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

特許第5689526号
(P5689526)

(45)発行日　平成27年3月25日(2015.3.25)　(24)登録日　平成27年2月6日(2015.2.6)

(51)Int.Cl.		F I		
H04L 29/10	(2006.01)	H04L	13/00	309Z
G06F 13/10	(2006.01)	G06F	13/10	330C
G06F 9/50	(2006.01)	G06F	9/46	462Z
H04L 13/08	(2006.01)	H04L	13/08	

請求項の数 15　(全 18 頁)

(21)出願番号　特願2013-505372(P2013-505372)
(86)(22)出願日　平成23年3月1日(2011.3.1)
(65)公表番号　特表2013-530573(P2013-530573A)
(43)公表日　平成25年7月25日(2013.7.25)
(86)国際出願番号　PCT/EP2011/052992
(87)国際公開番号　WO2011/131400
(87)国際公開日　平成23年10月27日(2011.10.27)
審査請求日　平成25年11月11日(2013.11.11)
(31)優先権主張番号　12/766,282
(32)優先日　平成22年4月23日(2010.4.23)
(33)優先権主張国　米国(US)

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

最終頁に続く

(54)【発明の名称】マルチキュー・ネットワーク・アダプタの動的再構成によるリソース・アフィニティ

(57)【特許請求の範囲】

【請求項１】

データ処理システムにおいて、動的再構成によってマルチキュー・ネットワーク・アダプタのリソース・アフィニティを提供する方法であって、

前記データ処理システム内のデバイス・ドライバによって、メモリ内に初期キュー・ペアを割り当てるステップと、

前記デバイス・ドライバによって、前記データ処理システムのワークロードが所定の高閾値より高くなったかどうかを判断するステップと、

前記ワークロードが前記所定の高閾値より高くなることに応じて、前記デバイス・ドライバによって、前記メモリ内に追加のキュー・ペアを割り当て、前記追加のキュー・ペアを初期化するステップと、

前記追加のキュー・ペアに関連する追加の処理エンジンの動的挿入を可能にするために、前記デバイス・ドライバによって、ネットワーク・アダプタ内の受信側スケーリング（ＲＳＳ）メカニズムをプログラムするステップと、

前記デバイス・ドライバによって、前記追加のキュー・ペアへの送信タプル・ハッシングを有効化するステップと、

を含む方法。

【請求項２】

前記追加のキュー・ペアを割り当て、前記追加のキュー・ペアを初期化するステップ、前記プログラムするステップ、および前記有効化するステップが、前記デバイス・ドライ

バによって、前記データ処理システムの前記ワークロードが前記所定の高閾値より高くなるたびに実行される、請求項１に記載の方法。
【請求項３】
前記追加のキュー・ペアに関連する前記追加の処理エンジンの動的挿入を可能にするために、前記ネットワーク・アダプタ内の前記受信側スケーリング（ＲＳＳ）メカニズムをプログラムするステップは、
前記デバイス・ドライバによって、前記初期キュー・ペアへの送信タプル・ハッシングを有効化するステップ
を含む、請求項１に記載の方法。
【請求項４】
前記データ処理システムの前記ワークロードが前記所定の高閾値より高くなったかどうかを判断するステップは、前記デバイス・ドライバによって、データ・フローおよびリソース利用可能性を介して前記データ処理システムの前記ワークロードを監視することによって実行される、請求項１に記載の方法。
【請求項５】
前記デバイス・ドライバは、前記ワークロードに関連する少なくとも１つのパラメータを監視し、前記少なくとも１つのパラメータは、送信／受信バイト・パー・セカンド、前記ネットワーク・アダプタによって送信および受信されているフロー制御フレームの数、前記ネットワーク・アダプタによって検出されるＤＭＡオーバーランの数、前記デバイス・ドライバによって検出される送信タイムアウト・イベントの数、割り込み毎に前記デバイス・ドライバによって処理される受信パケットの数、またはソフトウェア・キュー上の送信パケットの数のうちの少なくとも１つである、請求項４に記載の方法。
【請求項６】
前記デバイス・ドライバによって、前記ワークロードが所定の低閾値を下回ったかどうかを判断するステップと、
前記ワークロードが前記所定の低閾値を下回るのに応じて、前記メモリ内に割り当てられているキュー・ペアが残り１つのみあるかどうかを、前記デバイス・ドライバによって判断するステップと、
前記メモリ内に割り当てられているキュー・ペアが２つ以上残っていることに応じて、割り当てられているキュー・ペアの削除を可能にするために、前記デバイス・ドライバによって前記ネットワーク・アダプタ内の前記ＲＳＳメカニズムを再プログラムするステップと、
特定されたキュー・ペアへの送信タプル・ハッシングを、前記デバイス・ドライバによって無効化するステップと、
前記特定されたキュー・ペアへの前記ワークロードが休止したかどうかを、前記デバイス・ドライバによって判断するステップと、
前記特定されたキュー・ペアへの前記ワークロードが休止することに応じて、前記特定されたキュー・ペアを前記デバイス・ドライバによってメモリから除去し、その結果、前記特定されたキュー・ペアによって使用されたメモリを解放するステップと、
をさらに含む、請求項１に記載の方法。
【請求項７】
前記特定されたキュー・ペアをメモリから除去し、その結果、前記特定されたキュー・ペアによって使用されたメモリを解放するステップより前に、前記特定されたキュー・ペアへの前記ワークロードが休止するのを、前記特定されたキュー・ペアへの前記ワークロードが休止しないことに応じて、前記デバイス・ドライバによって待つステップ
をさらに含む、請求項６に記載の方法。
【請求項８】
請求項１～７の何れか１項に記載の方法の各ステップをコンピュータに実行させる、コンピュータ・プログラム。
【請求項９】

プロセッサと、
前記プロセッサに結合されたメモリと、
を含む装置であって、前記メモリは、命令を含み、前記命令は、前記プロセッサによって実行されると、前記プロセッサに、
メモリ内に初期キュー・ペアを割り当てること、
前記データ処理システムのワークロードが所定の高閾値より高くなったかどうかを判断すること、
前記ワークロードが前記所定の高閾値より高くなることに応じて、前記メモリ内に追加のキュー・ペアを割り当て、前記追加のキュー・ペアを初期化すること、
前記追加のキュー・ペアに関連する追加の処理エンジンの動的挿入を可能にするために、ネットワーク・アダプタ内の受信側スケーリング（ＲＳＳ）メカニズムをプログラムすること、および
前記追加のキュー・ペアへの送信タプル・ハッシングを有効化すること
を行わせる、装置。
【請求項１０】
前記命令は、前記プロセッサに、
前記追加のキュー・ペアを割り当て、前記追加のキュー・ペアを初期化すること、前記プログラムすること、および前記有効化することを行わせる命令を、前記データ処理システムの前記ワークロードが前記所定の高閾値より高くなるたびに実行させる、請求項９に記載の装置。
【請求項１１】
前記追加のキュー・ペアに関連する前記追加の処理エンジンの動的挿入を可能にするために、前記ネットワーク・アダプタ内の前記受信側スケーリング（ＲＳＳ）メカニズムをプログラムするための前記命令は、前記プロセッサに、
前記初期キュー・ペアへの送信タプル・ハッシングを有効化すること
をさらに行わせる、請求項９に記載の装置。
【請求項１２】
前記データ処理システムの前記ワークロードが前記所定の高閾値より高くなったかどうかを判断するための前記命令は、データ・フローおよびリソース利用可能性を介して前記データ処理システムの前記ワークロードを監視することを前記プロセッサによって実行される、請求項９に記載の装置。
【請求項１３】
前記命令は、前記ワークロードに関連する少なくとも１つのパラメータを監視することを前記プロセッサにさらに行わせ、前記少なくとも１つのパラメータは、送信／受信バイト・パー・セカンド、前記ネットワーク・アダプタによって送信および受信されているフロー制御フレームの数、前記ネットワーク・アダプタによって検出されるＤＭＡオーバーランの数、前記デバイス・ドライバによって検出される送信タイムアウト・イベントの数、割り込み毎に前記デバイス・ドライバによって処理される受信パケットの数、またはソフトウェア・キュー上の送信パケットの数のうちの少なくとも１つである、請求項１２に記載の装置。
【請求項１４】
前記命令は、前記プロセッサに、
前記ワークロードが所定の低閾値を下回ったかどうかを判断すること、
前記ワークロードが前記所定の低閾値を下回るのに応じて、前記メモリ内に割り当てられているキュー・ペアが残り１つのみあるかどうかを判断すること、
前記メモリ内に割り当てられているキュー・ペアが２つ以上残っていることに応じて、割り当てられているキュー・ペアの削除を可能にするために前記ネットワーク・アダプタ内の前記ＲＳＳメカニズムを再プログラムすること、
特定されたキュー・ペアへの送信タプル・ハッシングを無効化すること、
前記特定されたキュー・ペアへの前記ワークロードが休止したかどうかを判断すること

、および、
前記特定されたキュー・ペアへの前記ワークロードが休止することに応じて、前記特定されたキュー・ペアをメモリから除去し、その結果、前記特定されたキュー・ペアによって使用されたメモリを解放すること
をさらに行わせる、請求項９に記載の装置。
【請求項１５】
前記命令は、前記プロセッサに、
前記特定されたキュー・ペアをメモリから除去し、その結果、前記特定されたキュー・ペアによって使用されたメモリを解放することより前に、前記特定されたキュー・ペアへの前記ワークロードが休止するのを、前記特定されたキュー・ペアへの前記ワークロードが休止しないことに応じて待つこと
をさらに行わせる、請求項１４に記載の装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本願は、全般的に、改善されたデータ処理装置および方法に関し、特に、動的再構成（ｄｙｎａｍｉｃ　ｒｅｃｏｎｆｉｇｕｒａｔｉｏｎ）によってマルチキュー・ネットワーク・アダプタのリソース・アフィニティを提供するメカニズムに関する。
【背景技術】
【０００２】
イーサネット（Ｒ）媒体の速度が上がり続けるにつれて、可能な限り低いレイテンシで理論上最大限のパフォーマンスを実現するために、より多くのシステム・リソースを使用する必要性が高まっている。イーサネット（Ｒ）・アダプタ要件に関するシステム・リソースには、多数の送信／受信ディスクリプタおよびバッファの必要性から、大きな物理メモリ・フットプリントおよび対応する直接メモリ・アクセス入出力メモリ・マッピング・リソースが含まれる。１０Ｇｂｐｓイーサネット（Ｒ）・ドライバは、典型的には、アダプタ毎に約１５０～３００ＭＢの物理システム・メモリおよび直接メモリ・アクセス入出力メモリを消費する。
【発明の概要】
【発明が解決しようとする課題】
【０００３】
従来のドライバ・モデルのもとでは、デバイス・ドライバは、アダプタがその理論上最大限のパフォーマンスを実現できるであろう量に送信／受信リソースの割り当てを行う。しかし、最大パフォーマンス限界が必要とされないようなワークロード（ｗｏｒｋｌｏａｄ）またはネットワーク・トラフィックであれば、ドライバは必要以上に多くのリソースを消費していることになり、これはシステム・リソースの無駄である。さらに、このモデルには、変化するワークロードに効率的に対処する能力がない。
【課題を解決するための手段】
【０００４】
一例示実施形態では、データ処理システムにおいて、動的再構成によってマルチキュー・ネットワーク・アダプタのリソース・アフィニティを提供する方法が提供される。例示実施形態は、メモリ内に初期キュー・ペア（ｉｎｉｔｉａｌ　ｑｕｅｕｅ　ｐａｉｒ）を割り当てる。例示実施形態は、データ処理システムのワークロードが所定の高閾値より高くなったかどうかを判断する。例示実施形態は、ワークロードが所定の高閾値より高くなることに応じて、メモリ内に追加のキュー・ペアを割り当て、初期化する。例示実施形態は、追加のキュー・ペアに関連する追加の処理エンジン（ｐｒｏｃｅｓｓｉｎｇ　ｅｎｇｉｎｅ）の動的挿入（ｄｙｎａｍｉｃ　ｉｎｓｅｒｔｉｏｎ）を可能にするために、ネットワーク・アダプタ内の受信側スケーリング（ＲＳＳ：ｒｅｃｅｉｖｅ　ｓｉｄｅ　ｓｃａｌｉｎｇ）メカニズムをプログラムする。例示実施形態は、追加のキュー・ペアへの送信タプル・ハッシング（ｔｒａｎｓｍｉｔ　ｔｕｐｌｅ　ｈａｓｈｉｎｇ）を有効化する

。
【０００５】
別の例示実施形態では、コンピュータ可読プログラムを有するコンピュータ使用可能または可読媒体を含むコンピュータ・プログラム製品が提供される。コンピュータ可読プログラムは、コンピューティング・デバイス上で実行されると、方法の例示実施形態に関して上記に要点を記載した動作のうちの様々な動作、およびその組み合わせを、コンピューティング・デバイスに実行させる。
【０００６】
さらに別の例示実施形態では、システム／装置が提供される。システム／装置は、１つ以上のプロセッサと、該１つ以上のプロセッサに結合されたメモリとを含むとよい。メモリは、該１つ以上のプロセッサにより実行されると、方法の例示実施形態に関して上記に要点を記載した動作のうちの様々な動作、およびその組み合わせを該１つ以上のプロセッサに実行させる、命令を含む。
【０００７】
本発明のこれらの特徴および利点、ならびにその他の特徴および利点が、以下の本発明の例示の実施形態の詳細な説明に記載されるか、またはそれを考慮することで当業者には明らかとなる。
【０００８】
例示実施形態の以下の詳細な説明を添付の図面と併せて読み、参照することで、本発明、ならびにその好適な使用方法、ならびにさらなる目的および利点が、最も深く理解される。
【図面の簡単な説明】
【０００９】
【図１】例示実施形態の各側面が実装され得る例示の分散型データ処理システムの図的表現を示す。
【図２】例示実施形態の側面がともに有利に利用され得るデータ処理システムのブロック図を示す。
【図３】例示実施形態が実装され得る、例示的な論理分割されたプラットフォームのブロック図を示す。
【図４】例示実施形態による、動的再構成によってマルチキュー・ネットワーク・アダプタのリソース・アフィニティを提供するメカニズムの例示的な実装を示す。
【図５】例示実施形態による、動的再構成によってマルチキュー・ネットワーク・アダプタのリソース・アフィニティを提供する例示の動作の概要を示すフローチャートを提供する。
【発明を実施するための形態】
【００１０】
例示実施形態は、基礎をなすハードウェアの動的再構成によって、アクティブ・メモリ・シェアリング（ＡＭＳ：ａｃｔｉｖｅ　ｍｅｍｏｒｙ　ｓｈａｒｉｎｇ）および中央処理ユニット（ＣＰＵ：ｃｅｎｔｒａｌ　ｐｒｏｃｅｓｓｉｎｇ　ｕｎｉｔ）利用のためのリソース・アフィニティを提供するメカニズムを提供し、パフォーマンスまたはサービスの中断なしに、変化するワークロードの要求に応じる。現代のアダプタは、最大限のパフォーマンスのために複数のパケット・キュー・ペア（ＱＰ：ｑｕｅｕｅ　ｐａｉｒ）を提供することもある。こうしたアダプタは、インターフェイス毎に複数の送信／受信キュー（ＱＰ）を使用することによって、並列ネットワーク・データ処理を実行できることもあり、これは、高トランザクション・ワークロード、および小さなパケット・サイズでより高速な回線速度を実現するためには必須の機能である。イングレス、すなわち受信トラフィックは、アダプタにより、オペレーティング・システム処理のために、適切なＱＰおよび関連する割り込みにハッシュされるタプルとされ得る。イーグレス、すなわち送信トラフィックは、アダプタに配信するためにオペレーティング・システム（ＯＳ：ｏｐｅｒａｔｉｎｇ　ｓｙｓｔｅｍ）ドライバによってハッシュされるタプルとされ得る。アダプタ

およびＯＳドライバは、典型的には、最大限のパフォーマンスを実現するために、十分なディスクリプタおよびバッファとともに複数のＱＰ、通常はアダプタ毎に約２５０ＭＢの平均メモリ・フットプリントに対し２～４つのＱＰを割り当てる。各ＱＰが、関連する受信割り込みを有することもあるため、複数のＱＰが使用されていて、トラフィックが少ないと、単一のＱＰで容易に対処可能であるのに、増加した割り込み送出が原因で追加のＣＰＵ利用オーバーヘッドが存在することになる。複数のＱＰは、多くの通常の使用事例に関するパフォーマンスには負の影響を有するということ、しかし特定の高ストレス・高トランザクション・ワークロードに関しては、理論上最大限のパフォーマンスを実現するために必須であるということが、既知のアダプタの分析により指摘されているため、この問題に対するパフォーマンス・チームの関心は次第に高まっている。

【００１１】

したがって、例示実施形態は、分散型データ処理環境、単一データ処理デバイス、または同様のものを含む、多数の異なるタイプのデータ処理環境において利用され得る。例示実施形態の具体的な構成要素および機能性を説明するための背景を提供するために、以下、図１～３が、例示実施形態の各側面が実装されるとよい例示の環境として提供される。図１～３に続く説明は、主として、動的再構成によってマルチキュー・ネットワーク・アダプタのリソース・アフィニティを提供するメカニズムの単一データ処理デバイス実装に焦点を当てるが、これは例でしかなく、本発明の特徴に関していかなる制限を記載することも、示唆することも目的としていない。反対に、例示実施形態は、動的再構成によってマルチキュー・ネットワーク・アダプタのリソース・アフィニティが提供されるとよい分散型データ処理の環境および実施形態を含むものとする。

【００１２】

以下、図面、特に図１～３を参照する。本発明の例示実施形態が実装され得るデータ処理環境の例示の図が提供されている。当然のことながら、図１～３は例でしかなく、本発明の側面または実施形態が実装され得る環境に関していかなる制限を主張することも、示唆することも目的としていない。示される環境に対して多数の変更が、本発明の意図および範囲から逸脱することなく加えられ得る。

【００１３】

以下、図面を参照する。図１は、例示実施形態の各側面が実装され得る例示の分散型データ処理システムの図的表現を示す。分散型データ処理システム１００は、例示実施形態の各側面が実装され得るコンピュータのネットワークを含むとよい。分散型データ処理システム１００は、少なくとも１つのネットワーク１０２を含み、ネットワーク１０２は、分散型データ処理システム１００内で互いに接続された様々なデバイスとコンピュータとの通信リンクを提供するために使用される媒体である。ネットワーク１０２は、有線、無線通信リンク、または光ファイバ・ケーブルなどの接続を含み得る。

【００１４】

示されている例では、サーバ１０４およびサーバ１０６が、ストレージ・ユニット１０８とともに、ネットワーク１０２に接続されている。さらに、クライアント１１０、１１２、および１１４もネットワーク１０２に接続されている。これらのクライアント１１０、１１２、および１１４は、例えばパーソナル・コンピュータ、ネットワーク・コンピュータ、または同様のものであるとよい。示されている例では、サーバ１０４が、ブート・ファイル、オペレーティング・システム・イメージ、およびアプリケーションなどのデータを、クライアント１１０、１１２、および１１４に提供する。クライアント１１０、１１２、および１１４は、示されている例では、サーバ１０４のクライアントである。分散型データ処理システム１００は、示されていない追加のサーバ、クライアント、およびその他のデバイスを含んでもよい。

【００１５】

示されている例では、分散型データ処理システム１００は、インターネットであり、ネットワーク１０２は、互いに通信するために伝送制御プロトコル／インターネット・プロトコル（ＴＣＰ／ＩＰ：Ｔｒａｎｓｍｉｓｓｉｏｎ　Ｃｏｎｔｒｏｌ　Ｐｒｏｔｏｃｏｌ

／Ｉｎｔｅｒｎｅｔ　Ｐｒｏｔｏｃｏｌ）プロトコル・スイートを使用する、ネットワークおよびゲートウェイの世界的規模の集合を表現している。インターネットの中心には、データおよびメッセージをルーティングする数千の商用、政府機関用、教育用、およびその他のコンピュータ・システムから成る主要ノードまたはホスト・コンピュータ間の高速データ通信回線のバックボーンがある。当然、分散型データ処理システム１００は、例えばイントラネット、ローカル・エリア・ネットワーク（ＬＡＮ：ｌｏｃａｌ　ａｒｅａ　ｎｅｔｗｏｒｋ）、広域ネットワーク（ＷＡＮ：ｗｉｄｅ　ａｒｅａ　ｎｅｔｗｏｒｋ）、または同様のものなど、いくつかの異なるタイプのネットワークを含むようにも実装され得る。上記のように、図１は、本発明の種々の実施形態のアーキテクチャ上の制限としてではなく、例として意図されており、したがって、図１に示されている特定の構成要素は、本発明の例示実施形態が実装され得る環境に関する制限的なものと見なされてはならない。

【００１６】

例示実施形態では、コンピュータ・アーキテクチャは、ハードウェアおよびソフトウェアの組み合わせとして実装される。コンピュータ・アーキテクチャのソフトウェア部分は、マイクロコードまたはミリコードと呼ばれることもある。ハードウェアおよびソフトウェアの組み合わせは、命令セットおよびシステム・アーキテクチャを作成し、これに基づき、基本入出力システム（ＢＩＯＳ：Ｂａｓｉｃ　Ｉｎｐｕｔ／Ｏｕｔｐｕｔ　Ｓｙｓｔｅｍ）、仮想マシン・モニタ（ＶＭＭ：Ｖｉｒｔｕａｌ　Ｍａｃｈｉｎｅ　Ｍｏｎｉｔｏｒｓ）、ハイパーバイザ、アプリケーションなどのコンピュータのソフトウェアの残りが動作する。初期の組み合わせによって作成されるコンピュータ・アーキテクチャは、ごく少数しかないと考えられる定義済みインターフェイスを介する場合を除き、コンピュータ・ソフトウェア（ＢＩＯＳなど）に対し不変である。

【００１７】

以下、図２を参照する。例示実施形態の側面がともに有利に利用され得るデータ処理システムのブロック図が示されている。図のように、データ処理システム２００は、プロセッサ・ユニット２１１ａ～２１１ｎを含む。プロセッサ・ユニット２１１ａ～２１１ｎはそれぞれ、プロセッサおよびキャッシュ・メモリを含む。例えば、プロセッサ・ユニット２１１ａは、プロセッサ２１２ａおよびキャッシュ・メモリ２１３ａを含み、プロセッサ・ユニット２１１ｎは、プロセッサ２１２ｎおよびキャッシュ・メモリ２１３ｎを含む。

【００１８】

プロセッサ・ユニット２１１ａ～２１１ｎは、メイン・バス２１５に接続されている。メイン・バス２１５は、プロセッサ・ユニット２１１ａ～２１１ｎおよびメモリ・カード２２３を含むシステム・プレーナ２２０をサポートしている。システム・プレーナ２２０はさらに、データ・スイッチ２２１およびメモリ・コントローラ／キャッシュ２２２を含む。メモリ・コントローラ／キャッシュ２２２は、複数のデュアル・インライン・メモリ・モジュール（ＤＩＭＭ：ｄｕａｌ　ｉｎ－ｌｉｎｅ　ｍｅｍｏｒｙ　ｍｏｄｕｌｅ）を有するローカル・メモリ２１６を含むメモリ・カード２２３をサポートしている。

【００１９】

データ・スイッチ２２１は、ネイティブＩ／Ｏ（ＮＩＯ：ｎａｔｉｖｅ　Ｉ／Ｏ）プレーナ２２４内に位置するバス・ブリッジ２１７およびバス・ブリッジ２１８と接続している。図のように、バス・ブリッジ２１８は、システム・バス２１９を経由して、ペリフェラル・コンポーネント・インターコネクト（ＰＣＩ：ｐｅｒｉｐｈｅｒａｌ　ｃｏｍｐｏｎｅｎｔｓ　ｉｎｔｅｒｃｏｎｎｅｃｔ）ブリッジ２２５および２２６と接続している。ＰＣＩブリッジ２２５は、ＰＣＩバス２２８を経由して様々なＩ／Ｏデバイスと接続している。図のように、ハード・ディスク２３６が、小型コンピュータ・システム・インターフェイス（ＳＣＳＩ：ｓｍａｌｌ　ｃｏｍｐｕｔｅｒ　ｓｙｓｔｅｍ　ｉｎｔｅｒｆａｃｅ）ホスト・アダプタ２３０を経由してＰＣＩバス２２８に接続されているとよい。グラフィックス・アダプタ２３１が、直接的または間接的にＰＣＩバス２２８に接続されているとよい。ＰＣＩブリッジ２２６は、ＰＣＩバス２２７を経由して、ネットワーク・アダ

プタ２３４およびアダプタ・カード・スロット２３５ａ～２３５ｎを介して外部データ・ストリームに対する接続を提供する。

【００２０】

業界標準アーキテクチャ（ＩＳＡ：Ｉｎｄｕｓｔｒｙ　ｓｔａｎｄａｒｄ　ａｒｃｈｉｔｅｃｔｕｒｅ）バス２２９が、ＩＳＡブリッジ２３２を経由してＰＣＩバス２２８と接続している。ＩＳＡブリッジ２３２は、シリアル接続シリアル１およびシリアル２を有するＮＩＯコントローラ２３３を介して相互接続機能を提供する。データ処理システム２００が対応する入力デバイスを経由してユーザからのデータ入力を受け取ることを可能にするために、フレキシブル・ディスク・ドライブ接続、キーボード接続、およびマウス接続が、ＮＩＯコントローラ２３３によって提供されている。さらに、ＩＳＡバス２２９に接続されている不揮発性ＲＡＭ（ＮＶＲＡＭ：ｎｏｎ－ｖｏｌａｔｉｌｅ　ｒａｎｄｏｍ－ａｃｃｅｓｓ　ｍｅｍｏｒｙ）２４０は、電力供給問題などのシステム障害またはシステム故障から特定のタイプのデータを保護する不揮発性メモリを提供する。システム・ファームウェア２４１もＩＳＡバス２２９に接続されており、初期基本入出力システム（ＢＩＯＳ）機能を実装する。サービス・プロセッサ２４４が、ＩＳＡバス２２９に接続しており、システム診断またはシステム・サービスの機能性を提供する。

【００２１】

オペレーティング・システム（ＯＳ）は、ハード・ディスク２３６に記憶され、ハード・ディスク２３６はさらに、データ処理システムによって実行されるさらなるアプリケーション・ソフトウェアのストレージを提供するとよい。ＮＶＲＡＭ２４０は、フィールド交換可能ユニット（ＦＲＵ：ｆｉｅｌｄ　ｒｅｐｌａｃｅａｂｌｅ　ｕｎｉｔ）の分離のために、システム変数およびエラー情報を記憶するよう使用される。システム起動中、ブートストラップ・プログラムが、オペレーティング・システムをロードし、オペレーティング・システムの実行を開始する。オペレーティング・システムをロードするために、ブートストラップ・プログラムはまず、オペレーティング・システム・カーネル・イメージをハード・ディスク２３６上に置き、ＯＳカーネル・イメージをメモリ内にロードし、オペレーティング・システム・カーネルによって提供される先頭アドレスにジャンプする。典型的には、オペレーティング・システムは、データ処理システム内のランダム・アクセス・メモリ（ＲＡＭ：ｒａｎｄｏｍ－ａｃｃｅｓｓ　ｍｅｍｏｒｙ）内にロードされる。ロードおよび初期化が行われると、オペレーティング・システムは、プログラムの実行を制御し、リソース割り当て、スケジューリング、入出力制御、およびデータ管理などのサービスを提供するとよい。

【００２２】

例示実施形態は、いくつかの異なるハードウェア構成、ならびにブートストラップ・プログラムおよびオペレーティング・システムなどのソフトウェアを利用する、様々なデータ処理システムにおいて具現化され得る。データ処理システム２００は、例えば、スタンドアロン・システム、またはローカル・エリア・ネットワーク（ＬＡＮ）もしくは広域ネットワーク（ＷＡＮ）などのネットワークの一部としてもよい。上記のように、図２は、本発明の種々の実施形態のアーキテクチャ上の制限としてではなく、例として意図されており、したがって、図２に示されている特定の構成要素は、本発明の例示実施形態が実装され得る環境に関する制限的なものと見なされてはならない。

【００２３】

以下、図３を参照する。例示実施形態が実装され得る、例示的な論理分割されたプラットフォームのブロック図が示されている。論理分割されたプラットフォーム３００内のハードウェアは、例えば、図２のデータ処理システム２００のハードウェアを使用して実装されてもよい。

【００２４】

論理分割されたプラットフォーム３００は、分割されたハードウェア３３０、オペレーティング・システム３０２、３０４、３０６、３０８、および仮想マシン・モニタ３１０を含む。オペレーティング・システム３０２、３０４、３０６、および３０８は、単一の

オペレーティング・システムの複数のコピーであってもよく、または論理分割されたプラットフォーム３００上で同時に実行する複数の異種オペレーティング・システムであってもよい。これらのオペレーティング・システムは、例えば、ハイパーバイザなどのパーティション管理ファームウェアなど、仮想化メカニズムとインターフェイスをとるよう設計されている、ｚ／ＯＳ（ＩＢＭ社の登録商標）を使用して実装されてもよい。ｚ／ＯＳは、これらの例示実施形態において単なる例として使用されている。当然、特定の実装に応じて、ＯＳ／４００（ＩＢＭ社の登録商標）、ＡＩＸ（Ｒ）、およびＬｉｎｕｘ（Ｒ）など、他のタイプのオペレーティング・システムが使用されてもよい。オペレーティング・システム３０２、３０４、３０６、および３０８は、それぞれ論理パーティション３０３、３０５、３０７、および３０９に位置する。

【００２５】

ハイパーバイザ・ソフトウェアは、プラットフォームの実装に使用され得るソフトウェアの例であり（この例では、仮想マシン・モニタ３１０）、インターナショナル・ビジネス・マシーンズ・コーポレーションから入手できる。ファームウェアは、例えば読み取り専用メモリ（ＲＯＭ：ｒｅａｄ－ｏｎｌｙ　ｍｅｍｏｒｙ）、プログラム可能ＲＯＭ（ＰＲＯＭ：ｐｒｏｇｒａｍｍａｂｌｅ　ＲＯＭ）、消去可能プログラム可能ＲＯＭ（ＥＰＲＯＭ：ｅｒａｓａｂｌｅ　ｐｒｏｇｒａｍｍａｂｌｅ　ＲＯＭ）、および電気的消去可能プログラム可能ＲＯＭ（ＥＥＰＲＯＭ：ｅｌｅｃｔｒｉｃａｌｌｙ　ｅｒａｓａｂｌｅ　ｐｒｏｇｒａｍｍａｂｌｅ　ＲＯＭ）など、電力がなくてもそのコンテンツを保持するメモリ・チップに記憶された、「ソフトウェア」である。

【００２６】

論理分割されたプラットフォーム３００はさらに、システム・メモリ仮想化機能をＩＢＭ　Ｐｏｗｅｒ　Ｓｙｓｔｅｍｓに提供するＩＢＭ（Ｒ）ＰｏｗｅｒＶＭ（ＩＢＭ社の登録商標）の高度なメモリ仮想化技術である、ＩＢＭ（Ｒ）のＰｏｗｅｒＶＭ（ＩＢＭ社の登録商標）Ａｃｔｉｖｅ　Ｍｅｍｏｒｙ（ＴＭ）Ｓｈａｒｉｎｇを使用して、複数の論理パーティションが物理メモリの共通のプールを共有できるようにすることもできる。ＩＢＭ（Ｒ）Ｐｏｗｅｒ　Ｓｙｓｔｅｍｓ（ＴＭ）の物理メモリは、専用または共有モードのいずれかで、複数の論理パーティションに割り振られることが可能である。システム管理者は、一部の物理メモリをある論理パーティションに割り振り、一部の物理メモリを他の複数の論理パーティションにより共有されるプールに割り振る能力を有する。単一のパーティションは、専用または共有メモリのいずれかを有し得る。Ａｃｔｉｖｅ　Ｍｅｍｏｒｙ（ＴＭ）Ｓｈａｒｉｎｇは、必要システム・メモリを減らすこと、または既存システム上での追加の論理パーティションの作成を可能にすることによって、システム上でのメモリ利用度を上げるよう利用され得る。

【００２７】

論理パーティション３０３、３０５、３０７、および３０９はさらに、パーティション・ファームウェア・ローダ３１１、３１３、３１５、および３１７を含む。パーティション・ファームウェア・ローダ３１１、３１３、３１５、および３１７は、ＩＰＬまたは初期ブートストラップ・コード、ＩＥＥＥ－１２７５標準Ｏｐｅｎ　Ｆｉｒｍｗａｒｅ、およびインターナショナル・ビジネス・マシーンズ・コーポレーションから入手できるランタイム抽象化ソフトウェア（ＲＴＡＳ：ｒｕｎｔｉｍｅ　ａｂｓｔｒａｃｔｉｏｎ　ｓｏｆｔｗａｒｅ）を使用して実装され得る。

【００２８】

論理パーティション３０３、３０５、３０７、および３０９がインスタンス化されると、ブートストラップ・コードのコピーが、仮想マシン・モニタ３１０によって論理パーティション３０３、３０５、３０７、および３０９内へロードされる。その後、ブートストラップ・コードに制御が移され、次にブートストラップ・コードは、ｏｐｅｎ　ｆｉｒｍｗａｒｅおよびＲＴＡＳをロードする。続いて、論理パーティション３０３、３０５、３０７、および３０９に関連する、または割り振られているプロセッサが、論理パーティション・ファームウェアを実行するよう論理パーティションのメモリにディスパッチされる

。
【００２９】
分割されたハードウェア３３０は、複数のプロセッサ３３２～３３８、複数のシステム・メモリ・ユニット３４０～３４６、複数の入出力（Ｉ／Ｏ：ｉｎｐｕｔ／ｏｕｔｐｕｔ）アダプタ３４８～３６２、およびストレージ・ユニット３７０を含む。プロセッサ３３２～３３８、メモリ・ユニット３４０～３４６、ＮＶＲＡＭストレージ３９８、およびＩ／Ｏアダプタ３４８～３６２はそれぞれ、論理分割されたプラットフォーム３００内の複数の論理パーティション３０３、３０５、３０７、および３０９のうちの１つに割り振られるとよく、それぞれがオペレーティング・システム３０２、３０４、３０６、および３０８のうちの１つに対応する。
【００３０】
仮想マシン・モニタ３１０は、論理パーティション３０３、３０５、３０７、および３０９に関するいくつかの機能およびサービスを実行して、論理分割されたプラットフォーム３００の分割をもたらし、実行する。仮想マシン・モニタ３１０は、基礎をなすハードウェアと等しい、ファームウェアで実装される仮想マシンである。したがって、仮想マシン・モニタ３１０は、独立したＯＳイメージ３０２、３０４、３０６、および３０８の同時実行を、論理分割されたプラットフォーム３００のすべてのハードウェア・リソースを仮想化することによって可能にする。
【００３１】
サービス・プロセッサ３９０は、論理パーティション３０３、３０５、３０７、および３０９におけるプラットフォーム・エラーの処理などの様々なサービスを提供するために使用され得る。サービス・プロセッサ３９０はさらに、インターナショナル・ビジネス・マシーンズ・コーポレーションなどのベンダにエラーのレポートを返すサービス・エージェントとしての機能を果たしてもよい。種々の論理パーティションの動作は、ハードウェア・システム・コンソール３８０を介して制御され得る。ハードウェア・システム・コンソール３８０は、別個のデータ処理システムであり、このシステムからシステム管理者は、リソースを異なる論理パーティションに再割り当てすることを含む、様々な機能を実行することもできる。
【００３２】
例示実施形態は、通常動作を実現するために最低限必要なリソースを少し上回るリソースを用いて、単一のキュー・ペア（ＱＰ）、すなわち送信／受信ペアのみを最初に割り当てるよう、オペレーティング・システム（ＯＳ）ドライバを規定する。トラフィック・フローまたはワークロードが所定の閾値超に増えるのに従って、ＯＳドライバは、必要に応じて追加のＱＰを動的に割り当てる。ＱＰが実行中のＯＳドライバに追加され、アダプタに対して利用可能になるため、トラフィックは、イングレス、すなわち受信パス、およびイーグレス、すなわち送信パスの両方において、より多くの中央処理ユニット（ＣＰＵ）にハッシュされ、パフォーマンスおよびＣＰＵ／メモリ・リソース使用量が効果的に拡大される。パフォーマンス拡大は、静的なディスクリプタ・カウントではなく追加のＱＰによって実現されるため、システム・リソースは、必要なくなれば削減されることもできる。トラフィック・フローおよびワークロードが、規定の最小閾値未満に下がるのに従って、ＯＳドライバは、ＱＰを除去し、通常動作の最小限のリソース利用まで再び低下するとよい。ワークロードまたはトラフィックが増えると、このサイクルが繰り返される。したがって、例示実施形態は、ＣＰＵ利用およびアクティブ・メモリ・シェアリングのアフィニティを提供しながら、パフォーマンスのために調整する動的再構成を効果的に実現する。
【００３３】
図４は、例示実施形態による、主要な動作コンポーネントおよびその相互作用を示す例示のブロック図である。図４に示されている構成要素は、ハードウェア、ソフトウェア、またはハードウェアおよびソフトウェアの任意の組み合わせにおいて実装され得る。一例示実施形態では、図４の構成要素は、１つ以上のデータ処理デバイスまたはシステムの１

つ以上のプロセッサ上で実行されているソフトウェアとして実装される。

【００３４】

図４は、例示実施形態による、動的再構成によってマルチキュー・ネットワーク・アダプタのリソース・アフィニティを提供するメカニズムの例示的な実装を示す。データ処理システム４００は、オペレーティング・システム４０４内にデバイス・ドライバ４０２を含む。デバイス・ドライバ４０２は、アプリケーション４１０およびネットワーク・アダプタ４１２が使用するメモリ４０８内に１つ以上のキュー・ペア４０６ａ～４０６ｎを提供する。例示実施形態は、１つのデバイス・ドライバ４０２、１つのアプリケーション４１０、および１つのネットワーク・アダプタ４１２のみを示すが、当業者には当然のことながら、データ処理システム４００は、複数のデバイス・ドライバ、複数のアプリケーション、および複数のネットワーク・アダプタをデータ処理システム４００内に含むこともできる。

【００３５】

オペレーティング・システム４０４の初期化時、オペレーティング・システム４０４は、デバイス・ドライバ４０２を構成および初期化する。次に、デバイス・ドライバ４０２は、メモリ４０８内に、いくつかの受信ディスクリプタ／バッファおよびいくつかの送信ディスクリプタ／バッファを含むとよい初期キュー・ペア４０６ａを割り当てる。続いてデバイス・ドライバ４０２は、トラフィックがネットワーク・アダプタ４１２に送られることができるように、ネットワーク・アダプタ４１２を起動する。動作中、キュー・ペア４０６ａのみが初期化されているとき、デバイス・ドライバ４０２は、ネットワーク・アダプタ４１２へ送信されるトラフィックを受信し、トラフィックを、配信のためにネットワーク・アダプタ４１２上へ送る。続いてネットワーク・アダプタ４１２は、送信トラフィックを、サーバ４１６、サーバ４１８、クライアント４２０、クライアント４２２、または同様のものなどのデバイス上へ、ネットワーク４２４を経由して送る。反対に、ネットワーク・アダプタ４１２が、サーバ４１６、サーバ４１８、クライアント４２０、クライアント４２２、または同様のものなどのデバイスからデバイス・ドライバ４０２にネットワーク４２４を経由して送信されるトラフィックを受信すると、ネットワーク・アダプタ４１２は、オペレーティング・システム４０４による処理の割り込みを起こし、トラフィックは、デバイス・ドライバ４０２に送られる。キュー・ペア４０６ａ、および下記のようにさらに割り当てられるとよい４０６ｂ～４０６ｎはそれぞれ、それ自体の、関連する受信割り込みを有する。

【００３６】

デバイス・ドライバ４０２がキュー・ペア４０６ａを割り当ててインスタンス化した後、デバイス・ドライバ４０２は、データ・フローおよびリソース利用可能性を介して、オペレーティング・システム４０４のワークロードを継続的に監視する。デバイス・ドライバ４０２は、送信／受信バイト・パー・セカンド、ネットワーク・アダプタ４１２によって送信および受信されているフロー制御フレームの数、ネットワーク・アダプタ４１２によって検出されるＤＭＡオーバーラン（ＤＭＡ　ｏｖｅｒｒｕｎ）の数、デバイス・ドライバ４０２によって検出される送信タイムアウト・イベントの数、割り込み毎にデバイス・ドライバ４０２によって処理される受信パケットの数、ソフトウェア・キュー上の送信パケットの数、または同様のものなどのパラメータを監視するとよい。デバイス・ドライバ４０２が、所定の高閾値を上回ることなどによって、ワークロード監視によりリソース不足の状況を検出すると、デバイス・ドライバ４０２は、キュー・ペア４０６ｂ～４０６ｎのうちの追加のキュー・ペアを動的に割り当てて初期化する。続いてデバイス・ドライバ４０２は、ネットワーク・アダプタ４１２内の受信側スケーリング（ＲＳＳ）メカニズム４１４をプログラムし、キュー・ペア４０６ａおよびキュー・ペア４０６ｂ～４０６ｎのうちの追加のキュー・ペアに関連する追加の処理エンジンの動的挿入を可能にする。続いてデバイス・ドライバ４０２は、キュー・ペア４０６ａおよびキュー・ペア４０６ｂ～４０６ｎのうちの追加のキュー・ペアへの、送信タプル・ハッシングを有効化する。ネットワーク・アダプタ４１２による受信タプル・ハッシングおよび処理は、ＲＳＳメカニズ

ム４１４をプログラムすることにより自動的に有効化される。デバイス・ドライバ４０２は、キュー・ペア４０６ｂ～４０６ｎのうちの利用可能なキュー・ペアがすべて消費されるまで、またはキュー・ペア４０６ａ～４０６ｎがデータ処理システム４００内の中央処理ユニットの数を上回るまで、キュー・ペア４０６ｂ～４０６ｎのうちのキュー・ペアを、まだ割り当ておよび初期化されていなければ、ワークロードによる必要に応じて追加し続ける。デバイス・ドライバ４０２はさらに、ネットワーク・アダプタ４１２内のＲＳＳメカニズム４１４を再プログラムし、キュー・ペア４０６ｂ～４０６ｎのうちの新たなキュー・ペアが追加されるたびに、追加の処理エンジンの動的挿入を可能にし、さらにキュー・ペア４０６ｂ～４０６ｎのうちの新たなキュー・ペアへの送信タプル・ハッシングを有効化する。

【００３７】

デバイス・ドライバ４０２が、ワークロードが所定の低閾値を下回ることなどによってワークロードの減少を認識すると、デバイス・ドライバ４０２は、ネットワーク・アダプタ４１２内のＲＳＳメカニズム４１４を動的に再プログラムし、キュー・ペア４０６ａ～４０６ｎのうちの割り当てられているキュー・ペアの削除を可能にする。続いてデバイス・ドライバ４０２は、キュー・ペア４０６ａ～４０６ｎのうちの削除されたキュー・ペアへの送信タプル・ハッシングを無効化する。キュー・ペア４０６ａ～４０６ｎのうちの削除されたキュー・ペアが休止する（ｑｕｉｅｓｃｅ）と、デバイス・ドライバ４０２は、キュー・ペア４０６ａ～４０６ｎのうちの削除されたキュー・ペアを除去し、その結果、キュー・ペア４０６ａ～４０６ｎのうちの削除されたキュー・ペアによって使用されたメモリが解放される。ネットワーク・アダプタ４１２における受信タプル・ハッシングの有効化のように、ネットワーク・アダプタ４１２による受信タプル・ハッシングおよび処理は、ＲＳＳメカニズム４１４を再プログラムすることにより自動的に無効化される。

【００３８】

したがって、例示実施形態は、基礎をなすハードウェアの動的再構成によって、アクティブ・メモリ・シェアリング（ＡＭＳ）および中央処理ユニット（ＣＰＵ）利用のためのリソース・アフィニティを提供するメカニズムを提供し、パフォーマンスまたはサービスの中断なしに、変化するワークロードの要求に応じる。トラフィック・フローまたはワークロードが所定の閾値超に増えるのに従って、デバイス・ドライバは、必要に応じて追加のキュー・ペアを動的に割り当てる。トラフィック・フローおよびワークロードが所定の最小閾値未満に下がるのに従って、デバイス・ドライバは、キュー・ペアを除去し、通常動作の最小限のリソース利用まで再び低下するとよい。ワークロードまたはトラフィックが増加し、続いて減少すると、このサイクルが繰り返される。

【００３９】

当業者であれば当然のことであるが、本発明は、システム、方法、またはコンピュータ・プログラム製品として具現化され得る。したがって、本発明の側面は、完全にハードウェアの実施形態、完全にソフトウェアの実施形態（ファームウェア、常駐ソフトウェア、マイクロコードなどを含む）、または本願明細書においてすべて概して「回路」、「モジュール」もしくは「システム」と呼ばれ得る、ソフトウェアおよびハードウェアの側面を兼ね備えた実施形態の形態をとり得る。さらに、本発明の側面は、コンピュータ使用可能プログラム・コードが具現化された任意の１つ以上のコンピュータ可読媒体（単数または複数）において具現化されたコンピュータ・プログラム製品の形態をとることもできる。

【００４０】

１つ以上のコンピュータ可読媒体（単数または複数）の任意の組み合わせが利用され得る。コンピュータ可読媒体は、コンピュータ可読信号媒体またはコンピュータ可読ストレージ媒体とされ得る。コンピュータ可読ストレージ媒体は、例えば、限定はされないが、電子、磁気、光学、電磁気、赤外線、もしくは半導体のシステム、装置、デバイス、または前述のものの任意の適切な組み合わせとされ得る。コンピュータ可読媒体のより具体的な例（包括的でないリスト）には、１つ以上のワイヤを有する電気的接続、ポータブル・コンピュータ・ディスケット、ハード・ディスク、ランダム・アクセス・メモリ（ＲＡＭ

）、読み取り専用メモリ（ＲＯＭ）、消去可能プログラム可能読み取り専用メモリ（ＥＰＲＯＭまたはフラッシュ・メモリ）、光ファイバ、ポータブル・コンパクト・ディスク読み取り専用メモリ（ＣＤＲＯＭ：ｃｏｍｐａｃｔ　ｄｉｓｃ　ｒｅａｄ－ｏｎｌｙ　ｍｅｍｏｒｙ）、光学式ストレージ・デバイス、磁気ストレージ・デバイス、または前述のものの任意の適切な組み合わせが含まれるであろう。この文書の文脈では、コンピュータ可読ストレージ媒体は、命令実行システム、装置、もしくはデバイスによって、またはそれに関連して使用されるプログラムを含むこと、または記憶することができる任意の有形の媒体とされ得る。

【００４１】

コンピュータ可読信号媒体は、例えば、ベースバンドに、または搬送波の一部として、コンピュータ可読プログラム・コードが具現化された伝播データ信号を含み得る。そのような伝播信号は、電磁気、光学、またはその任意の適切な組み合わせを含むがこれらに限定はされない、様々な形態のいずれかをとってよい。コンピュータ可読信号媒体は、コンピュータ可読ストレージ媒体でなく、命令実行システム、装置もしくはデバイスによって、またはそれに関連して使用されるプログラムの伝達、伝播、または搬送をすることができる、任意のコンピュータ可読媒体としてよい。

【００４２】

コンピュータ可読媒体上に具現化されたコンピュータ・コードは、無線、有線、光ファイバ・ケーブル、無線周波数（ＲＦ：ｒａｄｉｏ　ｆｒｅｑｕｅｎｃｙ）など、またはその任意の適切な組み合わせを含むがこれらに限定はされない、任意の適切な媒体を使用して送られてもよい。

【００４３】

本発明の側面の動作を実行するコンピュータ・プログラム・コードは、Ｊａｖａ（Ｒ）、Ｓｍａｌｌｔａｌｋ（Ｒ）、Ｃ＋＋または同様のものなどのオブジェクト指向プログラミング言語、および「Ｃ」プログラミング言語もしくは同様のプログラミング言語などの従来の手続きプログラミング言語を含む、１つ以上のプログラミング言語の任意の組み合わせで書かれていてよい。プログラム・コードは、スタンド・アロン・ソフトウェア・パッケージとして、完全にユーザのコンピュータ上で実行されることも、部分的にユーザのコンピュータ上で実行されることも、または部分的にユーザのコンピュータ上で、かつ部分的にリモート・コンピュータ上で実行されることも、または完全にリモート・コンピュータもしくはサーバ上で実行されることもできる。後者のシナリオでは、ローカル・エリア・ネットワーク（ＬＡＮ）もしくは広域ネットワーク（ＷＡＮ）を含む任意の種類のネットワークを介してリモート・コンピュータがユーザのコンピュータに接続されてもよく、または、（例えば、インターネット・サービス・プロバイダを使用しインターネットを介して）外部コンピュータに接続されてもよい。

【００４４】

本発明の側面について、本発明の例示実施形態による方法、装置（システム）およびコンピュータ・プログラム製品のフローチャート図もしくはブロック図またはその両方を参照して以下に記載する。当然のことながら、フローチャート図もしくはブロック図またはその両方の各ブロック、およびフローチャート図もしくはブロック図またはその両方の複数ブロックの組み合わせは、コンピュータ・プログラム命令により実装可能である。マシンを生じるよう、こうしたコンピュータ・プログラム命令が、汎用コンピュータ、専用コンピュータ、またはその他のプログラム可能データ処理装置のプロセッサに提供されて、この命令が、コンピュータまたはその他のプログラム可能データ処理装置のプロセッサにより実行されて、フローチャートもしくはブロック図またはその両方のブロックもしくは複数ブロックにおいて指定された機能／動作を実装する手段を作り出すようにすることもできる。

【００４５】

さらに、特定の形で機能するようコンピュータ、その他のプログラム可能データ処理装置、またはその他のデバイスに指示することができるこうしたコンピュータ・プログラム

命令は、コンピュータ可読媒体に記憶されて、コンピュータ可読媒体に記憶されたこの命令が、フローチャートもしくはブロック図またはその両方のブロックもしくは複数ブロックにおいて指定された機能／動作を実装する命令を含む製品を生じるようにすることもできる。

【００４６】

さらに、コンピュータ・プログラム命令は、コンピュータ、その他のプログラム可能データ処理装置、またはその他のデバイスにロードされて、コンピュータ、その他のプログラム可能装置、またはその他のデバイス上で一連の動作ステップが実行されるようにしてコンピュータで実装されるプロセスを生じさせ、コンピュータまたはその他のプログラム可能装置上で実行される命令が、フローチャートもしくはブロック図またはその両方のブロックもしくは複数ブロックにおいて指定された機能／動作を実装するためのプロセスを提供するようにすることもできる。

【００４７】

以下、図５を参照する。この図面は、例示実施形態による、動的再構成によってマルチキュー・ネットワーク・アダプタのリソース・アフィニティを提供する例示の動作の概要を示すフローチャートを提供する。動作が開始するとき、構成および初期化されたデバイス・ドライバが、メモリ内に初期キュー・ペアを割り当てる（ステップ５０２）。続いてデバイス・ドライバは、トラフィックがネットワーク・アダプタに送られることができるように、ネットワーク・アダプタを起動する（ステップ５０４）。

【００４８】

デバイス・ドライバが、キュー・ペアを割り当ててインスタンス化した後、デバイス・ドライバは、データ・フローおよびリソース利用可能性を介して、オペレーティング・システムのワークロードを継続的に監視する（ステップ５０６）。デバイス・ドライバは、送信／受信バイト・パー・セカンド、ネットワーク・アダプタによって送信および受信されているフロー制御フレームの数、ネットワーク・アダプタによって検出されるＤＭＡオーバーランの数、デバイス・ドライバによって検出される送信タイムアウト・イベントの数、割り込み毎にデバイス・ドライバによって処理される受信パケットの数、ソフトウェア・キュー上の送信パケットの数、または同様のものなどのパラメータを監視するとよい。続いてデバイス・ドライバは、リソース不足状況を示す、所定の高閾値の超過が生じたかどうかを、ワークロード監視により判断する（ステップ５０８）。ステップ５０８にて、ワークロードが所定の高閾値より高くなっていれば、デバイス・ドライバは、メモリ内に追加のキュー・ペアを動的に割り当てて初期化する（ステップ５１０）。続いてデバイス・ドライバは、ネットワーク・アダプタ内のＲＳＳメカニズムをプログラム／再プログラムして追加の処理エンジンの動的挿入を可能にし（ステップ５１２）、デバイス・ドライバは、新たに割り当てられたキュー・ペアへの送信タプル・ハッシングを有効化し（ステップ５１４）、動作はその後、ステップ５０６に戻る。

【００４９】

ステップ５０８にて、ワークロードが所定の高閾値より高くなっていなければ、デバイス・ドライバは、ワークロードが所定の低閾値を下回ったかどうかを判断する（ステップ５１６）。ステップ５１６にて、デバイス・ドライバが、ワークロードは所定の低閾値を下回っていないと判断すると、動作はステップ５０６に戻る。ステップ５１６にて、デバイス・ドライバが、ワークロードが所定の低閾値を下回ったと判断すると、デバイス・ドライバは、割り当てられているキュー・ペアが残り１つのみあるかどうかを判断する（ステップ５１８）。ステップ５１８にて、デバイス・ドライバが、１つのみのキュー・ペアが残っていると判断すると、動作はステップ５０６に戻る。ステップ５１８にて、デバイス・ドライバが、２つ以上のキュー・ペアが残っていると判断すると、デバイス・ドライバは、ネットワーク・アダプタ内のＲＳＳメカニズムを動的に再プログラムして、割り当てられているキュー・ペアの削除を可能にするとよい（ステップ５２０）。続いてデバイス・ドライバは、特定されたキュー・ペアへの送信タプル・ハッシングを無効化する（ステップ５２２）。デバイス・ドライバは続いて、特定されたキュー・ペタへのワークロー

ドが休止したかどうかを判断する（ステップ５２４）。ステップ５２４にて、デバイス・ドライバが、特定されたキュー・ペアへのワークロードが休止していないと判断すると、動作はステップ５２４に戻る。ステップ５２４にて、デバイス・ドライバが、特定されたキュー・ペアへのワークロードが休止したと判断すると、デバイス・ドライバは、メモリから特定されたキュー・ペアを除去し（ステップ５２６）、その結果、特定されたキュー・ペアによって使用されたメモリが解放される。動作は続いて、ステップ５０６に戻る。

【００５０】

各図面のフローチャートおよびブロック図は、本発明の様々な実施形態によるシステム、方法およびコンピュータ・プログラム製品の考えられる実装のアーキテクチャ、機能性、および動作を示す。この関連で、フローチャートまたはブロック図内の各ブロックは、指定の論理機能（単数または複数）を実装する１つ以上の実行可能命令を含むモジュール、セグメント、またはコードの一部を表すこともできる。なお、さらに、いくつかの代わりの実装では、ブロック内に示されている機能が、図面に示されているのとは異なる順序で生じてもよい。例えば、関連する機能性次第で、連続して示されている２つのブロックが実際には事実上同時に実行されてもよく、または、各ブロックが逆順で実行されることがあってもよい。なお、さらに、ブロック図もしくはフローチャート図またはその両方の各ブロック、およびブロック図もしくはフローチャート図またはその両方の複数ブロックの組み合わせは、指定の機能もしくは動作を実行する専用ハードウェア・ベース・システム、または専用ハードウェアおよびコンピュータ命令の組み合わせにより実装することができる。

【００５１】

したがって、例示実施形態は、基礎をなすハードウェアの動的再構成によってアクティブ・メモリ・シェアリング（ＡＭＳ）および中央処理ユニット（ＣＰＵ）利用のためのリソース・アフィニティを提供するメカニズムを提供し、パフォーマンスまたはサービスの中断なしに、変化するワークロードの要求に応じる。トラフィック・フローまたはワークロードが、所定の閾値超に増えるのに従って、デバイス・ドライバは、必要に応じて追加のキュー・ペアを動的に割り当てる。トラフィック・フローおよびワークロードが、所定の最小閾値未満に下がるのに従って、デバイス・ドライバは、キュー・ペアを除去し、通常動作の最小限のリソース利用まで再び低下するとよい。ワークロードまたはトラフィックが増加し、続いて減少すると、このサイクルが繰り返される。

【００５２】

上述のとおり、当然のことながら、例示実施形態は、完全にハードウェアの実施形態、完全にソフトウェアの実施形態、またはハードウェアおよびソフトウェア両方の構成要素を含む実施形態の形態をとり得る。例示の一実施形態では、例示実施形態のメカニズムは、ファームウェア、常駐ソフトウェア、マイクロコードなどを含むがこれらに限定されないソフトウェアまたはプログラム・コードにおいて実装される。

【００５３】

プログラム・コードの記憶もしくは実行またはその両方に適したデータ処理システムは、システム・バスを介してメモリ要素に直接または間接的に結合された少なくとも１つのプロセッサを含む。メモリ要素は、プログラム・コードを実際に実行する間に用いられるローカル・メモリ、大容量ストレージ、および、実行中にコードが大容量ストレージから読み出されなければならない回数を減らすために少なくとも一部のプログラム・コードの一時的なストレージとなるキャッシュ・メモリを含むことができる。

【００５４】

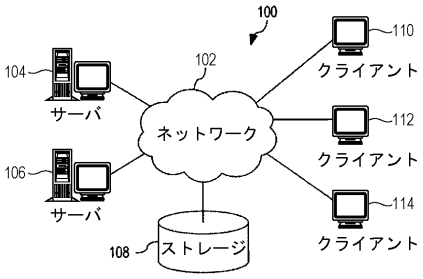
入出力、すなわちＩ／Ｏデバイス（限定はされないが、キーボード、ディスプレイ、ポインティング・デバイスなどを含む）は、直接、または介在するＩ／Ｏコントローラを介して、システムに結合されることが可能である。ネットワーク・アダプタもシステムに結合されて、データ処理システムが、他のデータ処理システムまたはリモート・プリンタまたはストレージ・デバイスに、介在するプライベートまたはパブリック・ネットワークを介して結合された状態となることを可能にしてもよい。モデム、ケーブル・モデム、およ

びイーサネット（Ｒ）・カードが、現在利用可能なタイプのネットワーク・アダプタのごく一部である。

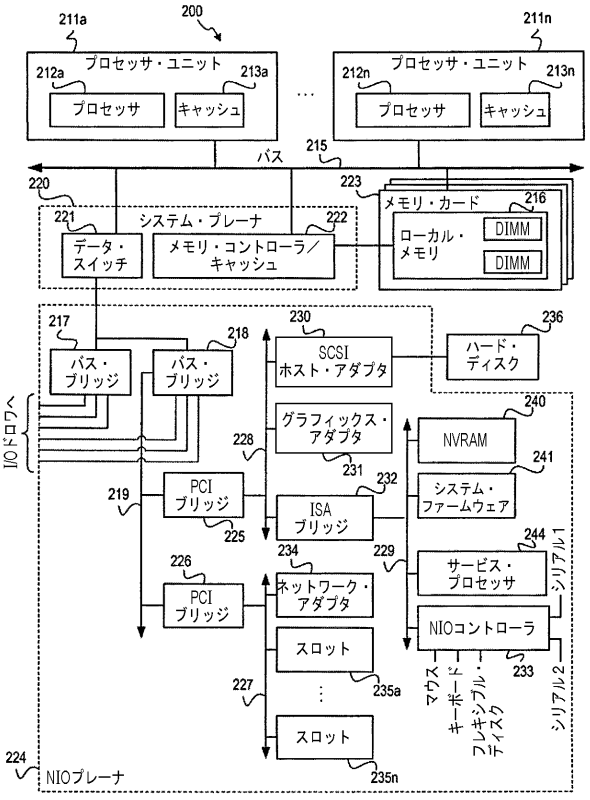
【００５５】

本発明の記載は、例示および説明のために示されたものであり、包括的であることも、開示された形態の発明に限定されることも目的としていない。当業者には、多数の変更および変形が明らかであろう。実施形態は、本発明の原理、実際の応用を最もよく説明して、当業者が、意図される特定の用途に適する様々な変更を用いた様々な実施形態に関して、本発明を理解できるように選ばれ、記載された。

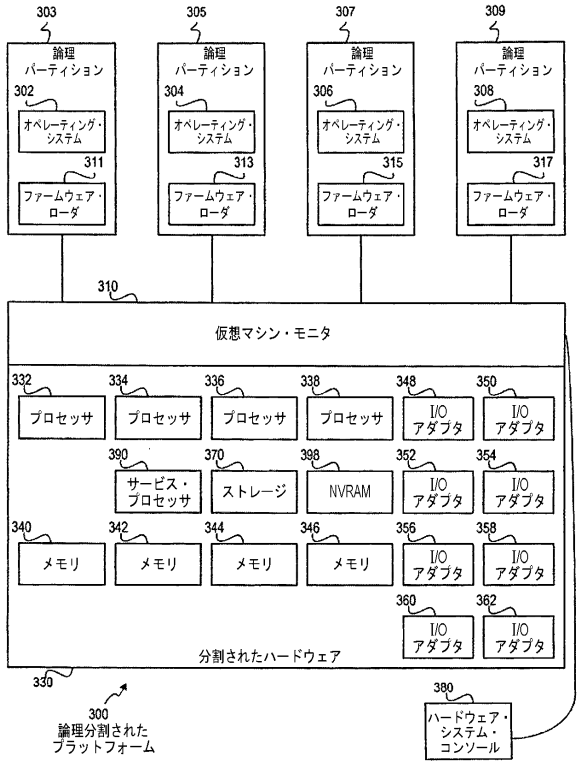
【図１】



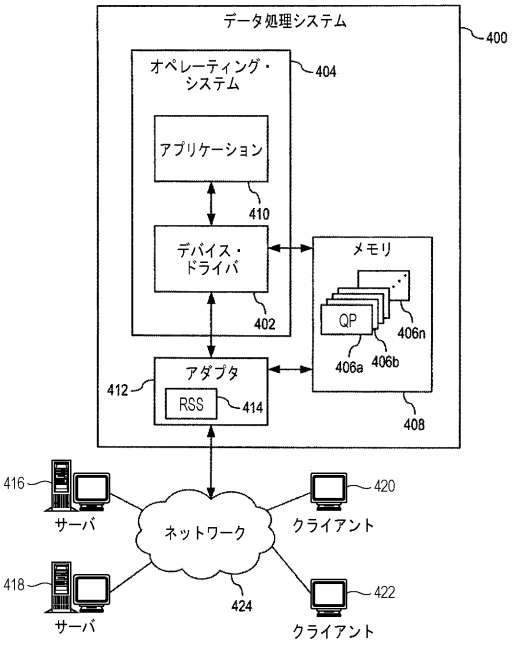
【図２】



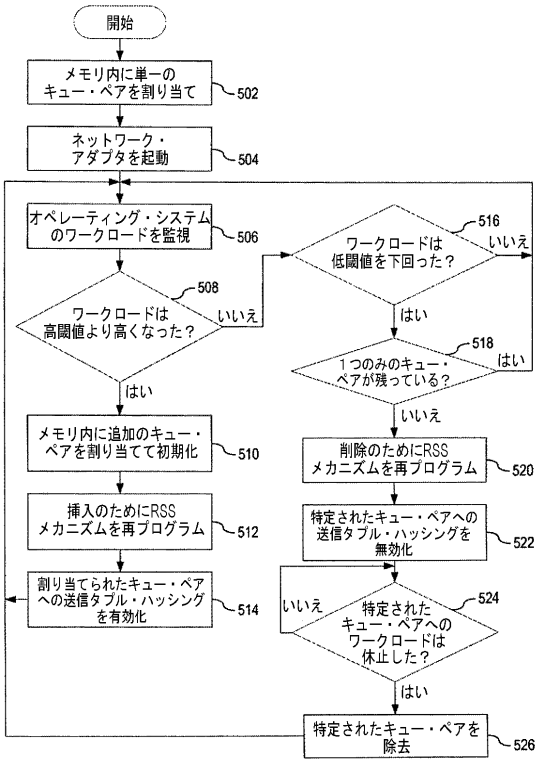
【図３】


303
論理パーティション
302
オペレーティング・システム
311
ファームウェア・ローダ
305
論理パーティション
304
オペレーティング・システム
313
ファームウェア・ローダ
307
論理パーティション
306
オペレーティング・システム
315
ファームウェア・ローダ
309
論理パーティション
308
オペレーティング・システム
317
ファームウェア・ローダ
310
仮想マシン・モニタ
332 プロセッサ
334 プロセッサ
336 プロセッサ
338 プロセッサ
348 I/Oアダプタ
350 I/Oアダプタ
390 サービス・プロセッサ
370 ストレージ
398 NVRAM
352 I/Oアダプタ
354 I/Oアダプタ
340 メモリ
342 メモリ
344 メモリ
346 メモリ
356 I/Oアダプタ
358 I/Oアダプタ
360 I/Oアダプタ
362 I/Oアダプタ
分割されたハードウェア
330
300
論理分割されたプラットフォーム
380
ハードウェア・システム・コンソール

【図４】


データ処理システム
400
オペレーティング・システム
404
アプリケーション
410
デバイス・ドライバ
402
メモリ
QP
406n
406a
406b
408
アダプタ
412
RSS
414
416
サーバ
ネットワーク
420
クライアント
418
サーバ
424
422
クライアント

【図５】


開始
メモリ内に単一のキュー・ペアを割り当て
502
ネットワーク・アダプタを起動
504
オペレーティング・システムのワークロードを監視
506
ワークロードは高閾値より高くなった？
508
いいえ
はい
メモリ内に追加のキュー・ペアを割り当てて初期化
510
挿入のためにRSSメカニズムを再プログラム
512
割り当てられたキュー・ペアへの送信タプル・ハッシングを有効化
514
ワークロードは低閾値を下回った？
516
いいえ
はい
1つのみのキュー・ペアが残っている？
518
はい
いいえ
削除のためにRSSメカニズムを再プログラム
520
特定されたキュー・ペアへの送信タプル・ハッシングを無効化
522
特定されたキュー・ペアへのワークロードは休止した？
524
いいえ
はい
特定されたキュー・ペアを除去
526

フロントページの続き

(72)発明者　カルドナ、オマール
アメリカ合衆国７８７５８　テキサス州　オースティン　バーネット　ロード　１１５０１　９０５－５ビー０１７
(72)発明者　オークス、マシュー、ライアン
アメリカ合衆国７８７５８　テキサス州　オースティン　バーネット　ロード　１１５０１　９０５－８ジー０１９
(72)発明者　カニンガム、ジェームス、ブライアン
アメリカ合衆国７８７５８　テキサス州　オースティン　バーネット　ロード　１１５０１　エムディー　９５５１
(72)発明者　シャルマ、ラケシュ
アメリカ合衆国７８７５８　テキサス州　オースティン　バーネット　ロード　１１５０１　エムディー　９５５１

審査官　森谷　哲朗

(56)参考文献　米国特許出願公開第２０１１／０１４２０６４（ＵＳ，Ａ１）
米国特許出願公開第２０１１／０１５３９３５（ＵＳ，Ａ１）
国際公開第２００９／０２７３００（WO，Ａ２）
特開２００２－２０２９５９（ＪＰ，Ａ）
特開２００８－０６０７００（ＪＰ，Ａ）
特開２００７－２５７０９７（ＪＰ，Ａ）