(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3589858号
(P3589858)
(45)発行日　平成16年11月17日(2004.11.17)　(24)登録日　平成16年8月27日(2004.8.27)

(51)Int.Cl.[7]　F I
G06F 12/02　G06F 12/02　570K
G06F 12/08　G06F 12/08　513
G06F 15/177　G06F 12/08　525Z
G06F 15/177　682B
G06F 15/177　682J

請求項の数 1　(全 12 頁)

(21)出願番号　特願平10-110274
(22)出願日　平成10年4月21日(1998.4.21)
(65)公開番号　特開平10-326223
(43)公開日　平成10年12月8日(1998.12.8)
審査請求日　平成11年7月5日(1999.7.5)
(31)優先権主張番号　845,306
(32)優先日　平成9年4月25日(1997.4.25)
(33)優先権主張国　米国(US)

前置審査

(73)特許権者　398038580
ヒューレット・パッカード・カンパニー
HEWLETT－PACKARD COMPANY
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　3000
(74)代理人　100087642
弁理士　古谷　聡
(74)代理人　100076680
弁理士　溝部　孝彦
(72)発明者　トニー・エム・ブルワー
アメリカ合衆国テキサス州プラノ　マリナーズ・ドライブ　5225

最終頁に続く

(54)【発明の名称】マイクロカーネルアクセス方法および処理装置エージェント

(57)【特許請求の範囲】
【請求項１】
クロスバーを介してそれぞれが独立したノードに記憶される複数のマイクロカーネルのうちの一つへのメモリ参照をする、改良された処理装置エージェントであって、ノードの各々がマイクロカーネルを該ノード内の、第１のプロトコルによって同定可能な予約アドレス空間に記憶し、前記処理装置エージェントが処理装置からのメモリ参照要求を受け取り、該処理装置から受け取るメモリ参照要求の一つは、マイクロカーネルに対する参照要求であり、全てのマイクロカーネル参照は共通ノードをアドレスし、
前記処理装置により行われたメモリ参照によって指定されたメモリアドレスのビットシーケンスを構文解析して宛先ノードの（１）宛先ノード識別子および（２）宛先ノードアドレスを求める手段と、
前記宛先ノードアドレスが、前記第１のプロトコルによりノードにマイクロカーネルを記憶するための予約されたアドレス空間にあると同定される時には、前記メモリ参照を前記マイクロカーネル参照として認識する手段と、
マイクロカーネル参照の前記宛先ノード識別子が、前記共通のノードを同定しない場合、プログラムエラーを報告する手段と、
マイクロカーネル参照上の宛先ノード識別子を、そのマイクロカーネル参照が参照するマイクロカーネルを記憶する局所ノードの識別子に変換する手段と、
前記処理装置への後続のキャッシュコヒーレンシー要求に応答して、前記キャッシュコヒーレンシー要求を処理する前に局所ノード識別子を前記宛先ノード識別子に変換する手段

と、
を有する処理装置エージェント。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は一般的には汎用コンピュータにおける大域共用メモリの管理に関し、特に、異なるメモリノードにロードされた異なるソフトウエアカーネルを変換テーブルあるいは一時記憶装置を用いることなく複数の処理装置によって同時にアクセスすることを可能にするものである。
【０００２】
【従来の技術】
大域共用メモリを用いた共用メモリシステムでは、一般的にはシステム全体に対して１つのアドレス空間が設けられている。この空間内では、複数の独立したハードウエアノードのそれぞれに対して、物理的メモリ内の固有のアドレス群が割り当てられている。かかる独立したハードウエアノードにはそれぞれそのシステム上で使用されるオペレーティングシステム（ＯＳ）の対応するマイクロカーネルがロードされている。
【０００３】
マイクロカーネルはシステム内のＯＳの最低位のレベルとみなすことができる。マイクロカーネルは仮想メモリ全体を実行するアプリケーションであるため、物理的メモリにロードされ、実行される。物理的メモリは、上述したようなマップを持たず、それぞれのノードに固有のアドレス空間を有する明確に同定されたメモリ領域である。それぞれのマイクロカーネルは正規のプログラムとしてコンパイルされ、従ってメモリへの参照は全て物理的メモリ上の固有アドレス空間内の絶対アドレスに対して行なわれる。従って、マイクロカーネルをメモリにロードする場合には、マイクロカーネルにコンパイルされた正確なメモリアドレスに対して行なわれなければならない。
【０００４】
マイクロカーネルを用いた再配置可能なアドレス指定を用いる方法は当該技術分野において周知であることに注意されたい。しかし、この方法は最適というにはほど遠く、従って一般的には用いられない。この方法は非標準実行モードを起動するが、このモードは処理時間がかかり、一般に効率が低い。更に、この方法を用いる場合には、安定性の低いコンパイラ戦略が必要になり、実行時にソフトウエアエラーをさらに発生させる可能性がある。従って、マイクロカーネルを処理する際には再配置可能でないメモリ参照（すなわち、絶対的アドレス指定）を用いることが好適である。
【０００５】
ＯＳ上でランするほとんどのアプリケーションはこのようなフレキシビリティの欠如に耐えるものである必要はない。これはそれぞれのアプリケーションが自らの仮想アドレス空間を備え、従ってほとんどあらゆるアドレスがそのアプリケーションの特定のメモリ参照に対処し得るようにマップすることができる。
【０００６】
マイクロカーネルは、しかしながら、この機能を持つように配置されないという点で変則的なものである。マイクロカーネルが物理的アドレス空間にロードされるときにはマップがなく、従って指定された実アドレスにロードしなければならない。また、あるシステムの物理的アドレス空間は、それぞれのノードに対する一連の固有のアドレスからなるため、異なるノードにロードされたマイクロカーネルをいくつかの処理装置間で共用することが望ましい場合にメモリ管理上の問題が発生する。
【０００７】
従って、それぞれのノードが自らの物理的メモリ領域を有し、そのアドレス空間を複数のマイクロカーネルによってコンパイルすることのできる機構が必要となってきている。同時に、それらを合わせた全アドレススペースはユーザーアプリケーション用の大域共用メモリシステムを可能とするように協働するものでなければならない。

【０００８】
当該技術分野の現行のシステムでは、煩雑な変換テーブルおよび一時記憶レジスタを用いて複数のマイクロカーネルによって物理的メモリアドレス空間をコンパイルすることが可能である。追加の記憶レジスタを用いたこれらのテーブルの処理は処理時間を著しく浪費する。
【０００９】
【発明が解決しようとする課題】
従って、当該技術分野において、マイクロカーネルが一時記憶や処理時間の浪費を生じることなくしかもキャッシュコヒーレントな環境で、物理的アドレス空間を「オンザフライ」で同時に共用するように現われることを可能にすることが必要とされている。
【００１０】
【課題を解決するための手段】
本発明によれば、上記の目的・特徴および改善、その他の目的・特徴および改善は、処理装置がメモリ要求を行なってマイクロカーネルがロードされた所定の物理的メモリ領域にアクセスするときこれを検出するように配設された機構によって達成される。かかるアクセスはそのマイクロカーネルがロードされたノードの固有アドレス範囲にマップされる。この同定マッピングおよびルート指定処理は、アクセス対象であるアドレス内のビットの状態を分析することによって「オンザフライ」で実行可能である。更に、他の処理装置からの参照がその情報を同時に返すか又はマップされた領域からそのマイクロカーネルにアクセスしようとするかするとき、その情報は「ノードゼロ」にマップし戻され、キャッシュコヒーレントな態様での処理が可能となる。
【００１１】
本発明をより詳細に説明するために、マイクロカーネルがロードされていると予想されるメモリ領域を、処理装置が参照するものと仮定する。かかる領域は通常、６４ＭＢのセグメントに分割されたノード内の最初の１６ＭＢのサブセグメントあるいは５番目の１６ＭＢのサブセグメントのいずれかである。本発明では、かかる処理装置による参照が「オンザフライ」で検出され、所望のマイクロカーネルがロードされたノードのメモリの最初の１６ＭＢのサブセグメントおよび５番目の１６ＭＢのサブセグメントにマップされる。これによって、それぞれのノードの処理装置はその処理装置のノードにロードされたマイクロカーネルにアクセスすることができる。
【００１２】
この原理は他のノードあるいは他の処理装置がそのノード上でキャッシュコヒーレントな環境で動作し得るように拡張される。本発明では、本発明に従ってマップされたメモリ領域をキャッシュに入れた第１の処理装置に対して第２の処理装置から送られたコヒーレンシートラフィックを検出する。このコヒーレンシートラフィックは、第１の処理装置に送り戻されて無効化又はフラッシュ態様の操作をされる前に、第１の処理装置が使用した元の物理メモリアドレス空間に自動的にマップし戻される。これによって、第２の処理装置は本発明に従ってマップされたアドレス空間をそのキャッシュ内にある空間として同定することができ、その結果そのノード内でキャッシュコヒーレントな動作が可能であり、同時にノード単位でマイクロカーネルをロードする機能は維持される。
【００１３】
従って、本発明の技術的利点は、処理装置が変換テーブルやそれに関係する中間的な記憶を必要とすることなく複数のマイクロカーネルに共用的にアクセスすることが可能であることである。これによって、共用環境におけるマイクロカーネル参照の処理が最適化される。
【００１４】
本発明の他の技術的利点は、前記アクセスをキャッシュコヒーレントな環境で実行可能であることである。
【００１５】
以上は後の本発明の詳細の詳細な説明の理解を助けるためにその特徴および技術的利点を

かなり概括的に説明したものである。本発明のこれ以外の特徴および利点もまた以下に説明され、それらは本発明の特許請求の範囲の対象となる。当業者にはここに説明する概念および具体的実施形態は本発明の目的を達するための変更あるいは他の構造の設計の基礎として容易に利用可能であることが理解されよう。また、当業者にはかかる均等な構造は特許請求の範囲に定める本発明の要旨および範囲から逸脱するものではないことは明らかであろう。

【００１６】

【発明の実施の形態】

図１には、ノード識別子（「ノードＩＤ」）００および０１を備えた２つの対応するノードに割り当てられた、２つのメモリ構造００および０１を備えた物理メモリ１０の一部分を示す。それぞれのノードは固有のアドレス範囲を有する。

【００１７】

ここに説明する実施形態においては、それぞれのノードはそれぞれが４つの１６ＭＢのサブセグメントを有する６４ＭＢのセグメントに分割されるが、本発明は異なる構成のメモリ構造にも適用可能であることは明らかである。図１において、メモリ構造００の最初の２つの６４ＭＢセグメントを１０１および１０２で示し、メモリ構造０１の最初の２つの６４ＭＢセグメントを１０３および１０４で示している。それぞれの６４ＭＢセグメントはそれぞれが１６ＭＢのサブセグメントを有する４つの下位区分に分割される。セグメント１０１についてはこれらの行は１０１－１、１０１－２、１０１－３および１０１－４で示されている。セグメント１０２についてはこれらの行は１０２－１、１０２－２、１０２－３および１０２－４で示されており、以下同様である。それぞれの行は４０ビット参照に基づく一連の独自にアドレス指定される記憶場所を含む。たとえば、行１０１－１は（１６進数すなわち「Ｈｅｘ」で表わした場合）アドレス００　００００００００から００　００ＦＦＦＦＦＦを備え、行１０２－１はアドレス００　０４０００００００から００　０４ＦＦＦＦＦＦを備え、行１０３－１はアドレス０８　０００００００００から０８　００ＦＦＦＦＦＦを備え、以下同様である。

【００１８】

ノードＩＤと図１に示す物理メモリアドレスとの間の関係を、図２と更に下記の表１を参照して説明する。しかし、以下の関係は例に過ぎず、本発明は他の態様で構成されたメモリ構造にも同様に適用可能であることをまず再度強調しておかねばならない。図２には、メモリアドレスとして用いられる４０ビットワードのビットレイアウトを示す。ノードＩＤは当該技術分野において通常行なわれるように最上位の５ビット（すなわちビット０ないし４）に配置される。残りの３５ビット（ビット５ないし３９）は、そのノード内の詳細な物理メモリアドレスを、そのアドレスが存在するそのノード内の物理的に連続するメモリの（ノードゼロからの）オフセットを示すことによって同定する。また本実施形態においては１４番目のビット（すなわちビット１３）も、このビットが「１」であるときこれはある特定のメモリアドレスがノードゼロから見て少なくとも５番目の１６ＭＢサブセグメントにある（すなわち、１６ＭＢサブセグメント４つ分だけオフセットされている）ことを示すという点で重要である。これは１４番目のビットはノードゼロからの６４ＭＢのオフセットを表わすためである。

【００１９】

ここで、下の表１を見ると、図２の４０ビットワードは１０文字の１６進（Ｈｅｘ）ワードとして表わすことも可能であることがわかる。また、表１には本発明で用いられるメモリ構造の例をさらに示すために図１が参照されている。すなわち、表１からノードＩＤは最初の５ビット（すなわちビット０ないし４）に配置され、ビット５ないし１２およびビット１４、１５が全て０であるとき、このアドレスはノードの最初あるいは５番目の１６ＭＢサブセグメントのいずれかにある（すなわちオフセットがゼロか、あるいは１６ＭＢサブセグメント４つ分だけオフセットされた１６ＭＢサブセグメントかのどちらかにある）。

【００２０】

【表１】

			初期メモリ位置	
図1の16MBサブセグメント	ノードID	16MBサブセグメントオフセット	Hex（16進）	Binary（2進）
101-1	00	0	00 00000000	0000 0000 0000 0000 …
101-2	00	1	00 01000000	0000 0000 0000 0001 …
101-3	00	2	00 02000000	0000 0000 0000 0010 …
101-4	00	3	00 03000000	0000 0000 0000 0011 …
102-1	00	4	00 04000000	0000 0000 0000 0100 …
103-1	01	0	08 00000000	0000 1000 0000 0000 …
104-1	01	4	08 04000000	0000 1000 0000 0100 …
図示せず	02	0	10 00000000	0001 0000 0000 0000 …
図示せず	02	4	10 04000000	0001 0000 0000 0100 …

【００２１】
更に、マイクロプロセッサ技術においては、メモリノードへのマイクロカーネルのローディングは、従来アドレスノードゼロから開始されていることは理解されよう。従来、マイクロカーネルの固定的に割り当てられた部分の大きさが３２ＭＢを超えることはなく、従って、マイクロカーネルの最初の１６ＭＢはノードの最初の１６ＭＢサブセグメント、すなわちアドレスノードゼロを含む１６ＭＢサブセグメントにロードされる。マイクロカーネルの大きさが１６ＭＢを超えると、その超過部分は通常そのノードの５番目の１６ＭＢサブセグメント（すなわち、そのノード内の２番目の６４ＭＢセグメントの最初の１６ＭＢサブセグメント、すなわち上の表１でいえば、１６ＭＢサブセグメント４つ分だけオフセットされた１６ＭＢサブセグメント）にロードされる。
【００２２】
従って、本発明は、処理装置が参照するアドレスのビットレイアウトの分析によってマイクロカーネルのメモリを参照するものとみなすことができる。ビット５ないし１２およびビット１４、１５が全てゼロである場合、あるノードのオフセットゼロの位置あるいは４つ分オフセットされた位置のいずれかにある１６ＭＢサブセグメントを参照しなければならない。これが、ノード内のマイクロカーネルがロードされる位置である。本発明によれば、その後ノードＩＤ（これはマイクロカーネルの参照であることがすでに判定されているため０でなければならない）が処理装置が現在参照しているノードのノードＩＤに変換される。その後、このマイクロカーネル参照は他の任意の通常の大域共用メモリの参照と同様に取り扱うことができる。
【００２３】
キャッシュのコヒーレンシーは逆マッピングによって実現可能である。第１の処理装置が、マイクロカーネル参照が行なわれている第２の処理装置に、コヒーレンシー要求を送るとき、もし第２の処理装置がそれ以前にこのマイクロカーネル参照のノードＩＤをノード００から現在のノードに変換していると、問題が生ずる。従って、このノードＩＤはコヒーレンシー要求が処理される前に００に変換し戻さなければならない。従って、この場合も、マイクロカーネル参照は上述したビット５ないし１２およびビット１４、１５を分析

することによってコヒーレンシー要求内で認識することができる。参照がマイクロカーネルへの参照として同定されると、処理装置にコヒーレンシー要求を送る前にノードＩＤが００に変換し戻される。

【００２４】

以上の論理は全て当該技術分野で現在実行されているようなテーブルや一時記憶レジスタを用いることなく「オンザフライ」で実行可能であることが理解されよう。処理時間と一時記憶の最適化の可能性があることは自明である。

【００２５】

図１Ａは本発明によって解決される問題の説明を助けるものである。図１Ａにはマイクロカーネルコード内で通常用いられる、コンパイラへの簡単なロードワード命令を示し、この命令はコンパイラに特定のメモリアドレス００　０００００００００の値をレジスタ２にロードするよう命じるものである。この命令の機能はあるノードの最初のメモリアドレス内の値をレジスタ２にロードすることである。しかし、これはマイクロカーネルコードであるため、絶対的メモリアドレスを用いなければならない。ここでさらに図１を見ると、このマイクロカーネルがノード００にロードされる場合、００　０００００００００はそのノードの最初のメモリアドレスの正確なアドレスであるため、システムはこの命令を適正に実行することができる。しかし、このマイクロカーネルがノード０１にロードされる場合、システムはこの命令を適正に実行することはできない。ノード０１の最初のメモリアドレスの正確なアドレスはアドレス０８　０００００００００の行１０３－１にある。しかし、図１Ａにおいてはアドレス００　０００００００００からのロードが必要であり、変換が必要である。従って、多ノードシステムによるマイクロカーネルの適切な共用を可能とするために、本発明ではマイクロカーネルのメモリ参照のノードＩＤアドレスを、処理装置が正確にアドレス指定された自らの物理空間が実際に他のノードのメモリ空間に「透明に」マップされていた場合に、この自らの物理空間内のマイクロカーネルを参照していると「信じる」ように変換する。

【００２６】

図３には本発明において上述したような処理装置からメモリへの要求を可能にするための論理の一例を示す。図４には本発明において上述したような処理装置へのコヒーレンシー要求を可能にするための論理の一例を示す。

【００２７】

まず「メモリへの処理装置要求」と題する図３に転じて、ブロック３０１および３０２でまずメモリ参照のアドレスがチェックされ、ビット５ないし１２およびビット１４、１５が全て値０であるかどうかが調べられる。これらが全て値０である場合、このメモリ参照がノード中の最初の１６ＭＢサブセグメントと５番目の１６ＭＢサブセグメントのどちらへの参照であるかを有効に推論し、それによってそのメモリ参照がマイクロカーネルに対して行なわれていることを示すことができる。

【００２８】

一方、ブロック３０１および３０２におけるチェックの結果がＮＯである場合、マイクロカーネルがロードされたと予想された以外の場所に対して通常の大域共用メモリ参照が行なわれたものと推論される（ブロック３０３）。この場合、通常のメモリマッピング機能がこの参照の要求に応えてそのコードの適正な実行を可能とする。

【００２９】

図３のブロック３０１および３０２に戻って、ビット５ないし１２およびビット１４、１５が全て０である場合、ノードＩＤ変換によって、このメモリ参照は参照対象であるマイクロカーネルがロードされたノードを参照するように強制される（ブロック３０４）。たとえば、図１に戻って、参照対象であるマイクロカーネルがノード０１にロードされている場合、ブロック３０４におけるノードＩＤ変換の結果、そのノードへの対応する参照が強制される。

【００３０】

この強制ノードＩＤ変換は図３においてブロック３０５および３０６に移行することによ

って実行される。まず、アドレスビット０ないし４が全て０であるかどうかがチェックされる。本発明では、この処理段階ではこれらは全て当然０でなければならない。これは、元の参照がマイクロカーネルに対して行なわれたものであることがすでに判定されているためである。実際に、ブロック３０５および３０６でアドレスビット０ないし４の全てが０でない場合、ブロック３０７でエラーが検出され、ソフトウエアの修正が必要であるものと同定される。しかし、これらのビットが全て０であるとすると、ブロック３０８でこれらのビット０ないし４が参照対象であるマイクロカーネルがロードされているノードのＩＤに置き換えられる。これによって、このマイクロカーネルメモリ参照は他の任意の通常の大域共用メモリ参照と同様に処理することができる（ブロック３０９）。

【００３１】

図３のブロック３１０および３１１はこの変換をさらに詳細に示している。ブロック３１０には、ビット５ないし１２およびビット１４、１５が全て０であったために、ブロック３０１および３０２を通過したマイクロカーネルメモリ参照の一例を示している。また、この種のメモリ参照について予想される通り、ビット０ないし４もまた全て０であることが理解されよう。ここでこの参照が図１に示すようなノード０１にロードされたマイクロカーネルへの参照であると仮定する。すると、このノードはこのメモリ参照の参照する正確な物理アドレス００　０００００００００を含んでいない。ブロック３１１に移行して、ブロック３０５、３０６および３０８における変換が行なわれ、ビット０ないし４のノードＩＤがそのマイクロカーネルに対応するノードに置き換えられる。図３に示す例では、これは値０００００１であり、これは１６進数ではそのノード内のオフセットの３つの最上位ビットを含めて、１６進値０８となる。

【００３２】

ここで一旦図１に戻って、この１６進値０８はノード０１内で参照した場合先頭の０８に対応し、図３のメモリ参照が１６ＭＢサブセグメント１０３－１への参照であることを意味する。従って、この変換によってメモリ参照を適正なノードに送られたことになる、この変換を行なわない場合、この参照は図１のノード００に対して行なわれたはずである。この場合おそらくソフトウエアエラーが発生する。

【００３３】

ここで「処理装置へのコヒーレンシー要求」と題する図４を見ると、フローチャートによって本発明に係る第１の処理装置から第２の処理装置へのマイクロカーネル参照を含むコヒーレンシー要求を実行可能とする論理の一例を示す。図４において、ブロック４０１において、ビット５ないし１２およびビット１４、１５が調べられ、ブロック４０２でこれらが全て０であるかどうかがチェックされる。全て０である場合、図３のブロック３０１および３０２を参照して上に説明したように、マイクロカーネルメモリ参照のコヒーレンシー要求と推論することができる。処理はブロック４０４に移行する。一方、ビット５ないし１２およびビット１４、１５が全て０でない場合、通常の大域共用メモリ参照のコヒーレンシー要求と推論することができ、このコヒーレンシー要求は変更を加えることなく第２の処理装置に直接送ることができる（ブロック４０３）。

【００３４】

ビット５ないし１２およびビット１４、１５が全て０であるときブロック４０４に移行して、コヒーレンシー要求を正確に満足するために、図３に示す変換で行なわれたようにノードＩＤ値を全て０に戻さなければならない。ブロック４０４でこの変換が行なわれ、変更された要求が第２の処理装置に送り返される（ブロック４０５）。

【００３５】

図４に示すノードＩＤ値の変更をさらに図４のブロック４１０および４１１の例によって説明する。ブロック４１０は図３のブロック３１１で変換されたメモリ参照を示す。このとき、ノードＩＤは値０００００１を有し、これは１６進値０８を指し、図１の行１０３－１内のアドレスを指定する。ブロック４０４においてこれらのビットはブロック４１１に示すような値０００００に置き換えられ、この値は１６進数値０を表わし、図３における変換の前のマイクロカーネルメモリ参照を反映するものである。従って、このアドレスは

ノード００のアドレスに逆マッピングされ、これは第１の処理装置の予想するところであり、これによってこのコヒーレンシー要求は第１の処理装置のキャッシュ内で適切に実行可能となる。これは、第１の処理装置が最初にノード００アドレスを参照するものとの前提でこの要求を行なったためである。

【００３６】

図５には本発明を実行することのできるアーキテクチャおよびトポロジーの一例を機能レベルで示す。大域共用メモリ環境で動作するほとんどのマイクロプロセッサシステムにおいて、個々の処理装置５０１はメモリアクセスプロトコル５０２を用いて局所メモリノードＬＭＮおよびこれも他の処理装置によって共用される複数（ｎ）の遠隔メモリノードＲＭＮ$_1$ないしＲＭＮ$_n$にアクセスする。

【００３７】

本発明はメモリアクセスプロトコル５０２内で実行することによって効果を上げることができる。これによって、処理装置５０１がマイクロカーネルがロードされていると予想されるメモリ領域へのメモリ要求を行なうと、この事象はマイクロカーネル領域検出およびマッピング機能５０２Ａによって認識される。このマッピング機能が終了し、要求が所望のマイクロカーネルがロードされたノードを参照するように変換されると、位置判定および経路指定機能５０２Ｂがこの要求を局所メモリノードＬＭＮあるいは遠隔メモリノードＲＭＮ$_1$ないしＲＭＮ$_n$のうち適切な方に送る。これによって、マイクロカーネル参照は特定のアドレス空間への参照ではなく、大域共用メモリの一部となる。

【００３８】

図６には当該技術分野において周知のアーキテクチャおよびトポロジーによる本発明の実施態様の一例を示す。図６において、処理装置Ｐ$_1$ないしＰ$_x$は同時に動作し、その過程でメモリ参照を行なう場合がある。かかるメモリ参照は対応する処理装置エージェントＰＡ$_1$ないしＰＡ$_x$を介してまたクロスバー６０１上で行なわれる。処理装置Ｐ$_1$ないしＰ$_x$に利用可能なメモリは大域共用される遠隔メモリ構造６０２および複数（ｙ）の局所メモリ構造ＬＭ$_1$ないしＬＭ$_y$に構成される。全てのメモリへのアクセスはメモリアクセスコントローラＭＡＣ$_1$ないしＭＡＣ$_y$によって制御され、それぞれのメモリアクセスコントローラは対応する局所メモリ空間を管理し、同時に大域共用される遠隔メモリ６０２にアクセスする。

【００３９】

図６の例では、本発明は処理装置エージェントＰＡ$_1$ないしＰＡ$_x$において実施されている。これによって、処理装置Ｐ$_1$ないしＰＡ$_x$がマイクロカーネル参照を行なうと、対応する処理装置エージェントＰＡ$_1$ないしＰＡ$_x$がこの事象を検出し、マッピング、位置判定および経路指定機能を実行して、必要な場合この参照をマイクロカーネルがロードされたノードを参照するように変換する。これによって、マイクロカーネルは局所メモリＬＭ$_1$ないしＬＭ$_y$であるか遠隔メモリ６０２であるかを問わず、このメモリ構造の任意の部分にロードすることができ、従って実際のマイクロカーネル参照が処理装置Ｐ$_1$ないしＰ$_x$からノード０の物理的アドレス空間に対して行なわれる場合であっても、マイクロカーネルは全ての処理装置Ｐ$_1$ないしＰ$_x$によって共用的にアクセス可能となる。

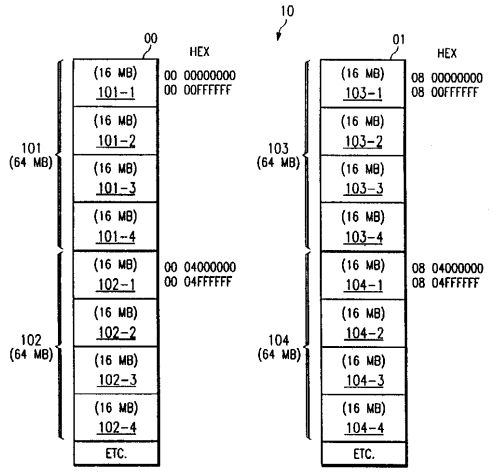
【００４０】

また、キャッシュのコヒーレンシーも実現可能となる。第１の処理装置Ｐ$_1$が処理装置Ｐ$_2$にキャッシュコヒーレンシー要求を行ない、この要求がマイクロカーネル参照を含んでいるとき、処理装置エージェントＰＡ$_1$は上述したように本発明に従ってその参照を変換することはいうまでもない。しかし、本発明は処理装置ＰＡ$_2$にも実施されていることが想起されよう。このコヒーレンシー要求は処理装置エージェントＰＡ$_2$によって同定およびテストされ、マイクロカーネル参照が行なわれるかどうかがチェックされる。マイクロカーネル参照が行なわれる場合、この参照のノードＩＤがノード００に変換し戻されて、処理装置Ｐ$_2$はこのコヒーレンシー要求を解釈することができる。なお、本願において、処理装置エージェントとは、それに接続された処理装置のメモリ参照を処理する装置をいう。

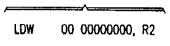
【００４１】
また、図６に加えて、本発明は他の態様のアーキテクチャおよびトポロジーを用いても実施可能であることが理解されよう。更に、本発明はコンピュータ可読記憶媒体および（キャッシュメモリおよびメインメモリを含む）にアクセスする処理装置を含む汎用コンピュータ上で実行可能なソフトウエア上で実施可能であることが理解されよう。
【００４２】
本発明およびその利点を詳細に説明したが、特許請求の範囲に定める本発明の精神および範囲から逸脱することなくさまざまな変更、代替および改造が可能であることを指摘しておく。特に、実施形態は本発明を実施可能な１つの構成のみを例示するメモリ構造からなることが理解されよう。マイクロカーネルの位置およびサイズ、メモリセグメントのサイズ、メモリアドレスワードのサイズ等の変数および前記ワード内のビットの同定方法は本発明の精神および範囲から逸脱することなく変更可能であることは明らかであろう。
【００４３】
以上、本発明の実施例について詳述したが、以下、本発明の各実施態様の例を示す。
【００４４】
（実施態様１）
それぞれが独立したメモリノードにロードされた複数のマイクロカーネルへの複数の処理装置によるアクセスを可能とするシステムであって、第１のプロトコルによって全ての処理装置によるマイクロカーネルへの参照によって同一にアドレス指定されるベースノードがあらかじめ定められ、第２のプロトコルによって前記ノード上に同定可能な予約アドレス空間があらかじめ定められ、前記予約スペースは前記ノードによるマイクロカーネルの記憶に用いられ、処理装置参照によって指定されるメモリアドレスのビットシーケンスもまた（１）宛先ノード識別子および（２）前記宛先ノード内の宛先ノードアドレスを含むようにあらかじめ定められるシステムにおいて、
処理装置のメモリ参照によって指定されたメモリアドレスのビットシーケンスを構文解析する手段と、
前記メモリ参照の宛先ノードアドレスが第２のプロトコルに従ってノードにマイクロカーネルを記憶するように予約されたアドレス空間内にあるとき、前記メモリ参照をマイクロカーネル参照として認識する手段と、
マイクロカーネル参照上の宛先ノード識別子をそのマイクロカーネル参照が参照する対応するマイクロカーネルを記憶する局所ノードの識別子に変換する手段と
を有することを特徴とするシステム。
【００４５】
（実施態様２）
マイクロカーネル参照の宛先ノード識別子が第１のプロトコルによるベースノードを同定しない場合、プログラムエラーを報告する手段を有することを特徴とする実施態様１記載のシステム。
【００４６】
（実施態様３）
前記処理装置への後続のキャッシュコヒーレンシー要求に応答して、前記キャッシュコヒーレンシー要求を処理する前に局所ノード識別子を宛先ノード識別子に変換する手段を有することを特徴とする実施態様１または２記載のシステム。
【００４７】
【発明の効果】
以上のように、本発明を用いると、処理装置が変換テーブルやそれに関係する中間的な記憶を必要とすることなく複数のマイクロカーネルに共用的にアクセスすることが可能である。これによって、共用環境におけるマイクロカーネル参照の処理が最適化される。
【００４８】
また、本発明により、前記アクセスをキャッシュコヒーレントな環境で実行可能である。
【図面の簡単な説明】

【図１】それぞれがノード００および０１に対応する２つのメモリ構造００および０１のメモリアドレスレイアウトの例を示す図である。
【図１Ａ】物理的メモリ内の正確なアドレスを参照するマイクロカーネルコード内の「ワードロード」命令の一例を示す図である。
【図２】本発明の説明のために用いる４０ビットメモリ参照のビットレイアウトの一例を示す図である。
【図３】本発明に係るメモリ要求の処理において用いられる論理の一例を示すフローチャートである。
【図４】本発明に係るコヒーレンシー要求の処理において用いられる論理の一例を示すフローチャートである。
【図５】本発明を実行することのできるアーキテクチャおよびトポロジーの一例を機能レベルで示すブロック図である。
【図６】本発明の多処理装置、多ノード環境における一実施態様を機能レベルで示すブロック図である。
【符号の説明】
１０：物理的メモリ
１０１，１０２、１０３，１０４：６４ＭＢセグメント
１０１－１，１０１－２，１０１－３，１０１－４，１０２－１，１０２－２，１０２－３，１０２－４，１０３－１，１０３－２，１０３－３，１３１－４：セグメントの行
３０１、３０２、３０３、３０４、３０５、３０６、３０７、３０８、３０９、３１０、３１１：ステップ
４０１、４０２、４０３、４０４、４０５、４１０、４１１：ステップ
５０１：処理装置
５０２：メモリアクセスプロトコル
５０２Ａ：マイクロカーネル領域検出およびマッピング機能
５０２Ｂ：位置判定および経路指定機能
６０１：クロスバー
６０２：遠隔メモリ構造

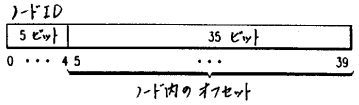
【図１】



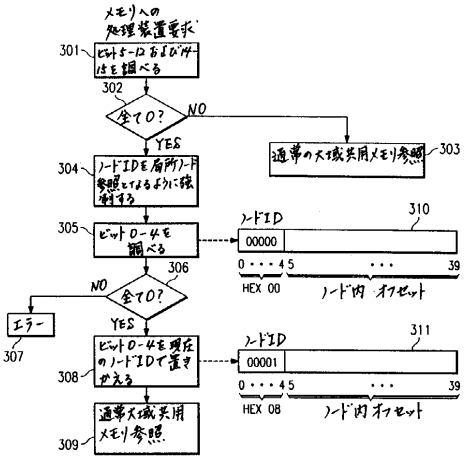
【図１Ａ】



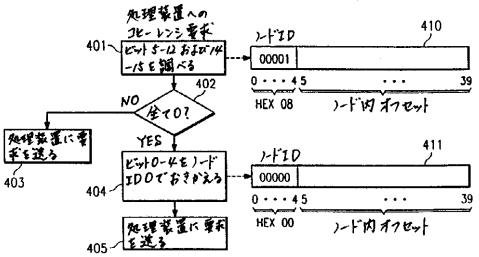
【図２】



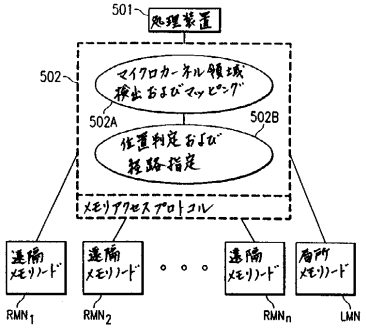
【図３】



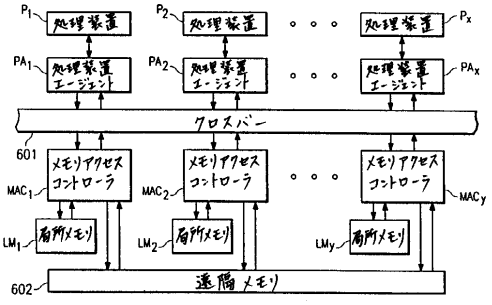
【図４】



【図５】



【図６】



フロントページの続き

(72)発明者 ケニース・チャネイ
アメリカ合衆国テキサス州プラノ ウエスト・メダリスト ６５４９
(72)発明者 ロジャー・サンシャイン
アメリカ合衆国テキサス州プラノ チャーバーグ・ドライブ ２００５

審査官 多賀 実

(56)参考文献 特開平０２－２４４２５３（ＪＰ，Ａ）
特開昭５９－１２１４５５（ＪＰ，Ａ）
特開平０８－０３０５６８（ＪＰ，Ａ）
特開平０７－２１０５２０（ＪＰ，Ａ）
特開昭６２－０３２５５５（ＪＰ，Ａ）
特開昭５７－０６０４５０（ＪＰ，Ａ）
A. Cox, R. Fowler著, The Implementation of a Coherent Memory Abstraction on a NUMA Multiprocessor:Experiences withPLATIUM, Proceedings of the 12th ACM Symposium on Operating Systems Principles, 米国, ACM, １９８９年, p.32-44

(58)調査した分野(Int.Cl.[7], ＤＢ名)
G06F12/02,12/08,15/177