された初期のプログラム実行容量とは、別である、判別することと、

第 1 の群の判定された利用可能性の変更の各々に1つ以上の原因を自動的に結論付けることであって、利用可能性の変更の少なくとも1つに結論付けられる1つ以上の原因は、多様な識別された独立したイベントの組み合わせである、結論付けることと、

50

少なくとも1つの利用可能性の変更のうちの1つ以上について、結論付けられる1つ以上の原因の指示を提供することと、を含む方法を実行することによって、そのコンテンツが、ユーザに提供されたプログラム実行容量の動的修正のために因果関係を判定するように、コンピューティングシステムを構成する、コンピュータ可読媒体。

【0113】

第22節。構成されたコンピューティングシステムは、プログラム実行サービスのユーザのプログラムを実行するように構成可能な複数のコンピューティングノードを提供するプログラム実行サービスの一部であり、第1の群の複数のコンピューティングノードは、複数のコンピューティングノードのサブセットであり、第1の群の指示を受信することは、1回目の前に第1のユーザから初期のプログラム実行容量のリクエストを受信することと、第1のユーザに初期のプログラム実行容量を提供するために使用するための1回目で第1のユーザに、コンピューティングノードの第1の群を自動的に関連付けることと、を含み、第1のユーザは、第1の群のプログラム実行容量に対する自動化された修正を後で開始するために使用するための、1回目の前の1つ以上の容量修正トリガをさらに示し、複数の独立したイベントは、満たされると判定され、かつ、第1のユーザによって動的に指定される少なくとも1つの容量修正命令を含む、容量修正トリガのうちの少なくとも1つを含む、第21節に記載のコンピュータ可読媒体。

10

【0114】

第23節。複数の独立したイベントの各々についてのプログラム実行容量の望ましい修正はリクエストされた修正であり、方法は、少なくとも1つの満たされた容量修正トリガと、第1のユーザからの少なくとも1つの容量修正命令とのリクエストされた修正の総計に基づいて、第1の群のプログラム実行容量を増加させるために行う単一の総計された修正を自動的に判別することと、2回目の前に判定された単一の総計された修正を開始することとをさらに含み、判定された単一の総計された修正は、少なくとも1つの利用可能性の変更のうちの1つであり、判定された単一の総計された修正の1つ以上の原因として結論付けられる複数の独立したイベントの組み合わせは、少なくとも1つの満たされた容量修正トリガおよび第1のユーザからの少なくとも1つの容量修正命令を含む、第22節に記載のコンピュータ可読媒体。

20

【0115】

第24節。コンピュータ可読媒体は、コンテンツを記憶するコンピューティングシステムのメモリと、コンテンツを含有する生成された記憶されたデータ信号を含むデータ伝送媒体とのうちの少なくとも1つであり、コンテンツは、実行されるとコンピューティングシステムに方法を実行させる命令である、第21節に記載のコンピュータ可読媒体。

30

【0116】

第25節。ユーザのために、プログラム実行容量における動的修正のために因果関係を判定するように構成されたコンピューティングシステムであって、

1つ以上のプロセッサと、
複数のユーザの各々について、

ユーザの代わりに、1つ以上のソフトウェアプログラムを実行するために使用するために提供される初期のプログラム実行容量を判別することと、ユーザのために提供されているプログラム実行容量に対する自動化された修正を後で開始するために使用するための1つ以上の容量修正トリガを判別することと、

40

ユーザに初期の望ましいプログラム実行容量を提供するために使用するために、複数の利用可能なコンピューティングノードの群を1回目にユーザに自動的に関連付けることと、

1回目した後、1回目と後の2回目との間で生じ、かつ、群のプログラム実行容量の関連付けられた望ましい修正を各々有する、複数の独立したイベントを自動的に識別することとであって、複数のイベントは満たされると判定される容量修正トリガのうちの少なくとも1つを含む、識別することと、

群によって提供されるプログラム実行容量に影響する群の1つ以上のコンピューティン

50

グノードの利用可能性に対する1つ以上の変更を自動的に判別することと、

群の判定された利用可能性の変更の各々に対して、1つ以上の原因を自動的に結論付けることであって、利用可能性の変更の少なくとも1つに結論付けられる1つ以上の原因は、多様な識別された独立したイベントの組み合わせであり、利用可能性の変更のうちの少なくとも1つの他の変更に結論付けられる1つ以上の原因は、識別された独立したイベントのうちの単一の1つであり、群の判定された利用可能性の変更のうちの1つ以上について結論付けられる1つ以上の原因の指示を提供することと、によって、

1つ以上のプロセッサのうちの少なくとも1つによって実行されると、ネットワークでアクセス可能なサービスの複数のユーザのためにプログラム実行容量を管理するように構成されるシステムマネージャモジュールと、を備える、コンピューティングシステム。

10

【0117】

第26節。複数のユーザのうちの1人のための複数の独立したイベントは、1人のユーザによって動的に指定される少なくとも1つの容量修正命令をさらに含み、システムマネージャモジュールは、少なくとも1つの満たされた容量修正トリガと、1人のユーザによって指定された少なくとも1つの容量修正命令との望ましい修正の総計に基づいて、1人のユーザの群のプログラム実行容量を増加させるために行う単一の総計された修正を自動的に判定し、かつ、2回目の前に1人のユーザのために群に対して判定された単一の総計された修正を開始するように構成され、判定された単一の総計された修正は、1人のユーザの少なくとも1つの利用可能性の変更のうちの1つであり、判定された単一の総計された修正の1つ以上の原因として結論付けられる複数の独立したイベントの組み合わせは、

20

少なくとも1つの満たされた容量修正トリガおよび1人のユーザによって指定された少なくとも1つの容量修正命令を含む、第25節に記載のコンピューティングシステム。

【0118】

第27節。1人のユーザのための判定された容量修正トリガは1人のユーザによって指定され、ネットワークでアクセス可能なサービスは、プログラム実行サービスのリモートユーザのプログラムを実行するように構成可能な複数のコンピューティングノードを提供するプログラム実行サービスであり、複数のユーザの少なくとも1人の各々についての群の複数のコンピューティングノードは、複数のコンピューティングノードのサブセットである、第26節に記載のコンピューティングシステム。

【0119】

第28節。システムマネージャコンポーネントは、コンピューティングシステムによって実行するためのソフトウェア命令を含む、第25節に記載のコンピューティングシステム。

30

【0120】

第29節。複数のユーザの各々について、

ユーザの代わりに1つ以上のソフトウェアプログラムを実行するために使用するために提供される初期のプログラム実行容量を判別することと、ユーザに提供されているプログラム実行容量に対する自動化された修正を後で開始するために使用するための1つ以上の容量修正トリガを判別することと、

ユーザへ初期の望ましいプログラム実行容量を提供するために使用するための、1回目でユーザに、複数の利用可能なコンピューティングノードの群を自動的に関連付けることと、

40

1回目した後、1回目と後の2回目との間に生じ、かつ、群のプログラム実行容量の関連付けられた望ましい修正を各々有する複数の独立したイベントを自動的に識別することであって、複数のイベントは、満たされると判定される容量修正トリガの少なくとも1つを含む、識別することと、

群によって提供されたプログラム実行容量に影響する群の1つ以上のコンピューティングノードの利用可能性に対する1つ以上の変更を自動的に判別することと、

群の判定された利用可能性の変更の各々に1つ以上の原因を自動的に結論付けることであって、利用可能性の変更の少なくとも1つに結論付けられる1つ以上の原因は、多様な

50

識別された独立したイベントの組み合わせであり、利用可能性の変更のうちの少なくとも1つの他の変更結論付けられる1つ以上の原因は、単一の識別された独立したイベントのうちの1つであり、群の判定された利用可能性の変更のうちの1つ以上について、結論付けられる1つ以上の原因に指示を提供すること、によって、システムマネージャコンポーネントは、ネットワークでアクセス可能なサービスの複数のユーザのプログラム実行容量を管理するための手段で構成される、第25節に記載のコンピューティングシステム。

【0121】

第30節。ユーザのプログラム実行容量を動的に修正するための、プログラム実行サービスの構成されたコンピューティングシステムの方法であって、方法は、

プログラムの実行サービスの構成されたコンピューティングシステムの制御下であって、複数のリモートユーザのプログラムを実行するように構成可能な複数のコンピューティングノードをプログラム実行サービスが提供する、当該制御下において、多様なユーザの各々について、

ユーザの指示されたプログラムを実行するために使用するための初期の望ましいコンピューティングノードの数を指定し、かつ、望ましいコンピューティングノード数に対する自動化された修正を後で開始するために使用するための1つ以上の数修正トリガを指定する、ユーザからの情報を受信することであって、数修正トリガの各々は、数修正トリガが満たされているかどうかを判別するために使用するための1つ以上の基準を含み、かつ、数修正トリガが満たされる場合に指定されたコンピューティングノード数変更を含む、受信することと、

ユーザの指示されたプログラムを実行するために使用するための複数のコンピューティングノードの複数の群を自動的に判別することであって、群の複数のコンピューティングノードは初期の望ましいコンピューティングノード量である、判別することと、1回目を使用するための複数のコンピューティングノードユーザを提供することと、

ユーザからの1つ以上の命令にตอบสนองして、群のうちのコンピューティングノードの各々の上に指示されたプログラムのコピーの実行を開始することと、1回目にユーザに利用可能であると記録される公的コンピューティングノード量として初期の望ましいコンピューティングノード量を指定することと、

期間の間、これらのコンピューティングノードの操作から性能特性を判定するために、群のうちのコンピューティングノードを自動的に監視することと、ユーザによって指定された数修正トリガのうちの少なくとも1つが判定された性能特性に基づいて満たされることを自動的に判別することと、

期間の間、ユーザによって動的に指定される群のうちのコンピューティングノードのユーザ指示されたコンピューティングノード量の変更を示す少なくとも1つの命令を受信することと、

2回目に修正された望ましいコンピューティングノード量を自動的に判別することであって、修正された望ましいコンピューティングノード数は、少なくとも1つの満たされた数修正トリガからの指定されたコンピューティングノード量の変更と、ユーザ指示されたコンピューティングノード数の変更とに少なくとも部分的に基づいて、1回目からの望ましいコンピューティングノード数に対する変更を含む、判別することと、

ユーザについてプログラムを実行するために、2回目に実際に利用可能な群のうちのコンピューティングノードの数を反映する。2回目で実際のコンピューティングノード量を自動的に判別することと、

2回目に、ユーザの公的コンピューティングノード数を修正された望ましいコンピューティングノード数に自動的に調和させることであって、調和させることは、修正された望ましいコンピューティングノード数になるように判定された実際のコンピューティングノード数を変更するための動作を行うことによって、群の一部であるコンピューティングノードの数を自動的に修正することを含む、調和させることと、

ユーザが使用するための群の修正されたコンピューティングノードの数を提供することと、2回目にユーザに利用可能な公的コンピューティングノード数になるように修正され

10

20

30

40

50

た望ましいコンピューティングノード量を指定することと、によって、1回目から後の2回目への期間、群のうちのコンピューティングノードを自動的に管理することと、を含む、方法。

【0122】

第31節。複数のユーザのうちの1人について、2回目に判定された実際のコンピューティングノード数は、1回目から2回目への期間の間に使用することが利用不可能になる1人のユーザのための群のうちのコンピューティングノードのうちの1つ以上に少なくとも部分的に基づいて、1回目の公的コンピューティングノード数未満であり、期間の間、1人のユーザについて群のうちのコンピューティングノードを自動的に監視することは、1人のユーザの現行の実際のコンピューティングノード数を繰り返し判断することと、判定された現行の実際のコンピューティングノード量で、更新された公的コンピューティングノード数を調整するために、1人のユーザの公的コンピューティングノード数を更新することとを含み、1人のユーザの少なくとも1つの満たされた数修正トリガは、1人のユーザによって指定された初期の望ましいコンピューティングノードの数に、2回目の公的コンピューティングノード量を維持するために、1つ以上の利用不可能なコンピューティングノードの数だけ、1人のユーザの群のうちのコンピューティングノードを増加させるように指定するトリガであり、1人のユーザによって動的に指定されるユーザ指示された量の変更は、少なくとも1つのコンピューティングノードの初期の望ましいコンピューティングノードの数の指示された減少を含み、2回目に1人のユーザの修正された望ましいコンピューティングノード数を自動的に判断することは、1つ以上の利用不可能なコンピューティングノードで、指示された減少を少なくとも部分的に相殺することを含み、2回目に1人のユーザに利用可能な公的コンピューティングノード数は、1人のユーザによって指定された初期の望ましいコンピューティングノードの数から、1人のユーザによって指定される指示された減少を引いたものである、第30節に記載の方法。

10

20

【0123】

第32節。プログラム実行サービスは、複数のユーザが各々、1回目にユーザが使用するための複数のコンピューティングノードを提供するための第1の料金を支払い、かつ、各々、2回目でユーザが使用するための群の修正されたコンピューティングノードの数を提供するための個別の第2の料金を支払うように、料金ベースのネットワークでアクセス可能なサービスであり、第1および第2の料金は、各々、ユーザの群のうちのコンピューティングノードの現行の数に少なくとも部分的に基づく、第31節に記載の方法。

30

【0124】

第33節。ユーザのためにプログラム実行容量を動的に修正するためのコンピュータで実装可能な方法であり、方法は、

複数のユーザが使用するためのプログラム実行サービスを提供するように構成される1つ以上のコンピューティングシステムの制御下であって、プログラム実行サービスのユーザのプログラムを実行するように構成可能な複数のコンピューティングノードをプログラム実行サービスが有する、当該制御下において、

プログラム実行サービスの第1のユーザの代わりに1つ以上のソフトウェアプログラムを実行するために使用するための初期の望ましいプログラム実行容量を指定し、かつ、望ましいプログラム実行容量に対する自動化された修正を後で開始するために使用するための1つ以上の容量修正トリガを指定する情報を受信することであって、容量修正トリガの各々は、容量修正トリガが満たされるかどうかを判断するために使用するための1つ以上の基準を含み、かつ、容量修正トリガが満たされる場合にリクエストされる望ましいプログラム実行容量に対する指定された種類の修正を含む、受信することと、

40

第1のユーザの初期の望ましいプログラム実行容量を提供するために使用するための多様な複数のコンピューティングノード第1の群を自動的に判断することと、各々、第1のユーザに1つ以上のソフトウェアプログラムを実行するために、1回目で第1の群のうちのコンピューティングノードを利用可能にすることと、1回目に、第1のユーザに利用可能であるものとして記録される公的プログラム実行容量として初期の望ましいプログラム

50

実行容量を指定することと、

1 回目の後、1 つ以上のソフトウェアプログラムを実行することに関連してこれらのコンピューティングノードの性能特性の1 つ以上のタイプの現行の値を判定するように、第1の群のうちのコンピューティングノードを監視することであって、監視することは、1 回目の後、後の2 回目において第1の群のうちのコンピューティングノードから第1のユーザへ利用可能な実際のプログラム実行容量を判別することを含む、監視することと、

1 回目と2 回目との間の期間において生じる複数のイベントに基づいて、第1のユーザの望ましいプログラム実行容量を修正するために自動的に判別することであって、複数のイベントは、1 つ以上の性能特性タイプの判定された現行の値に少なくとも部分的に基づいて満たされると判定される指定された容量修正トリガのうちの少なくとも1 つを含み、かつ、第1のユーザによって動的に指定される少なくとも1 つの容量修正命令を含み、少なくとも1 つの指定された容量修正トリガおよび少なくとも1 つの容量修正命令の各々は、望ましいプログラム実行容量に対する指定された修正を示す、判別することと、

2 回目に判定された実際のプログラム実行容量および修正された望ましいプログラム実行容量によって、第1のユーザに利用可能な記録された公的プログラム実行容量を自動的に調和させることであって、記録された公的プログラム実行容量を調和させることは、修正されたコンピューティングノードが修正された望ましいプログラム実行容量を提供するために利用可能になるように、かつ、記録された公的プログラム実行容量が提供された修正された望ましいプログラム実行容量を反映するように更新されるように、判定された実際のプログラム実行容量および修正された望ましいプログラム実行容量に従って、第1の群の一部であるコンピューティングノードを自動的に修正することを含む、調和させることと、を含む、方法。

【 0 1 2 5 】

第3 4 節。初期の望ましいプログラム実行容量が初期の望ましいコンピューティングノードの数になるように、第1の群のプログラム実行容量は、第1の群の一部であるコンピューティングノードの数に基づいて計測され、2 回目に判定された実際のプログラム実行容量は、2 回目に第1の群の一部として利用可能な実際のコンピューティングノードの数であり、望ましいプログラム実行容量を修正するために自動的に判別することは、第1の群のうちのコンピューティングノードの修正された望ましい数として、修正された望ましいプログラム実行容量を判別することを含み、第1の群の一部であるコンピューティングノードを自動的に修正することは、修正された望ましい量になるように、第1の群の一部であるコンピューティングノードの実際の数を変更させることを含む、第3 3 節に記載の方法。

【 0 1 2 6 】

第3 5 節。少なくとも1 つの指定された容量修正トリガおよび少なくとも1 つの容量修正命令の各々の望ましいプログラム実行容量に対する指定された修正は、第1の群のうちのコンピューティングノードの望ましい数における指示された数の変更を反映し、第1の群についての修正された望ましいプログラム実行容量を判別することは、少なくとも1 つの指定された容量修正トリガおよび少なくとも1 つの容量修正命令のうちの指示された数の変更を総計することと、総計された量の変更によってコンピューティングノードの初期の望ましい数を修正することと、を含む、第3 4 節に記載の方法。

【 0 1 2 7 】

第3 6 節。監視することによって現行の値が判定される、第1の群のうちのコンピューティングノードの性能特性の1 つ以上のタイプは、第1の群の一部として利用可能なコンピューティングノードの実際の数を含み、監視することは、1 回目と2 回目との間の期間、複数回の各々においてコンピューティングノードの現行の実際の数を判別することを含み、記録された公的プログラム実行容量を調和させることは、2 回目に第1のユーザに利用可能な現行の公的プログラム実行容量へ、第1の群のうちのコンピューティングノードの修正された望ましい量を指定することをさらに含む、第3 4 節に記載の方法。

【 0 1 2 8 】

10

20

30

40

50

第37節。第1の群のプログラム実行容量は、第1の群の一部であるコンピューティングノードから利用可能なコンピューティングリソースの複数のタイプの各々を総計することに基づいて計測され、第1の群の一部であるコンピューティングノードを自動的に修正することは、コンピューティングノードを自動的に修正後に第1の群の一部であるコンピューティングノードから利用可能なコンピューティングリソースの複数のタイプのうちの少なくとも1つの総計量を変更させることを含む、第33節に記載の方法。

【0129】

第38節。少なくとも1つの指定された容量修正トリガのうちの各々の1つ以上の基準は、1つ以上の性能特性タイプのうちの少なくとも1つについて1つ以上の指定された値に基づき、方法は、少なくとも1つの指定された容量修正トリガの各々についての1つ以上の指定された値と、1つ以上の性能特性タイプについて判定された現行の値との間の一致に少なくとも部分的に基づいて、少なくとも1つの指定された容量修正トリガが満たされることを自動的に判別することをさらに含む、第33節に記載の方法。

10

【0130】

第39節。少なくとも1つの性能特性タイプは、第1の群のうちのコンピューティングノード上のコンピューティング負荷に対応し、少なくとも1つの指定された容量修正トリガが満たされていることを判別することは、一時的に増加した状態である第1の群のうちのコンピューティングノード上のコンピューティング負荷に基づき、第1の群の一部であるコンピューティングノードを自動的に修正することは、コンピューティング負荷の一時的に増加した状態を受け入れるために、第1の群のうちのコンピューティングノードから利用可能な実際のプログラム実行容量を一時的に増加させることを含み、方法は、コンピューティング負荷の一時的に増加した状態の終了を反映するために、第1の群のうちのコンピューティングノードの一時的に増加した実際のプログラム実行容量を減少させるために、実際のプログラム実行容量を一時的に増加した後で、後の3回目において自動的に判別することをさらに含む、第38節に記載の方法。

20

【0131】

第40節。少なくとも1つの性能特性タイプは、第1の群の一部として利用可能なコンピューティングノードの実際の数を含み、少なくとも1つの指定された容量修正トリガが満たされることを判別することは、指定された望ましいコンピューティングノードの数とは異なっている第1の群のうちのコンピューティングノードの実際の数に基づき、第1の群の一部であるコンピューティングノードを自動的に修正することは、指定された望ましいコンピューティングノードの数に、第1の群のうちのコンピューティングノードの異なる実際の数を戻すことを含む、第38節に記載の方法。

30

【0132】

第41節。少なくとも1つの指定された容量修正トリガのうちの1つの1つ以上の基準は、指示された性能特性タイプの判定された現行の値の閾値と、指示された性能特性タイプの判定された現行の値の経時的变化量と、および複数の性能特性タイプの判定された現行の値の論理的な組み合わせと、の1つ以上に基づく、第38節に記載の方法。

【0133】

第42節。複数のイベントは、その1つ以上の基準が1つ以上の指定された時間に基づく、トリガされた指定された容量修正トリガのうちの1つをさらに含み、方法は、1つ以上の指定された時間に一致する現時刻に少なくとも部分的に基づいて、1つの指定された容量修正トリガが満たされると自動的に判別することをさらに含む、第33節に記載の方法。

40

【0134】

第43節。複数のイベントは、その1つ以上の基準が履歴データから識別された1つ以上のパターンに基づく、トリガされた指定された容量修正トリガのうちの1つをさらに含み、方法は、1つ以上のパターンのうちの少なくとも1つに対応する1つ以上の性能特性タイプについて、少なくとも部分的に判定された現行の値に基づいて、1つの指定された容量修正トリガが満たされることを自動的に判別することをさらに含む、第33節に記

50

載の方法。

【 0 1 3 5 】

第 4 4 節。複数のイベントは、その 1 つ以上の基準が、第 1 の群のうちのコンピューティングノード上で実行される 1 つ以上のソフトウェアプログラムによって実行される仕事量の 1 つ以上の指示に基づく、トリガされている指定された容量修正トリガのうちの 1 つをさらに含み、方法は、1 つ以上のソフトウェアプログラムによって実行される実際の仕事量に関連する取得された情報に少なくとも部分的に基づいて、1 つの指定された容量修正トリガが満たされることを自動的に判別することをさらに含む、第 3 3 節に記載の方法。

【 0 1 3 6 】

第 4 5 節。第 1 の群について修正された望ましいプログラム実行容量を判別することは、少なくとも 1 つの指定された容量修正トリガおよび少なくとも 1 つの容量修正命令の各々によって指定される望ましいプログラム実行容量に対する指示された修正を総計することと、望ましいプログラム実行容量に対する総計された指示された修正に基づいて、判定された修正された望ましいプログラム実行容量を指定することと、を含む、第 3 3 節に記載の方法。

【 0 1 3 7 】

第 4 6 節。望ましいプログラム実行容量に対する指示された修正を総計することは、指示された修正を累積することと、指示された修正のうちの少なくとも 1 つを、指示された修正のうちの少なくとも 1 つの他の修正で相殺することと、1 つ以上の選択基準に基づいて指示された修正のうちの単一の 1 つを選択することと、指示された修正のうちの 1 つ以上に関連付けられる 1 つ以上の優先を使用することと、のうちの少なくとも 1 つを含む、第 4 5 節に記載の方法。

【 0 1 3 8 】

第 4 7 節。第 1 のユーザによって動的に指定され、かつ、複数のイベントに含まれる少なくとも 1 つの容量修正命令は、第 1 のユーザによって指定された単一の容量修正命令を含み、第 1 のユーザによって指定される容量修正命令は、満たされると判定される指定された容量修正トリガよりも優先され、第 1 の群の修正された望ましいプログラム実行容量を判別することは、少なくとも 1 つの指定された容量修正トリガの指示された修正の代わりに使用される単一の容量修正命令によって指定される望ましいプログラム実行容量に対する指示された修正の選択を含む、第 3 3 節に記載の方法。

【 0 1 3 9 】

第 4 8 節。その地理的位置にある第 1 の群のうちのコンピューティングノードのうちの 1 つ以上に基づいて、第 1 の群のうちのコンピューティングノードのプログラム実行容量の指示されたサブセットを含むことが各々望ましい、複数の個別の地理的な位置の第 1 のユーザによる指定を受信することをさらに含み、修正されたコンピューティングノードが、複数の地理的な位置の各々において第 1 の群のうちのコンピューティングノードのプログラム実行容量の指示されたサブセットを含むように修正されるように、受信した指定に従って、第 1 の群の一部であるコンピューティングノードを自動的に修正することがさらに実行される、第 3 3 節に記載の方法。

【 0 1 4 0 】

第 4 9 節。第 1 の群の一部であるコンピューティングノードを自動的に修正することは、1 つ以上の他のコンピューティングノードで、第 1 の群のうちのコンピューティングノードのうちの 1 つ以上を置換することを含み、1 つ以上の他のコンピューティングノードは、各々、1 つ以上の置換されたコンピューティングノードとは異なる量の利用可能なプログラム実行容量を有する、第 3 3 節に記載の方法。

【 0 1 4 1 】

第 5 0 節。第 1 のユーザについて 1 つ以上のソフトウェアプログラムを各々実行するために、1 回目第 1 の群のうちのコンピューティングノードを利用可能にすることは、第 1 の群のうちのコンピューティングノードの各々を自動的に供給することを含み、かつ、

10

20

30

40

50

第1の群のうちのコンピューティングノードの各々の上の1つ以上のソフトウェアプログラムの各々の実行を自動的に開始することを含む、第33節に記載の方法。

【0142】

第51節。プログラム実行サービスは、第1のユーザからリモートである料金ベースのネットワークでアクセス可能なサービスであり、プログラム実行サービスは、1つ以上のネットワークにおいてプログラム実行サービスとプログラムの相互作用するための、リモートコンピューティングシステムの1つ以上のアプリケーションプログラミングインターフェースと、1つ以上のネットワークにおいてリモートユーザによって使用するためのグラフィカルユーザインターフェースとの少なくとも1つを提供し、第1のユーザは、自動的に調和させることの一部として第1の群の一部であるコンピューティングノードを自動的に修正するための1つ以上の料金を支払う、第33節に記載の方法。

10

【0143】

第52節。プログラム実行サービスは、複数のコンピューティングノードが、複数の物理的コンピューティングシステムの各々について、各々、少なくとも1つのプログラムを実行することができる物理的コンピューティングシステムによってホストされる複数の仮想マシンを含むように、仮想技法を使用し、第1の群のうちのコンピューティングノードは、ホストされた仮想マシンであり、第1のユーザの1つ以上のプログラムは仮想マシンイメージの一部である、第33節に記載の方法。

【0144】

第53節。

20

構成されたコンピューティングシステムの制御下において、第1のユーザの代わりに、1つ以上のソフトウェアプログラムを実行するために使用するための望ましいプログラム実行容量を提供するために、1回目に第1のユーザに関連付けられる複数のコンピューティングノードの第1の群の指示を受信することと、

1回目後の2回目において、第1の群のうちのコンピューティングノードから第1のユーザに利用可能な実際のプログラム実行容量を自動的に判別することと、2回目の実際のプログラム実行容量は、第1の群の複数のコンピューティングノードのうちの1つ以上の利用可能性に対する1つ以上の変更少なくとも部分的に基づき、1回目に第1のユーザに提供された望ましいプログラム実行容量とは別である、判別することと、

各々、1回目後で生じ、かつ、望ましいプログラム実行容量の指示された修正を各々指定する、複数の独立したイベントに基づいて、2回目に第1のユーザに提供される修正された望ましいプログラム実行容量を自動的に判別することと、2回目の修正された望ましいプログラム実行容量は、複数の独立したイベントの指定された指示された修正の総計に基づき、かつ、1回目で提供される指示された望ましいプログラム実行容量とは別である、判別することと、

30

2回目に修正された望ましいプログラム実行容量を、2回目に判定された実際のプログラム実行容量と調和させるように、2回目に第1の群の一部であるコンピューティングノードを自動的に修正することと、修正されたコンピューティングノードは、第1のユーザに対する修正された望ましいプログラム実行容量を提供するために使用するためのものである、修正することと、を含む方法を実行することにより、そのコンテンツが、ユーザのプログラム実行容量を動的に修正するようにコンピューティングシステムを構成する、コンピュータ可読媒体。

40

【0145】

第54節。構成されたコンピューティングシステムは、プログラム実行サービスのユーザのプログラムを実行するように構成可能な複数のコンピューティングノードを提供するプログラム実行サービスの一部であり、第1の群の複数のコンピューティングノードは、複数のコンピューティングノードのサブセットであり、かつ、1回目の前に第1のユーザから受信される望ましいプログラム実行容量のリクエストに回答するプログラム実行サービスによって、1回目に第1のユーザに自動的に関連付けられ、第1のユーザは、望ましいプログラム実行容量に対する自動化された修正を後で開始するために使用するための1

50

つ以上の容量修正トリガをさらに含み、判定された修正された望ましいプログラム実行容量が基づく複数の独立したイベントは、満たされると判定される容量修正トリガの少なくとも1つを含み、かつ、第1のユーザによって動的に指定される少なくとも1つの容量修正命令を含む、第53節に記載のコンピュータ可読媒体。

【0146】

第55節。1つ以上のソフトウェアプログラムは、第1の群の一部ではない少なくとも1つのコンピューティングシステム上で実行されている少なくとも1つの他のソフトウェアプログラムを支援するために、第1の群の複数のコンピューティングノード上で実行され、満たされると判定される少なくとも1つの容量修正トリガのうちの1つ以上は、各々、少なくとも1つの他のソフトウェアプログラムの実行について取得された状態情報に基づき1つ以上の基準を有する、第54節に記載のコンピュータ可読媒体。

10

【0147】

第56節。第1のユーザへ1回目に提供される指示された望ましいプログラム実行容量は、1回目に第1のユーザに利用可能な初期の公的プログラム実行容量として、プログラム実行サービスによって記録され、方法は、1つ以上の利用可能性の変更を反映するように、第1の群の1つ以上のコンピューティングノードに対する1つ以上の利用可能性の変更を識別し、かつ、第1のユーザに利用可能なプログラム実行サービスの記録された公的プログラム実行容量を更新するために、1回目から2回目への期間において、第1の群のうちのコンピューティングノードを監視することをさらに含み、第1のユーザに利用可能なプログラム実行サービスの記録された公的プログラム実行容量は、2回目に修正された望ましいプログラム実行容量の自動的に判別に基づいて更新されず、2回目に第1の群の一部であるコンピューティングノードを修正することは、2回目に記録された公的プログラム実行容量を、2回目に修正された望ましいプログラム実行容量に調和させるようにさらに実行され、記録された公的プログラム実行容量を調和させることは、第1のユーザに利用可能なプログラム実行サービスの追加の更新された記録された公的プログラム実行容量として2回目に修正された望ましいプログラム実行容量を識別することを含む、第54節に記載のコンピュータ可読媒体。

20

【0148】

第57節。コンピュータ可読媒体は、コンテンツを記憶するコンピューティングシステムのメモリと、コンテンツを含有する生成された記憶されたデータ信号を含むデータ伝送媒体との少なくとも1つであり、コンテンツは、実行されるとコンピューティングシステムに方法を実行させる命令である、第53節に記載のコンピュータ可読媒体。

30

【0149】

第58節。ユーザのためにプログラム実行容量を動的に修正するように構成されたコンピューティングシステムであって、

1つ以上のプロセッサと、

複数のユーザの各々について、

ユーザの代わりに1つ以上のソフトウェアプログラムを実行するために使用するための望ましいプログラム実行容量を判別することと、後で、ユーザについて望ましいプログラム実行容量に対して自動化された修正を開始するために使用するための1つ以上の容量修正トリガを判別することと、

40

判定された望ましいプログラム実行容量をユーザへ提供するために使用するための、複数の利用可能なコンピューティングノードの群を1回目にユーザへ自動的に関連付けることと、

1回目の中の後の2回目で、群のうちのコンピューティングノードからユーザへ利用可能な実際のプログラム実行容量を1回目の後に自動的に判別することであって、2回目の実際のプログラム実行容量は、1回目にユーザに提供された望ましいプログラム実行容量とは別である、判別することと、

各々、1回目の後に生じ、かつ、各々、望ましいプログラム実行容量の関連付けられた修正を有する複数の独立したイベントに基づいて、2回目にユーザに提供される修正され

50

た望ましいプログラム実行容量を自動的に判別することであって、複数の独立したイベントは、満たされると判定される容量修正トリガのうちの少なくとも1つを含み、かつ、ユーザによって動的に指定される少なくとも1つの容量修正命令を含み、2回目に修正された望ましいプログラム実行容量は、複数の独立したイベントの関連付けられた修正の総計に基づき、判別することと、

2回目に修正された望ましいプログラム実行容量を、2回目に判定された実際のプログラム実行容量に調和させるように、2回目に群の一部であるコンピューティングノードを自動的に修正することであって、修正されたコンピューティングノードは、ユーザに修正された望ましいプログラム実行容量を提供するために使用するためのものである、修正することと、

1つ以上のプロセッサのうちの少なくとも1つによって実行されると、ネットワークでアクセス可能なサービスの複数のユーザについて、プログラム実行容量を管理するように構成されるシステムマネージャーモジュールと、を備える、コンピューティングシステム。

【0150】

第59節。複数のユーザのうちの1人のための判定された望ましいプログラム実行容量は、1人のユーザによって指示され、かつ、1回目で1人のユーザに利用可能であるとして記録される初期の公的プログラム実行容量として識別され、2回目に1人のユーザに利用可能である実際のプログラム実行容量を自動的に判別することは、1つ以上の変更を識別するように、1回目から2回目への期間において、1人のユーザの群のうちのコンピューティングノードを監視することに少なくとも部分的に基づき、コンピューティングノードを監視することは、1つ以上の変更を反映するように、1人のユーザに利用可能な記録された公的プログラム実行容量を更新することを含み、1人のユーザに利用可能な記録された公的プログラム実行容量は、2回目に修正された望ましいプログラム実行容量を自動的に判別することに基づいて更新されず、2回目に群の一部であるコンピューティングノードを修正することは、2回目に記録された公的プログラム実行容量を、2回目に修正された望ましいプログラム実行容量に調和させるようにさらに実行され、記録された公的プログラム実行容量を調和させることは、1人のユーザに利用可能なさらに更新された記録された公的プログラム実行容量として、2回目に修正された望ましいプログラム実行容量を識別することを含む、第58節に記載のコンピューティングシステム。

【0151】

第60節。1人のユーザのために判定された容量修正トリガは1人のユーザによって指定され、ネットワークでアクセス可能なサービスは、プログラム実行サービスのリモートユーザのプログラムを実行するように構成可能な複数のコンピューティングノードを提供するプログラム実行サービスであり、複数のユーザの少なくとも1人の各々について、群の複数のコンピューティングノードは、複数のコンピューティングノードのサブセットである、第59節に記載のコンピューティングシステム。

【0152】

第61節。システムマネージャーモジュールは、コンピューティングシステムによる実行のためのソフトウェア命令を含む、第58節に記載のコンピューティングシステム。

【0153】

第62節。複数のユーザの各々について、

ユーザの代わりに1つ以上のソフトウェアプログラムを実行するために使用するための望ましいプログラム実行容量を判別することと、ユーザのための望ましいプログラム実行容量に対する自動化された修正を後で開始するために、1つ以上の容量修正トリガを判別することと、

ユーザに判定された望ましいプログラム実行容量を提供するために使用するための、複数の利用可能なコンピューティングノードの群を1回目にユーザに自動的に関連付けることと、

1回目の後の後の2回目で、群のうちのコンピューティングノードからユーザに利用可能な実際のプログラム実行容量を1回目の後で自動的に判別することであって、2回目の

10

20

30

40

50

実際のプログラム実行容量は、1回目にユーザに提供された望ましいプログラム実行容量とは別である、判別することと、

各々、1回目の後に生じ、かつ、各々、望ましいプログラム実行容量の関連付けられた修正を有する、複数の独立したイベントに基づいて、2回目にユーザに提供される修正された望ましいプログラム実行容量を自動的に判別することであって、複数の独立したイベントは、満たされると判定される容量修正トリガの少なくとも1つを含み、かつ、ユーザによって動的に指定される少なくとも1つの容量修正命令を含み、2回目に修正された望ましいプログラム実行容量は、複数の独立したイベントの関連付けられた修正の総計に基づく、判別することと、

2回目に修正された望ましいプログラム実行容量を、2回目に判定された実際のプログラム実行容量に調和させるように、2回目に群の一部であるコンピューティングノードを自動的に修正することであって、修正されたコンピューティングノードは、修正された望ましいプログラム実行容量をユーザへ提供するために使用するものである、修正することと、により、システムマネージャモジュールは、ネットワークでアクセス可能なサービスの複数のユーザについて、プログラム実行容量を管理するための手段で構成される、第58節に記載のコンピューティングシステム。

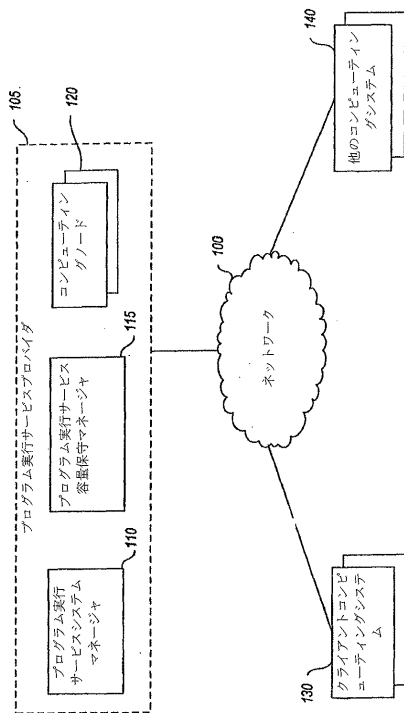
10

【0154】

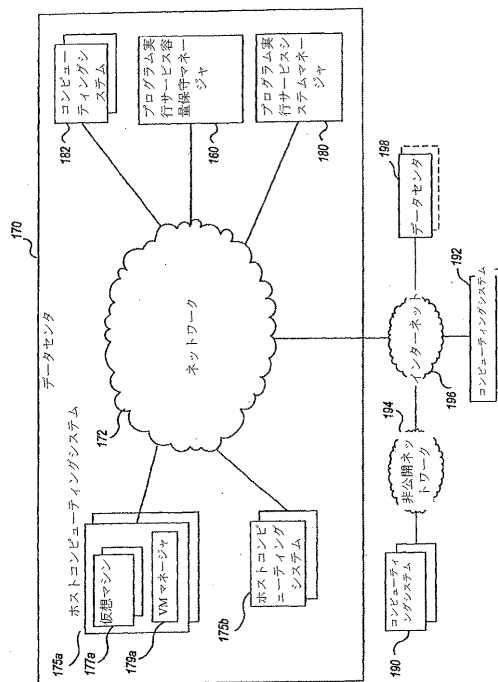
上記から、例証のために本明細書に特定の実施形態が記載されているが、本発明の精神および範囲から逸脱せずに多様な修正を行うことができることが理解されよう。従って、本発明は、添付の請求項およびそこで列挙されている要素により限定される場合を除いて制限されない。さらに、本発明の特定の態様は、特定の請求の形態で以下に表されているが、発明者は、任意の利用可能な請求の形態の、本発明の多様な態様を検討する。例えば、本発明のいくつかのみの態様が、コンピュータ可読媒体において具現化されるものとして現在、列挙されてもよく、他の態様も、同様に、そのように具現化されてもよい。

20

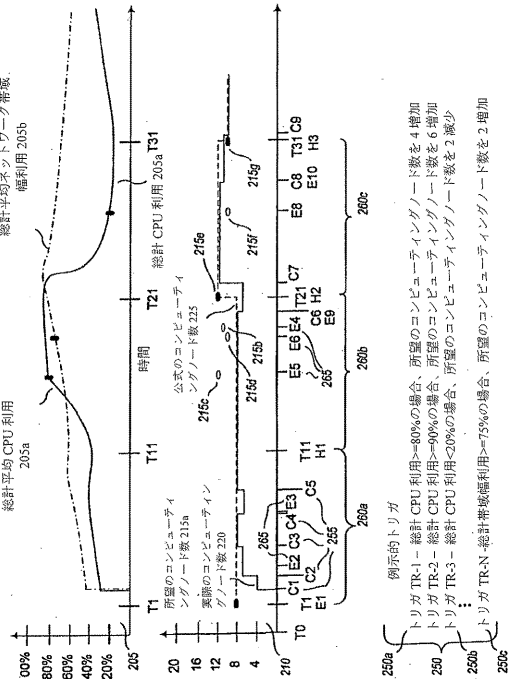
【図1A】



【図1B】



【図2A】



【図2B】

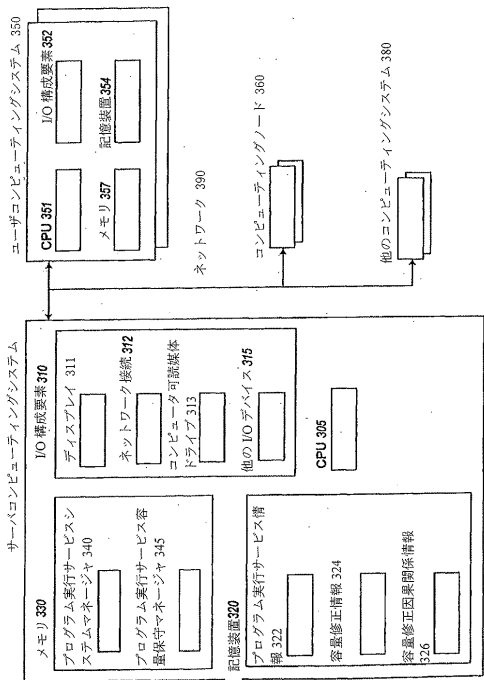
例示的コンピュティングノード群変更データベース

変更ID	グループ	変更日時	変更理由	変更時刻	終了時刻	変更イベントID
2500	TR-1	2004/10/10	ノード追加	10:00	10:05	N1
2501	TR-1	2004/10/10	ノード削除	10:05	10:10	N2
2502	TR-2	2004/10/10	ノード追加	10:10	10:15	N3
2503	TR-2	2004/10/10	ノード削除	10:15	10:20	N4
2504	TR-3	2004/10/10	ノード追加	10:20	10:25	N5
2505	TR-3	2004/10/10	ノード削除	10:25	10:30	N6
2506	TR-N	2004/10/10	ノード追加	10:30	10:35	N7
2507	TR-N	2004/10/10	ノード削除	10:35	10:40	N8
2508	TR-N	2004/10/10	ノード追加	10:40	10:45	N9
2509	TR-N	2004/10/10	ノード削除	10:45	10:50	N10

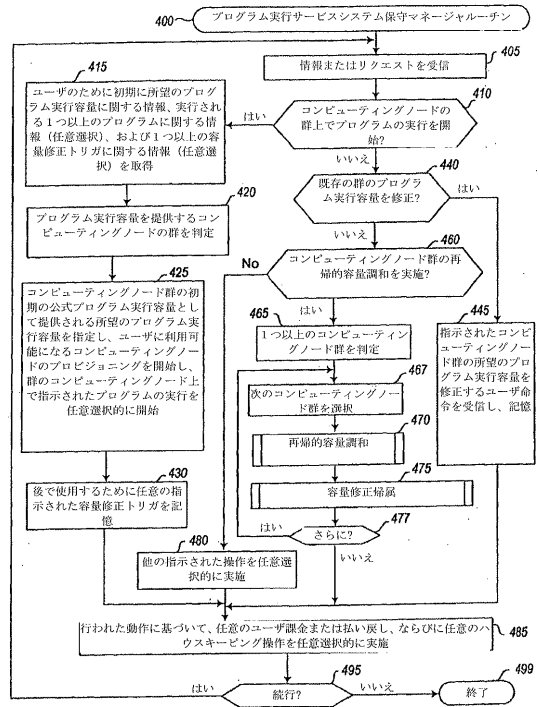
例示的コンピュティングノード群イベントデータベース

イベントID	イベント種類	時刻	時刻	時刻	時刻
2500	ノード追加	10:00	10:05	10:10	10:15
2501	ノード削除	10:05	10:10	10:15	10:20
2502	ノード追加	10:10	10:15	10:20	10:25
2503	ノード削除	10:15	10:20	10:25	10:30
2504	ノード追加	10:20	10:25	10:30	10:35
2505	ノード削除	10:25	10:30	10:35	10:40
2506	ノード追加	10:30	10:35	10:40	10:45
2507	ノード削除	10:35	10:40	10:45	10:50
2508	ノード追加	10:40	10:45	10:50	10:55
2509	ノード削除	10:45	10:50	10:55	11:00

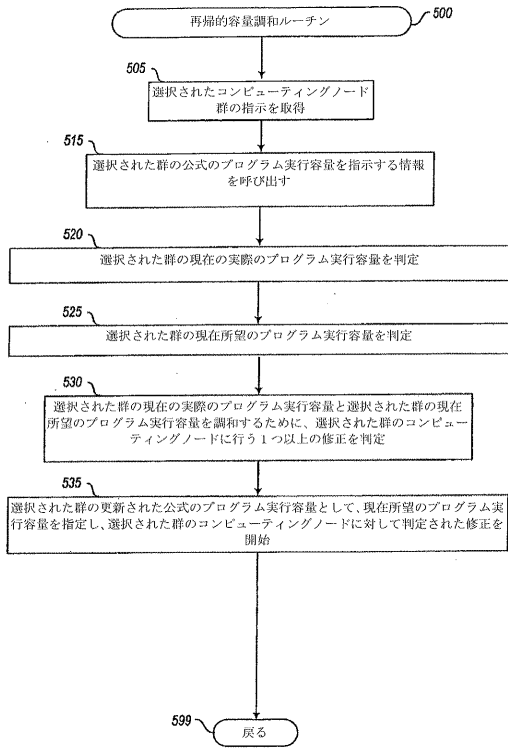
【図3】



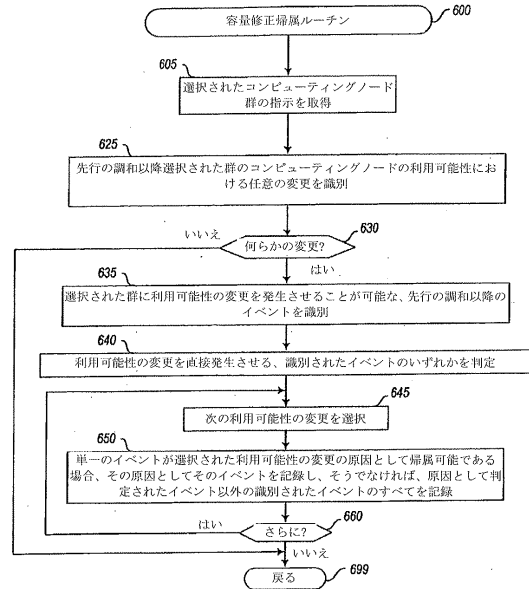
【図4】



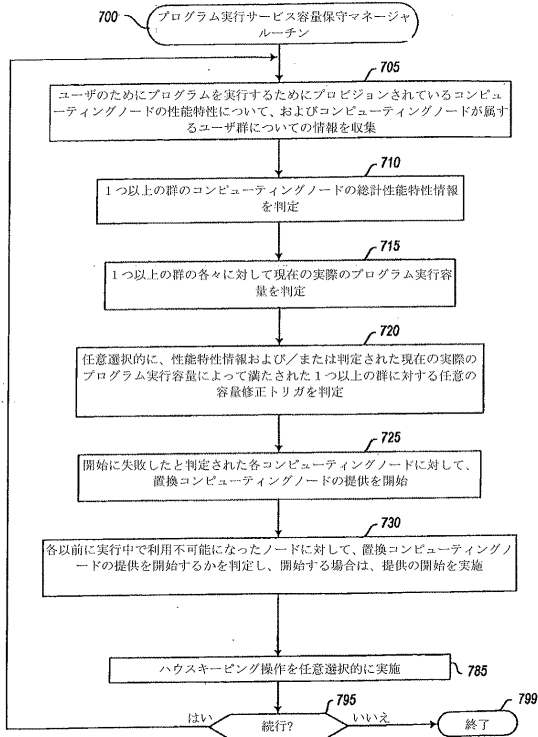
【図5】



【図6】



【図7】



フロントページの続き

- (72)発明者 チランジーブ ブラゴハイン
アメリカ合衆国 98109 ワシントン州 シアトル テリー アベニュー ノース 410
- (72)発明者 クリストファー レディ コマレディ
アメリカ合衆国 98109 ワシントン州 シアトル テリー アベニュー ノース 410
- (72)発明者 ジェフリー スコット パレ
アメリカ合衆国 98109 ワシントン州 シアトル テリー アベニュー ノース 410
- (72)発明者 ジョン ダブリュ.ヘイトマン
アメリカ合衆国 98109 ワシントン州 シアトル テリー アベニュー ノース 410
- (72)発明者 サミット ロヒア
アメリカ合衆国 98109 ワシントン州 シアトル テリー アベニュー ノース 410
- (72)発明者 チェン リアン
アメリカ合衆国 98109 ワシントン州 シアトル テリー アベニュー ノース 410
- (72)発明者 ザカリー エス.マスグレイブ
アメリカ合衆国 98109 ワシントン州 シアトル テリー アベニュー ノース 410

審査官 鈴木 修治

- (56)参考文献 特開2005-250818(JP,A)
特開2005-196601(JP,A)
特表2009-532771(JP,A)
米国特許出願公開第2006/0173856(US,A1)

- (58)調査した分野(Int.Cl., DB名)
G06F 9/46-9/54