(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2005-519391
(P2005-519391A)
(43)公表日　平成17年6月30日(2005.6.30)

(51)Int.Cl.[7]	F I			テーマコード（参考）
G06F 12/08	G06F	12/08	531B	5B005
G06F 12/06	G06F	12/08	531E	5B060
	G06F	12/08	551C	
	G06F	12/06	530A	

審査請求 有　予備審査請求 有　（全 21 頁）

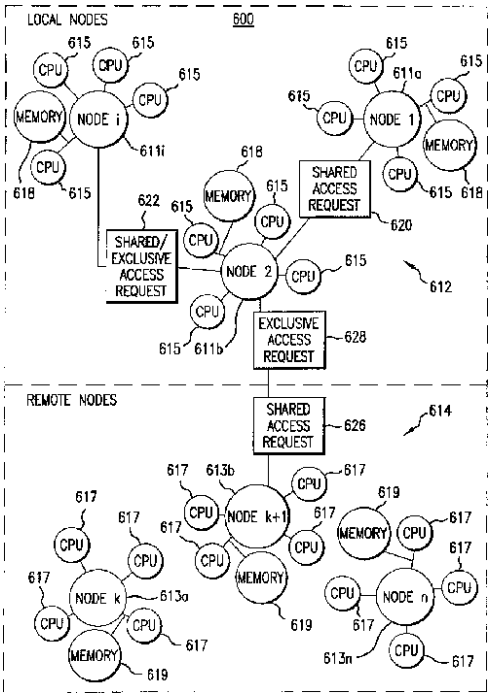
(21)出願番号　特願2003-573550 (P2003-573550)
(86)(22)出願日　平成14年2月28日 (2002.2.28)
(85)翻訳文提出日　平成16年10月25日 (2004.10.25)
(86)国際出願番号　PCT/US2002/005779
(87)国際公開番号　WO2003/075162
(87)国際公開日　平成15年9月12日 (2003.9.12)
(81)指定国　EP (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR) , JP

(71)出願人　597132333
シリコン、グラフィクス、インコーポレイテッド
ＳＩＬＩＣＯＮ　ＧＲＡＰＨＩＣＳ，　ＩＮＣ．
アメリカ合衆国カリフォルニア州、マウンテン　ビュー、アンフィシアター、パークウェイ、１６００
(74)代理人　100078282
弁理士　山本　秀策
(74)代理人　100062409
弁理士　安村　高明
(74)代理人　100113413
弁理士　森下　夏樹

最終頁に続く

(54)【発明の名称】共有ベクトルの増加を伴わないＤＳＭマルチプロセッサシステムにおけるキャッシュコヒーレンスのための方法およびシステム

(57)【要約】
本発明は、分散共有記憶（ＤＳＭ）マルチプロセッサシステム（６００）において、キャッシュコヒーレンスを保持するための方法とシステムを目的としている。この方法は、受信ノードによって、共有アクセス要求（６２０、６２６）が受信されることによって開始される。ここで、受信ノードとは、アクセスしたい情報を持つ少なくとも１つの主記憶装置を有する任意のノードである。次に、その共有アクセス要求が、ローカルノード（６１１）から発生したのかリモートノード（６１３）から発生したのかを判断する。共有アクセス要求がローカルノードから発生した場合、共有アクセス要求は、共有アクセス要求として処理される。共有アクセス要求が認可され、共有ベクトルが生成または更新されて共有ローカルノードに反映される。



【特許請求の範囲】
【請求項１】
分散共有記憶（ＤＳＭ）マルチプロセッサシステムにおけるキャッシュコヒーレンスのための方法であって、
（１）情報の共有アクセス要求を受信するステップと、
（２）前記共有アクセス要求は、ローカルノードから発生したのか、リモートノードから発生したのかを判断するステップと、
（３）前記共有アクセス要求が前記リモートノードから発生した場合、前記共有アクセス要求を排他的アクセス要求として処理するステップと、
（４）前記共有アクセス要求が前記ローカルノードから発生した場合、前記共有アクセス要求を処理するステップとを有する方法。
【請求項２】
前記共有アクセス要求が前記ローカルノードから発生したのか、前記リモートノードから発生したのかを判断する前記ステップは、さらに前記情報が格納されているメモリ位置を判断するステップを有することを特徴とする請求項１に記載の方法。
【請求項３】
排他的アクセス要求として前記共有アクセス要求を処理するステップは、さらに、
他のローカルノードまたはリモートノードへ介入要求を送信し、他のローカルノードまたはリモートノードから前記要求情報のコピーを削除するステップを有することを特徴とする請求項１に記載の方法。
【請求項４】
前記共有アクセス要求が前記ローカルノードから発生したのか、前記リモートノードから発生したのかを判断する前記ステップは、さらに共有アクセスリクエスタのアドレス上位ビットと受信ノードのアドレス上位ビットを比較するステップを有することを特徴とする請求項１に記載の方法。
【請求項５】
前記排他的アクセス要求として、前記共有アクセス要求を処理するステップは、さらに前記共有アクセスリクエスタのアドレス上位ビットと前記受信ノードのアドレス上位ビットとが一致する場合には、前記共有アクセス要求を前記排他的アクセス要求に変換するステップを有することを特徴とする請求項４に記載の方法。
【請求項６】
前記共有アクセス要求が前記ローカルノードから発生したのか、前記リモートノードから発生したのかを判断する前記ステップは、さらに、共有アクセス要求のアドレスと受信ノードに格納されているアドレス表とを比較するステップを有することを特徴とする請求項１に記載の方法。
【請求項７】
前記排他的アクセス要求として、前記共有アクセス要求を処理するステップは、さらに、
前記共有アクセスリクエスタの前記アドレスと、前記受信ノードに格納されている前記アドレス表中の少なくとも１つのアドレスとが一致する場合には、前記共有アクセス要求を前記排他的アクセス要求に変換するステップを有することを特徴とする請求項６に記載の方法。
【請求項８】
さらに、
（５）前記共有アクセス要求が前記排他的アクセス要求に変換される場合は、他のローカルノードまたはリモートノードから前記要求情報のコピーを削除するステップと、
（６）前記共有アクセス要求がリモートから発生する場合には、共有アクセス要求へのポインタ、前記共有アクセス要求の関連位置または前記情報のキャッシュ場所を格納するステップと、
（７）前記共有アクセス要求は有効かどうかを判断するステップと、
（８）前記排他的アクセス要求が最早、有効でない場合には、排他的アクセスを終了す

るステップと、
（９）前記共有アクセス要求によって要求された情報をキャッシュへ戻すステップと、
を有することを特徴とする請求項１に記載の方法。
【請求項９】
リモートノードからの共有アクセス要求を排他的アクセス要求として処理した場合において、さらに、
（５）前記ローカルノードから後続の共有アクセス要求を受信ノードで受信するステップと、
（６）前記ローカルノードが前記共有アクセス要求を送信する場合には、前記受信ノードからの介入要求を、前記リモートノードへ送信するステップと、
（７）前記リモートノードの排他的アクセスを送信するステップと、
（８）前記ローカルノードの前記共有アクセス要求を前記排他的アクセス要求として処理するステップとを有することを特徴とする請求項１に記載の方法。
【請求項１０】
ステップ（６）は、さらに前記ローカルノードの共有アクセスリクエスタのアドレス上位ビットと前記受信ノードのアドレス上位ビットとを比較するステップを有することを特徴とする請求項９に記載の方法。
【請求項１１】
ステップ（８）は、さらに前記ローカルノードの前記共有アクセスリクエスタのアドレス上位ビットと前記受信ノードのアドレス上位ビットとが一致する場合には、前記ローカルノードの前記共有アクセス要求を前記排他的アクセス要求に変換するステップを有することを特徴とする請求項１０に記載の方法。
【請求項１２】
別の介入要求を送信するステップ（６）は、さらに、前記ローカルノードの前記共有アクセス要求のアドレス上位ビットと前記受信ノードに格納されているアドレス表とを比較するステップを有することを特徴とする請求項９に記載の方法。
【請求項１３】
ステップ（８）は、さらに前記ローカルノードの前記共有アクセス要求の前記アドレスと、前記受信ノードに格納されている前記アドレス表中の少なくとも１つのアドレスとが一致する場合には、前記ローカルノードの前記共有アクセス要求を前記排他的アクセス要求に変換するステップを有することを特徴とする請求項１２に記載の方法。
【請求項１４】
さらに、
（９）前記ローカルノードの前記共有アクセス要求へのポインタを格納するステップと、
（１０）前記ローカルノードの前記共有アクセス要求が有効かどうかを判断するステップと、
（１１）前記排他的アクセス要求が最早、有効でない場合には、前記ローカルノードの前記排他的アクセス要求を終了するステップと、
（１２）前記ローカルノードの前記共有アクセス要求によって要求された情報を戻すステップとを有することを特徴とする請求項９に記載の方法。
【請求項１５】
ディレクトリに基づくプロトコルを有する分散共有記憶（ＤＳＭ）マルチプロセッサシステムにおけるキャッシュコヒーレンスのための方法であって、
（１）受信ノードによって、情報の共有アクセス要求を受信するステップと、
（２）前記共有アクセス要求は、ローカルノードから発生したのか、リモートノードから発生したのかを判断するステップと、
（３）前記共有アクセス要求が前記リモートノードから発生した場合、前記共有アクセス要求を、排他的アクセス要求として処理するステップと、
（４）前記共有アクセス要求が前記ローカルノードから発生した場合、前記共有アクセ

ス要求を処理するステップと、
（５）他のローカルノードまたはリモートノードへ介入要求を送信し、他のローカルノードまたはリモートノードから前記要求情報のコピーを削除するステップと、
（６）前記共有アクセス要求が前記排他的アクセス要求に変換される場合は、他のローカルノードまたはリモートノードから前記要求情報のコピーを削除するステップと、
（７）前記共有アクセス要求がリモートから発生する場合には、共有アクセス要求へのポインタを格納するステップと、
（８）前記共有アクセス要求は有効かどうかを判断するステップと、
（９）前記排他的アクセス要求が最早、有効でない場合には、排他的アクセスを終了するステップと、
（１０）前記共有アクセス要求によって要求された情報をキャッシュへ戻すステップとを有することを特徴とする方法。
【請求項１６】
ステップ（２）は、さらに前記共有アクセスリクエスタのアドレス上位ビットと前記受信ノードのアドレス上位ビットを比較するステップを有することを特徴とする請求項１５に記載の方法。
【請求項１７】
ステップ（３）は、さらに前記共有アクセスリクエスタのアドレス上位ビットと前記受信ノードのアドレス上位ビットとが一致する場合には、前記共有アクセス要求を前記排他的アクセス要求に変換するステップを有することを特徴とする請求項１６に記載の方法。
【請求項１８】
ステップ（２）は、さらに前記共有アクセス要求のアドレスと前記受信ノードに格納されているアドレス表とを比較するステップを有することを特徴とする請求項１５に記載の方法。
【請求項１９】
ステップ（３）は、さらに前記共有アクセスリクエスタの前記アドレスと、前記受信ノードに格納されている前記アドレス表中の少なくとも１つのアドレスとが一致する場合には、前記共有アクセス要求を前記排他的アクセス要求に変換するステップを有することを特徴とする請求項１８に記載の方法。
【請求項２０】
さらに、
（１１）前記ローカルノードから共有アクセス要求を受信するステップと、
（１２）前記ローカルノードが前記共有アクセス要求を送信する場合には、前記受信ノードからの介入要求を、前記リモートノードへ送信するステップと、
（１３）前記リモートノードの前記排他的アクセスを送信するステップと、
（１４）前記ローカルノードの前記共有アクセス要求を前記排他的アクセス要求として処理するステップとを有することを特徴とする請求項１５に記載の方法。
【請求項２１】
ステップ（１２）は、さらに前記ローカルノードの前記共有アクセスリクエスタのアドレス上位ビットと前記受信ノードのアドレス上位ビットとを比較するステップを有することを特徴とする請求項２０に記載の方法。
【請求項２２】
ステップ（１４）は、さらに前記ローカルノードの前記共有アクセスリクエスタのアドレス上位ビットと前記受信ノードのアドレス上位ビットとが一致する場合には、前記ローカルノードの前記共有アクセス要求を前記排他的アクセス要求に変換するステップを有することを特徴とする請求項２１に記載の方法。
【請求項２３】
ステップ（１２）は、さらに前記ローカルノードの前記共有アクセスリクエスタのアドレス上位ビットと前記受信ノードに格納されているアドレス表とを比較するステップを有することを特徴とする請求項２０に記載の方法。

【請求項２４】
ステップ（１４）は、さらに前記ローカルノードの前記共有アクセスリクエスタの前記アドレスと、前記受信ノードに格納されている前記アドレス表中の少なくとも１つのアドレスとが一致する場合には、前記ローカルノードの前記共有アクセス要求を前記排他的アクセス要求に変換するステップを有することを特徴とする請求項２３に記載の方法。
【請求項２５】
さらに、
（１５）前記ローカルノードの前記共有アクセス要求へのポインタを格納するステップと、
（１６）前記ローカルノードの前記共有アクセス要求が有効かどうかを判断するステップと、
（１７）前記排他的アクセス要求が最早、有効でない場合には、前記ローカルノードの前記排他的アクセス要求を終了するステップと、
（１８）前記ローカルノードの前記共有アクセス要求によって要求された情報をキャッシュへ戻すステップとを有することを特徴とする請求項２０に記載の方法。
【請求項２６】
分散共有記憶（ＤＳＭ）マルチプロセッサシステムにおけるキャッシュコヒーレンスのための方法であって、
（１）共有情報を有する受信ノードによって、第１共有アクセス要求を受信するステップと、
（２）前記受信ノードによって、第２共有アクセス要求を受信するステップと、
（３）第１共有アクセスリクエスタおよび第２共有アクセスリクエスタの一方を、排他的アクセス要求として処理するステップと、
（４）前記第１共有アクセスリクエスタおよび前記第２共有アクセスリクエスタの他方へ前記介入要求を送信するステップとを有することを特徴とする方法。
【請求項２７】
ステップ（３）は、さらに、
（５）前記第１共有アクセスリクエスタおよび前記第２共有アクセスリクエスタの一方は、ローカルノードから発生したのか、リモートノードから発生したのかを判断するステップと、
（６）前記第１共有アクセス要求が前記リモートノードから発生した場合、前記第１共有アクセス要求を、前記排他的アクセス要求として処理するステップと、
（７）第２共有アクセスリクエスタへ介入要求を送信するステップとを有することを特徴とする請求項２６に記載の方法。
【請求項２８】
ステップ（３）は、さらに、
（５）前記第１共有アクセスリクエスタおよび前記第２共有アクセスリクエスタの一方は、ローカルノードから発生したのか、リモートノードから発生したのかを判断するステップと、
（６）前記第１共有アクセス要求が前記リモートノードから発生した場合、前記第２共有アクセス要求を、前記排他的アクセス要求として処理するステップと、
（７）第１共有アクセスリクエスタへ介入要求を送信するステップとを有することを特徴とする請求項２６に記載の方法。
【請求項２９】
さらに、
前記第１共有アクセス要求および前記第２共有アクセス要求の他方が、ローカルノードから発生した場合には、前記第１共有アクセスリクエスタおよび前記第２共有アクセスリクエスタの一方を共有アクセス要求として処理するステップを有することを特徴とする請求項２６に記載の方法。
【請求項３０】

さらに、
前記第１共有アクセス要求および前記第２共有アクセス要求の他方が、ローカルノードから発生した場合には、リモートノードから発生した前記第１共有アクセス要求および前記第２共有アクセス要求の一方の排他的アクセスを無効とするステップを有することを特徴とする請求項２６に記載の方法。
【請求項３１】
前記無効にするステップは、さらに、
（５）前記第１共有アクセスリクエスタおよび前記第２共有アクセスリクエスタの一方のアドレス上位ビットと受信ノードのアドレス上位ビットを比較するステップと、
（６）前記第１共有アクセス要求および前記第２共有アクセス要求の一方のアドレス上位ビットと受信ノードのアドレス上位ビットとが一致する場合には、前記第１共有アクセス要求および前記第２共有アクセス要求の一方を前記排他的アクセス要求として処理するステップとを有することを特徴とする請求項３０に記載の方法。
【請求項３２】
前記無効にするステップは、さらに、
（５）前記第１共有アクセスリクエスタおよび前記第２共有アクセスリクエスタの一方のアドレス上位ビットと受信ノードに格納されたアドレス表とを比較するステップと、
（６）前記第１共有アクセス要求および前記第２共有アクセス要求の一方のアドレス上位ビットと受信ノードに格納されたアドレス表とが一致する場合には、前記第１共有アクセス要求および前記第２共有アクセス要求の一方を前記排他的アクセス要求として処理するステップとを有することを特徴とする請求項３０に記載の方法。
【請求項３３】
さらに、
（７）前記第１共有アクセスリクエスタおよび前記第２共有アクセスリクエスタの一方へのポインタを格納するステップと、
（８）前記第１共有アクセス要求および前記第２共有アクセス要求の前記一方が有効かどうかを判断するステップと、
（９）前記第１共有アクセス要求および前記第２共有アクセス要求の一方が最早、有効でない場合には、前記第１共有アクセスリクエスタおよび前記第２共有アクセスリクエスタの一方の排他的アクセスを終了するステップと、
（１０）前記第１共有アクセスリクエスタおよび前記第２共有アクセスリクエスタの一方によって要求された情報をキャッシュへ戻すステップとを有することを特徴とする請求項３２に記載の方法。
【請求項３４】
分散共有記憶（ＤＳＭ）マルチプロセッサシステムにおけるキャッシュコヒーレンスを保持するためのシステムであって、
複数の処理ノードであって、各々の前記処理ノードは少なくとも１つの中央処理装置（ＣＰＵ）および記憶装置に接続されている処理ノードと、
共有アクセス要求を受信するための少なくとも１つの受信ノードと、
前記共有アクセス要求が前記ローカルノードから発生したか、または前記リモートノードから発生したかを判断するための手段と、
複数の共有メモリベクトルであって、各々の前記共有メモリベクトルは、前記受信ノードから情報共有を要求する前記ローカルノードまたは前記受信ノードから情報共有を要求する前記リモートノードのいずれかを定義する共有メモリベクトルと、
排他的メモリアクセスポインタであって、前記排他的メモリアクセスのポインタは、前記受信ノードの情報に対して排他的アクセスを認可された前記ＤＳＭマルチプロセッサシステムにおける前記リモートノードを指示すメモリアクセスポインタと、
前記排他的メモリアクセスポインタを格納するための記憶装置と、
前記排他的メモリアクセスポインタが無効になった場合には、前記排他的メモリアクセスポインタを削除するための削除手段とを有することを特徴とするシステム。

【請求項３５】
前記システムは、ディレクトリに基づくプロトコル環境を使用して動作する請求項３４に記載のシステム。
【請求項３６】
前記削除手段はさらに、一度、排他的メモリアクセスポインタが前記ＤＳＭマルチプロセッサシステムにおける前記リモートノードを指し示すと、前記他の処理ノードへ介入要求を送信する手段を有することを特徴とする請求項３４に記載のシステム。
【請求項３７】
前記システムはさらに、前記受信ノードが前記ローカルノードから前記共有アクセス要求を受信した場合には、前記リモートノードへ別の介入要求を送信する手段を有することを特徴とする請求項３４に記載のシステム。
【請求項３８】
ディレクトリに基づくプロトコル環境を使用して、分散共有記憶（ＤＳＭ）マルチプロセッサシステムにおけるキャッシュコヒーレンスを保持するためのシステムであって、
複数の処理ノードであって、各々の前記処理ノードは少なくとも１つの中央処理装置（ＣＰＵ）および記憶装置に接続されている処理ノードと、
共有アクセス要求を受信するための少なくとも１つの受信ノードと、
前記共有アクセス要求が前記ローカルノードから発生したか、または前記リモートノードから発生したかを判断するための手段と、
複数の共有メモリベクトルであって、各々の前記共有メモリベクトルは、前記受信ノードから情報共有を要求する前記ローカルノードまたは前記受信ノードから情報共有を要求する前記リモートノードのいずれかを定義する共有メモリベクトルと、
排他的メモリアクセスポインタであって、前記排他的メモリアクセスのポインタは、前記受信ノードの情報に対して排他的アクセスを認可された前記ＤＳＭマルチプロセッサシステムにおける前記リモートノードを指示すメモリアクセスポインタと、
前記排他的メモリアクセスポインタが、一度、前記ＤＳＭマルチプロセッサシステムにおける前記リモートノードを指示すと、前記他の処理ノードへ介入要求を送信する手段と、
前記排他的メモリアクセスポインタを格納するための記憶装置と、
前記排他的メモリアクセスポインタが無効になった場合には、前記排他的メモリアクセスポインタを削除するための削除手段とを有することを特徴とするシステム。
【請求項３９】
ディレクトリに基づくプロトコル環境を使用して、分散共有記憶（ＤＳＭ）マルチプロセッサシステムにおけるキャッシュコヒーレンスを保持するためのシステムであって、
情報の共有アクセス要求を受信する手段と、
前記共有アクセス要求は、ローカルノードから発生したのか、リモートノードから発生したのかを判断する手段と、
前記共有アクセス要求が前記リモートノードから発生した場合、前記共有アクセス要求を、排他的アクセス要求として処理する手段と、
前記共有アクセス要求が前記ローカルノードから発生した場合、前記共有アクセス要求を処理する手段とを有することを特徴とするシステム。
【請求項４０】
ディレクトリに基づく環境を使用して、分散共有記憶（ＤＳＭ）システムにおけるキャッシュコヒーレンスを保持するためのシステムであって、
情報のアクセス要求を受信する手段と、
前記アクセス要求は、ローカルノードから発生したのか、リモートノードから発生したのかを判断する手段と、
前記アクセス要求が前記ローカルノードから発生したか前記リモートノードから発生したかによって、前記アクセス要求を、異なった仕方で処理する手段と、
を処理する手段とを有することを特徴とするシステム。

【請求項４１】
コンピュータプログラムロジックをその中に備えるコンピュータ使用可能媒体を有するコンピュータプログラムプロダクトであって、前記コンピュータプログラムロジックは、分散共有記憶（ＤＳＭ）システムにおけるキャッシュコヒーレンスを保持することを、コンピュータシステムによって可能にし、前記コンピュータプログラムロジックは、
情報の共有アクセス要求を受信するのをコンピュータで可能にする第１機能と、
前記共有アクセス要求は、ローカルノードから発生したのか、リモートノードから発生したのかを判断するのをコンピュータで可能にする第２機能と、
前記共有アクセス要求が前記リモートノードから発生した場合、前記共有アクセス要求を、排他的アクセス要求として処理するのをコンピュータで可能にする第３機能と、
前記共有アクセス要求が前記ローカルノードから発生した場合、前記共有アクセス要求を処理するのをコンピュータで可能にする第４機能とを有することを特徴とするコンピュータプログラムプロダクト。
【発明の詳細な説明】
【技術分野】
【０００１】
（発明の分野）
本発明は、分散共有記憶（ＤＳＭ）マルチプロセッサシステムにおけるキャッシュコヒーレンスのための方法およびシステムに関する。
【背景技術】
【０００２】
（背景技術）
単一プロセッサコンピュータシステムおよびマルチプロセッサコンピュータシステムを含むコンピュータシステムにおいては、一般的に一時点で、多数のプロセスまたはスレッドが実行される。各プロセスは、いくらかの物理メモリを必要とする。多くの場合、物理メモリは限られていて、異なるプロセス間で割り当てられることが必要である。
【０００３】
コンピュータシステムは、一般にメモリアクセスタイムを低減するために、主記憶と各プロセッサとの間に、一段以上のキャッシュメモリを使用している。キャッシュメモリは、主記憶から取り出された情報を格納する。プロセッサによって取り出された情報は、プロセッサに達するために、一段以上のキャッシュを経る必要がある。キャッシュは、小さくてプロセッサに物理的に接近していることが多く、プロセッサと同一チップ上に位置することがある。このため、キャッシュされた情報は、概して主記憶に格納された情報より、はるかに高速でアクセス可能である。よって、キャッシュは一般にプロセッサによって、繰返しアクセスが必要な情報を格納する。
【０００４】
キャッシュの整合性を保持するためのシステムおよび方法には、ディレクトリプロトコルを含み、このディレクトリプロトコルにおいて、集中ディレクトリにメモリの状態が保持されている。情報は、その情報を「共有」している異なるプロセスによって、多数の場所にキャッシュされている。あるいは、１つのプロセスは、ある期間中、その情報に対して「排他的」権限をもつことができる。主記憶またはキャッシュ場所のいずれかにある情報をプロセスが変更した場合は、その情報の他のインスタンスを無効とするかまたは更新する必要がある。これを、キャッシュ整合性の保持と呼ぶ。分散共有記憶（ＤＳＭ）システムにおいては、ディレクトリを分散することができる。集中制御装置が、共有情報の整合性の保持を担当する。あるメモリ位置に格納されている情報が変更された場合はいつでも、集中ディレクトリのチェックが行われ、当該情報のコピーがキャッシュに格納されているかどうかが判断される。格納されている場合には、各コピーは、更新されるかまたは無効にされる。
【０００５】
大規模なＤＳＭマルチプロセスシステムにおいては、キャッシュコヒーレンスの保持は

、困難な作業となることがある。ＤＳＭシステムでは、一般的に、当該情報をどこにキャッシュしているかを識別するのに、共有ベクトルを使用している。しかし、ＤＳＭシステムの規模が増加するのにつれて（例えば、プロセスまたは処理ノードの数）、共有ベクトルが、それにつれて増大し、これにより処理速度が低下し、キャッシュにある情報が他のプロセッサによって使用できない期間が増える。

【発明の開示】

【発明が解決しようとする課題】

【０００６】

したがって、ＤＳＭマルチプロセッサシステムにおけるキャッシュコヒーレンスに関する、より優れたシステムが必要となっている。

【課題を解決するための手段】

【０００７】

（発明の要旨）

本発明は、分散共有記憶（ＤＳＭ）マルチプロセッサシステムにおけるキャッシュコヒーレンシのためのシステムおよび方法に関する。ＤＳＭマルチプロセッサシステムは、複数のノードを有することがある。各ノードは、さらに少なくとも１つの中央処理装置（ＣＰＵ）、キャッシュ記憶装置およびオプションの入出力（Ｉ／Ｏ）装置を有する。ＤＳＭマルチプロセッサシステムは、一般に、プロセッサに対して制御（例えば、スレッドスケジューリング、メモリアクセス等）を維持する１つ以上のオペレーティングシステムを備えている。

【０００８】

一般に、リクエスタ（例えば、プロセッサ上で実行されるプロセス）が、情報へのアクセスを要求すると、その情報のコピーが、その情報を格納しているメモリアドレスからリクエスタに関連付けられたキャッシュアドレスへ送信される。次に、リクエスタは、キャッシュアドレスから当該情報をアクセスする。キャッシュコヒーレンスの目的のために、状態表示は一般に当該メモリアドレスに関連付けられていて、当該情報をキャッシュしている場所を示している。

【０００９】

アクセス要求は、排他的アクセスまたは共有アクセスとすることができる。上記の状態表示は、一般にそのリクエスタに認可されたアクセスの種類の表示を含む。例えば、排他的アクセスが認可された場合、その状態表示には当該リクエスタに対して、排他的アクセスが認可されたことが識別される。共有アクセスが認可された場合には、当該状態表示は、一般に当該情報のコピーが存在する１つ以上のキャッシュ位置を示す共有ベクトルを含んでいる。付加的な規則および手順を、競合する規則または／およびその他のシナリオを処理するために実装することができる。

【００１０】

当今のＤＳＭマルチプロセッサシステムは、数十、数百、さらに数千ものＣＰＵを備えることができる。より多くのリクエスタが当該情報の共有アクセスを取得すると、その共有ベクトルのサイズが増加し、場合により劇的に増加する。これにより、メモリ空間が占有され、処理速度が減少する。

【００１１】

本発明では、ローカルリクエスタおよびリモートリクエスタの概念を導入することによって、発生し得る共有ベクトルのサイズを減少する。一般的に、「ローカル」および「リモート」という用語は、リクエスタの要求されたメモリ位置への物理的な近接度に対して定義される。リモートおよびローカルに対するその他の定義も使用可能である。

【００１２】

運用中、リクエスタが情報の共有アクセスを要求すると、当該情報が格納されているメモリ位置に関してそのリクエスタがローカルかまたはリモートかを判断する。リクエスタがローカルの場合、その共有アクセス要求は、通常の共有アクセス要求手順に従って、処理される。リクエスタがリモートの場合には、その共有アクセス要求は、排他的要求に変

換され排他的要求の手順に従って処理される。したがって、リモートリクエスタのキャッシュ整合性を維持するのに共有アクセスベクトルは必要ではない。このことは、メモリ要求および処理要求をかなり低減し、特に大規模なＤＳＭマルチプロセッサシステムにおいてはこの低減が著しい。

【００１３】

ある実施の形態においては、追加の規則を実行できる。例えば、前述のように、リモートリクエスタからの共有アクセス要求が排他的アクセス要求に変換され、ローカルリクエスタから後続の共有アクセス要求が受信された状況に対して、新しい規則を実行することができる。新しい規則の例としては、リモートリクエスタに認可された排他的アクセスを終了し、当該ローカルリクエスタへの共有アクセスを認可することである。当該状態表示が、排他的アクセスの表示から共有ベクトルの表示に変更され、当該ローカルリクエスタへの共有アクセスが表示される。リモートリクエスタが、その情報を再度必要とする場合は、再要求をする必要がある。

【００１４】

（好ましい実施形態の詳細な説明）

（１．概要）

本発明は、分散共有記憶（ＤＳＭ）マルチプロセッサシステムにおいて、キャッシュコヒーレンシを向上するための方法およびシステムに関する。ＤＳＭシステムは一般に、複数のノードを含み、これらの複数のノードは、これらのノードに接続された複数の中央処理装置（ＣＰＵ）および記憶装置ならびにノードから情報を送受信するための入出力（Ｉ／Ｏ）モジュールを含む。ノード内の記憶装置は、種々の機能や手順を実行するために、他のノードまたはリクエスタで必要な情報を含むことがある。１つ以上のリクエスタが、要求する情報への共有アクセスを認可された場合には、共有ベクトルが共有リクエスタを識別する。この共有ベクトルは、キャッシュ整合性を維持するために使用される。当該リクエスタが、要求する情報への排他的アクセスを認可されると、ポインタ（要求元ノードの２進数）が、共有ベクトルの代わりに格納される。例えば、当該共有ベクトルの大きさがＮビットならば、ポインタの大きさがｌｏｇ（Ｎ）ビットである。ここでＮはシステム内のノード数である。

【００１５】

共有リクエスタの数が増加すると、共有ベクトルの大きさが増大する。共有ベクトルの大きさが増大するにつれて、より多くのメモリを消費し、システム全体および各特定のノードの機能と手順の処理速度を低下させる。したがって、共有ベクトルの大きさを低減するより良好なシステムが必要性である。

【００１６】

本発明によれば、要求情報が存在するノードから離れたノードから、共有アクセスの要求が発生すると、当該共有アクセス要求は排他的アクセス要求に変換される。これによって、当該要求情報の共有ベクトルを保持する必要がなくなる。ある実施例では、リクエスタが要求された情報から離れているかどうかの判定は、当該リクエスタおよび当該要求情報の物理アドレスに基づく。

【００１７】

１つの実施例では、アドレス比較は、要求元ノードのアドレス上位ビットと要求された情報のアドレス上位ビットとの間で行われる。別の実施例では、あるノードは共有アクセスを認可されたり、されていない他のノードのアドレス表を格納している。換言すれば、１つのノードが別のノードへの共有情報アクセス要求を送信すると、後者のノードは要求元ノードのアドレスをアドレス表と比較する。当該アドレスが表中のいずれかのアドレス一致すると、共有情報アクセス要求は認可されるか、または排他的要求に変換される。

【００１８】

ある実施例において、あるローカルノードが、リモートノードによって排他的に保持されている情報へのアクセス要求を送信すると、当該リモートノードのアクセスは除去され、そのローカルノードに対するアクセスが認可される。

【００１９】
（２．事例環境）
本発明は、分散共有記憶（ＤＳＭ）マルチプロセッサシステムにおいて実現できる。しかし、上記の実現は、本発明の使用をＤＳＭシステムに限定するものとして考えられるべきではない。
以下は、本発明が実現される環境を理解するのに有用となるシステムの検討である。
【００２０】
図１において、単一プロセッサシステム１１０は、主記憶１１２へ接続された単一のプロセッサおよびキャッシュノード１１４を備える。主記憶１１２は、プロセッサおよびキャッシュノード１１４によって使用される情報を格納する。入出力Ｉ／Ｏシステム１１６は、例えばコンピュータ端末および記憶ディスクを含むユーザインタフェースおよび記憶装置などの周辺装置へのインタフェースを提供する。
【００２１】
図４において、プロセッサおよびキャッシュノード１１４は、プロセッサおよびキャッシュノード４１０として実現できる。プロセッサおよびキャッシュノード４１０は、主記憶に接続されたプロセッサ４１２を備え、これはキャッシュ４１６を介した主記憶１１２であってもよい。明確さのために、単一のプロセッサ４１２およびキャッシュメモリ４１６を表示してある。当業者は、マルチプロセッサおよび多数段のキャッシュが使用できることを理解するであろう。
【００２２】
キャッシュ４１６は、主記憶１１２のような主記憶から取り出した情報を、キャッシュするために設けられている。情報がキャッシュ４１６にキャッシュされるとすぐに、プロセッサ４１２は、キャッシュ４１６から情報を取り出すことができる。プロセッサ４１２は、一般に、主記憶１１２内の情報のアクセスより高速で、キャッシュ４１６から情報を取り出すことができる。この理由は、キャッシュ４１６がプロセッサ４１２に近接しており、またキャッシュ４１６を構成しているメモリ部品が、主記憶を構成しているメモリ部品より高速であるからである。キャッシュ４１６は、ユーザの必要性によって指定された１段以上のキャッシュを備えることができる。
【００２３】
運用中、プロセッサ４１２は、１つ以上のプロセスのスレッドを処理する。プロセッサ４１２が、主記憶１１２に格納されている情報のアクセスを必要とするときは、アクセス要求が送信される。プロセッサ４１２が、要求した情報へのアクセスを許可されると、主記憶１１２は、要求された情報をキャッシュ４１６へ返却する。一度、要求された情報がキャッシュ４１６に格納されると、プロセッサ４１２は、その情報を必要に応じてアクセスできる。以後、プロセッサ４１２は主記憶１１２をアクセスせずに、キャッシュ４１６内の情報をアクセスできる。
【００２４】
図２において、集中化共有メモリ対称型マルチ処理（ＳＭＰ）システム２１０の例は、複数のプロセッサおよびキャッシュノード２１２から２１８を備えている。ＳＭＰ２１０は、任意の数のノード２１２から２１８を含むことができる。プロセッサおよびキャッシュノード２１２から２１８は、バス２２２を介して、集中化共有主記憶２２０に接続されている。Ｉ／Ｏシステム２２４を、ＳＭＰ２１０と、コンピュータ端末および記憶ディスクのような種々の外部装置および周辺装置とのインターフェースのために設けることができる。
【００２５】
プロセッサおよびキャッシュノード２１２から２１８は、例えば、上記の図４のプロセッサおよびキャッシュノード４１０として実装できる。あるいは、１つ以上のプロセッサおよびキャッシュノード２１２および２１８は、複数のプロセッサ４１２およびキャッシュ４１６を使用できる。どちらの実装においても、ＳＭＰ２１０は、多数のプロセッサ４１２に複数タスクの並列処理を可能にする。集中化共有メモリ２２０は、マルチプロセッ

サ４１２がタスク間の情報を共有するのを許可する。
【００２６】
図３において、分散共有記憶（ＤＳＭ）システム３１０は、相互接続ネットワーク３４４を介して相互接続されたいくつかの処理ノード３５０から３６０を含む。ＤＳＭ３１０は、任意の数の処理ノード３５０から３６０を含むことができる。各処理ノード３５０から３６０は、プロセッサおよびキャッシュノード３１２から３２２および分散共有記憶３２８から３３８の一部と共に描かれている。当業者には明らかなように、１つ以上の処理ノード３５０から３６０では、プロセッサおよびキャッシュノードを使用する必要がない。
【００２７】
プロセッサおよびキャッシュノード３１２から３２２は、例えば、上記の図４のプロセッサおよびキャッシュノード４１０として実装できる。ここでは、各プロセッサ４１２は、１段以上のキャッシュ４１６を経て、共有メモリ３２８から３３８の一部をアクセスする。あるいは、１つ以上のプロセッサおよびキャッシュノード３１２から３２２は、複数のプロセッサ４１２およびキャッシュ４１６をもつことができる。
【００２８】
分散共有メモリ部分３２８から３３８は、単一の連続ブロックの物理メモリから形成されている場合は、処理ノード３５０から３６０内のプロセッサによってアクセスされる。当業者には明らかなように、１つ以上の処理ノード３５０から３６０は、共有メモリの一部を使用する必要がない。
【００２９】
図３の例において、各処理ノード３５０から３６０は、オプションの入出力（Ｉ／Ｏ）装置と共に表示されている。当業者には明らかなように、１つ以上の処理ノード３５０から３６０は、Ｉ／Ｏ装置を有する必要がない。さらに、異なる種類のＩ／Ｏ装置および外部周辺機器および資源の組合せがＤＳＭシステムで使用できる。よって、１つ以上の処理ノード３５０から３６０は、プロセッサまたはプロセッサなし、共有メモリまたは共有メモリなし、およびＩ／ＯありまたはＩ／Ｏなしのどのような組合せをも含むことができる。
【００３０】
図３の例において、各処理ノード３５０から３６０は、キャッシュコヒーレンシディレクトリと共に表示されている。ディレクトリ情報は、メモリ制御装置および／またはキャッシュタグと関連付けることができる。
【００３１】
ＤＳＭ３１０全体で、物理記憶または主記憶３２８から３３８を区別することによって、各処理ノード３５０から３６０は、主記憶の一部を含むことができる。このプロセッサとメモリとの間の物理的な近接によって、処理ノード内のプロセッサおよびメモリに対する記憶待ち時間が減少される。
【００３２】
図１から図３と関連して記載したシステム１１０、２１０および３１０のような単一プロセッサシステム、ＳＭＰＳおよびＤＳＭは周知のものである。このようなシステムのさらに詳細については、例えば、パターソンとヘネシー、コンピュータアーキテクチャの数量的なアプローチ、第２版（Ｈｅｎｎｅｓｓｙ ａｎｄ Ｐａｔｔｅｒｓｏｎ，Ｃｏｍｐｕｔｅｒ Ａｒｃｈｉｔｅｃｔｕｒｅ Ａ Ｑｕａｎｔｉｔａｔｉｖｅ Ａｐｐｒｏａｃｈ，２ｄ Ｅｄ．（Ｍｏｒｇａｎ ａｎｄ Ｋａｕｆｍａｎｎ Ｐｕｂｌ．：ＵＳＡ１９９６））に見られ、ここに参照のため、その全体が組込まれている。
【００３３】
（３．ＤＳＭマルチプロセッサシステムにおけるキャッシュコヒーレンスのためのシステムおよび方法）
本発明は、分散共有記憶マルチプロセッサシステムにおいて、いくつかの共有アクセス要求を排他的アクセス要求へ変換することによって、発生し得る共有ベクトルの大きさを

減少する。これによって、これらの環境において共有ベクトルの必要性を除去する。本発明は、ソフトウェア、ハードウェア、ファームウェアまたはこれらのいずれかの組合せによって実現できる。

【００３４】

大きなベクトルの問題は、図３に関連して記述する。処理ノード３５２が情報を必要とする場合、一般に、要求は相互接続ネットワーク３４４に渡って送信される。要求された情報が、処理ノード３５０の主記憶３２８に存在する場合、処理ノード３５２からその要求は、処理ノード３５０へと送られる。その要求が共有アクセス要求で、その要求が認可されたものならば、その主記憶３２８に関連付けられた共有ベクトルが生成または更新され、処理ノード３５２に対して当該アクセスがなされたことが反映される。

【００３５】

その要求が、排他的アクセス要求であれば、共有ベクトルの必要はない。その代わり、処理ノード３５２へ排他的アクセスが認可された表示が格納される。共有ベクトルおよび排他的アクセスの表示が、キャッシュコヒーレンシー機構によって使用される。

【００３６】

さらに多くの処理ノードが、同一情報に対して、共有アクセスを要求すると、その共有ベクトルは、大きさが増大し、より多くのメモリ空間を必要として、システム全体の処理速度を低下させる。

【００３７】

本発明によれば、発生し得る共有エントリの数の減少によって、発生し得る共有ベクトルの大きさは減少する。これは、ローカルノードとリモートノードとを区別することによって行われる。ローカルノードとリモートノードは、要求情報が存在するところの「ホーム」ノードに対比して定義される。上記の例において、処理ノード３５２が処理ノード３５０から情報を要求する場合には、処理ノード３５０はホームノードと呼ばれる。この要求に関して、ホームノード３５０に対して、リクエスタがローカルまたはリモートと定義される。本発明によれば、共有アクセス要求がリモートノードから発生する場合、その共有要求は排他的要求に変換される。その共有アクセス要求がローカルノードから発生する場合、その要求は共有アクセス要求として処理される。

【００３８】

したがって、ノード３５０から３６０は、そのようなノードが情報アクセスを必要とする場合、他のノードのキャッシュ装置へ向かう共有ベクトルを有することがある。図３のシステム３１０には、６個の処理ノードが表示されている。しかし、当業者にとっては、より小さいシステムおよびより大きいシステムが可能なことが理解される。一般に、多数の処理ノードは、潜在的に大きな共有ベクトルとなる。大きな共有ベクトルは、情報処理の速度を減少する。これは、一般に大きな共有ベクトルを扱うには、時間を要するからである。

【００３９】

本発明の例を、図６を参照して記述する。図６において本発明の実施形態にかかわるＤＳＭシステム６００を示す。システム６００は、複数の処理ノードつまりリクエスタ６１１および６１３を有する。処理ノード６１１はローカルノードまたはローカルリクエスタを表す。処理ノード６１３は、リモートノードまたはリモートリクエスタを表す。ノードは、システム内の任意の処理装置、キャッシュメモリ位置、または情報および／または処理機能または手順のいずれかを格納可能な他の任意の媒体とすることができる。

【００４０】

「ホームノード」という用語は、要求された情報が存在するノードである。「ホームノード」という用語は、処理装置またはメモリ位置または情報を格納可能な、およびＤＳＭマルチプロセッサシステムにおいて、他の処理装置が情報を要求する他の任意の媒体を表現できる。

【００４１】

「ホームノード」という用語は、相対的なものであり本発明の範囲を限定するものでは

なく、説明のために使用している。したがって、本検討のためだけに、ノード２をホームノード６１１ｂに指定する。当業者は、一般にＤＳＭマルチプロセッサシステム内の任意のノードは、同上システムの他の任意のノードおよびホームノードに指定したノードからも情報を要求できることを理解している。

【００４２】

ローカルノードは、ホームノード６１１ｂに比較的に近接したノードグループ６１２に位置している。リモートノードは、ノードグループ６１４に位置していて、概してホームノード６１１ｂに比較的に近接していない。

【００４３】

図６の例において、ノードグループ６１２内にｉ個のローカル処理ノード６１１があり、ノードグループ６１４内にｎ個のリモート処理ノードがある。システム６００の要求に従って、ｉおよびｎの数は可変であることを当業者は理解している。

【００４４】

各ローカルノード６１１は、複数の中央処理装置（ＣＰＵ）６１５および複数の記憶装置６１８を有することがある（図６に示すように、各ノード６１１毎に、４個のＣＰＵ６１５および単一記憶装置６１８がある）。同様に、各リモートノード６１３は複数のＣＰＵ６１７および複数の記憶装置６１９を有することがある（図６に示すように、各ノード６１３毎に、４個のＣＰＵ６１７および単一の記憶装置６１９がある）。当業者は、他の実施形態が可能であることを理解している。

【００４５】

さて、本発明の使用例を説明する。ローカルノード６１１ａは、共有アクセス要求６２０をホームノード６１１ｂへ送信する。ノード６１１ａは、ホームノード６１１ｂに対してローカルノードであるので、その要求６２０はホームノード６１１ｂによって共有アクセス要求として処理される。

【００４６】

共有アクセス要求がリモートノードから到着すると、排他的アクセス要求に変換される。例えば、リモートノード６１３ｂが共有アクセス要求６２６をホームノード６１１ｂへ送信すると、その共有アクセス要求６２６は、排他的アクセス要求６２８に変換される。

【００４７】

一度、排他的アクセス要求６２８が認可されると、ホームノード６１１ｂは、リモートノード６１３ｂのアドレスへのポインタを格納する。そのアクセスは、排他的であるので、共有ベクトルを有する必要はない。換言すれば、当該情報への共有アクセスを要求する各ノード毎のビットを潜在的に格納しなければななない代わりに、当該リモートノード６１３ｂの位置を参照するホームノード６１１ｂによって、２進数番号（またはノード番号またはＣＰＵ番号）が格納される。

【００４８】

ある実施例において、ローカルノード６１１ｉからの後続の共有アクセス要求６２２が、リモートノード６１３ｂの排他的アクセスを終了する。次に、ローカルノード６１１ｉは、当該情報への共有アクセスまたは排他的アクセスを認可される。

【００４９】

以下は、本発明の実施例にかかわる方法５１０の記載である。下記に説明のために、ＤＳＭ６００に関する方法５１０示す。しかし、方法５１０は、ＤＳＭ６００には限定されない。ここにある記述に基づいて、当業者は、方法５１０が、その他のＤＳＭシステムにおいても実現できることを理解されるであろう。

【００５０】

図５ａは、本発明の実施例にかかわるキャッシュコヒーレンス維持の方法５１０を示すフローチャートである。方法５１０はステップ５１２から始まり、ここで前述のように共有アクセス要求が、情報を保持するノード、つまりホームノードによって受信される。ある実施例では、ホームノードは、主記憶の一部を含むＤＳＭマルチプロセッサシステムの任意のノードである。ホームノードの主記憶装置は、マルチプロセッサシステム内の種々

のノードによって要求される情報を含む。このような情報は、当該システム中の他のノードが実行している種々の機能および手順の実行を要求することがある。要求は、当該主記憶の位置から読まれたり書かれたりする。あるノードが、ホームノードの主記憶装置に格納されている特定の情報を使用する必要がある場合、このようなノードは、そのような情報を求めてアクセス要求をホームノードへ送信する。このアクセス要求は、共有アクセス要求または排他的アクセス要求のどちらでもよい。

【００５１】

ステップ５１４において、共有アクセスがローカルノード（ローカルリクエスタ）が発生したかまたはリモートノード（リモートリクエスタ）発生したかが判断される。ある実施例では、ステップ５１４は、リクエスタのアドレスと要求情報のアドレスとの比較に基づいて行われる。あるいは、ホームノードがシステム内のノードのアドレス表を格納していて、このアドレス表ではアドレスをノードの近接パラメータによってソートしていることができる。近接パラメータによって、共有アクセを要求するノードが、ホームノードに対して、リモートノードであるかローカルノードであるかが判断される。リクエスタノードが共有アクセスの要求を送信すると、ホームノードは、要求元ノードのアドレスを読み、要求元ノードのアドレスとアドレス表とを比較する。要求元ノードのアドレスがアドレス表のアドレスと一致すると、要求元ノードは、ローカルノードと判断されて、共有アクセス要求が、共有アクセス要求として処理される。要求元ノードのアドレスが、アドレス表に格納されているアドレスと一致しない場合は、その要求元ノードは、リモートノードと判断されて、その共有アクセス要求は、排他的アクセス要求に変換される。

【００５２】

他の実施例では、要求元ノードのアドレス上位ビットとホームノードのアドレス上位ビットとの比較が行われる。このビットが一致した場合は、要求元ノードローカルノードと判断されて、共有アクセス要求が共有アクセス要求として処理される。一致しない場合は、要求元ノードはリモートノードと判断されて、その共有アクセス要求は排他的アクセス要求に変換される。

【００５３】

ステップ５１４において、共有アクセス要求がローカルノードから発生したと判断されると、処理はステップ５１６へ進み、ここではその共有アクセス要求は共有アクセス要求として処理される。共有アクセス要求が認可された場合は、大体、共有ベクトルが生成され、および／または更新されてその要求元ローカルノードは要求情報への共有アクセスを有することが識別される。

【００５４】

ステップ５１４では、共有アクセス要求がリモートノードから発生したと判断されると、処理はステップ５１８へ進み、ここではその共有アクセス要求は排他的アクセス要求に変換される。これを図６に示す。ここでは、共有アクセス要求６２６が排他的アクセス要求６２８に変換されている。次に、排他的アクセス要求は、排他的アクセス要求手順、例えばここで記述したようにステップ５２０および／またはステップ５２４などに従って処理される。

【００５５】

例えば、その要求情報が他のリクエスタの間で共有されいる場合、および当該リモートノードに排他的アクセスが与えられている場合には、ステップ５２０において、当該要求情報のコピーを以前にキャッシュした他のノードへ、ホームノードは無効命令を送信する。これによって、そのリモートノードがその情報への排他的アクセスを有している間は、他のプロセスによって、その情報のキャッシュされたコピーが使用されるのが防止される。

【００５６】

そのリモートノードは、排他的アクセスを認可されているので共有ベクトルは必要ない。その代わりにステップ５２４において、ポインタが格納されて要求情報への排他的アクセスが認可されたリモートノードを識別する。このポインタは、共有ベクトルの代わりに

キャッシュコヒーレンス機構で使用される。このようにして、本発明はローカルノードの数に基づいた共有ベクトルの発生しうる大きさを制限する。換言すれば、ローカルノードと共にリモートノードを潜在的に識別する大きな共有ベクトルを有するという問題が、大幅に解消される。

【００５７】

図５ｂにおいて、本発明の別の実施例の方法５１０を表示している。この実施例では、リモートノードが、ホームノードによって排他的アクセス要求を認可された場合でも、図５ａのステップ５１２から５２４に記載されているように、ローカルノードはリモートノードの排他的アクセス権を終了することができる。これを図６に示す。ここでは、リモートノード６１３ｂは排他的アクセスが認可され、続いてローカルノード６１１ｉが共有アクセスを要求する。

【００５８】

図５ｂにおいて、ステップ５３０で方法５１０は、アクセス要求を受信する。次に、方法５１０はステップ５３２に進む。ここで、受信したアクセス要求が排他的要求かどうが判定処理される。ステップ５３２において、アクセス要求が排他的アクセス要求でなければ（すなわち、共有アクセス要求であれば）、ステップ５３４において、その共有アクセス要求がリモートノードから発生したかどうかが判定処理される。ステップ５３４は、前述したように実行することができる。例えば、アドレス比較を行うことができる。

【００５９】

要求が排他的要求（ステップ５３２で判断される）またはリモートノードから発生する場合は（ステップ５３４で判断される）、本方法では、処理がステップ５３６に進む。ここでは、要求情報が他のリクエスタによって排他的に保持されているかどうかが判定処理される。ステップ５３６において、要求情報が現時点で他のリクエスタによって排他的に保持されている場合は、ステップ５４８に示すように、介入要求が当該情報の排他的所有者に送信される。

【００６０】

ステップ５４８の機能は、現在の排他的所有者が情報を修正した場合には、新しいリクエスタが排他的アクセスを認可される前に、修正情報がホームノードへ返却されることを保証する。一度、要求情報の排他的所有者によって介入要求が送信されると、処理はステップ５５０に進む。

【００６１】

ステップ５５０において、新しいリクエスタ（例えば、リモートノード）は、要求情報への排他的アクセスを認可される。例えば図６において、ノード６１３ｂは、ノード６１１ｂに位置している要求情報への排他的アクセスを有している新しいリクエスタであってもよい。

【００６２】

新しいリクエスタが、現在、要求情報への排他的アクセスを有することを表示するために、ステップ５５２において、ホームノードによって、当該リモートリクエスタに排他的アクセスが与えられたことを示すポインタが格納される。

【００６３】

ステップ５３６に戻って、当該要求情報は他のリクエスタによって現在、排他的に保持されていないとシステムが判断すると、ステップ５５４に示すように、本方法では同一情報への共有アクセスを現在保持している全ノードへ無効化命令を送信する。次に、処理は前述のようにステップ５５０および５５２進む。

【００６４】

ステップ５３２に戻って、受信したアクセス要求が排他的要求ではない場合は、処理はステップ５３４へ進む。また、ステップ５３４に戻って、受信したアクセス要求がリモートノードから発生したのではないと判定処理された場合は、処理はステップ５３８へ進む。

【００６５】

ステップ５３８において（ステップ５３６と同様）、当該要求情報が、現在、他のリクエスタによって排他的に保持されているかどうかを判定処理する。ステップ５３８において、要求情報が現在、他のリクエスタによって排他的に保持されてない場合には、処理はステップ５４４へ進む。

【００６６】

ステップ５４４はステップ５５０と同様であり、ここではリクエスタは、要求情報へのアクセスを認可される。しかし、ステップ５４４では、リクエスタはローカルノード（ステップ５３４で判断される）であり、したがって、要求情報へ共有アクセスが与えられる。

【００６７】

ステップ５４４に続くステップ５４６において、新しい共有ベクトルは、要求情報が位置しているノードによって格納される。例えば図６において、ローカルノード６１１ｉが、ノード６１１ｂに位置している要求情報への共有アクセスを要求すると、ノード６１１ｂは新しい共有ベクトルを格納する。

【００６８】

ステップ５３８に戻って、要求情報が現在、他のリクエスタによって排他的に保持されていると判定処理されると、処理はステップ５４０に進む。ステップ５４０において、リクエスタの位置に対して、ホームノードがリモートに位置付けられているかどうかが判定処理される。要求情報の現在の排他的所有者が、ホームノードに対して、リモートであると判断されると、処理はステップ５４８へ進む。ステップ５４８およびこれに続くステップは上記のとおりである。

【００６９】

ステップ５４０において、ホームノードに対してリクエスタがローカルであると判定処理されると、処理はステップ５４２に進む。ステップ５４２（ステップ５４８と同様）において実行される機能によって、現在の排他的所有者が情報を修正した場合は新しいリクエスタが排他的アクセスを認可される前に当該修正情報がホームノードへ返却されることが保証される。ステップ５４２において、ローカルリクエスタは当初のリクエスタの排他的情報アクセス終了する。当初のリクエスタの情報要求は、共有アクセス要求となる。

【００７０】

一度、介入要求が、要求情報の排他的所有者へ送信されると、処理はステップ５４４に進む。ステップ５４４および続くステップ５４６については前述されている。ステップ５４４において、情報アクセスはローカルリクエスタに与えられ、ステップ５４６において共有ベクトルがシステムによって格納される。

【００７１】

リモートノードが、さらに当該情報へアクセスする必要がある場合は、再度、ステップ５３０を実行するなどによって、その情報を再要求せねばならない。あるいは、ローカルリクエスタと共に当該情報へ共有アクセスできるように、リモートノードの排他的アクセスを共有アクセスに置き換える。

【００７２】

上述の方法５１０は、ディレクトリに基づくプロトコルに実装することができるが、他の実装も可能である。

【００７３】

（４．結論）

本発明の方法、システム、構成要素の実施事例を、ここに記載した。他の所で言及したように、これらの実施事例は説明だけの目的で記載されたもので、限定するものではない。その他の実施例も可能であり、本発明によって包含される。このような実施例は、ここに含まれている教示に基づいて、関連技術の当業者にとっては明白である。よって、本発明の広さと範囲は上記の実施事例によって限定されるべきものではなく、以下の請求項およびその等価物に従ってのみ定義されるべきである。

【図面の簡単な説明】

【００７４】
本発明は、添付の図面を参照して記載する。各図面において、同一の参照番号は、同一の要素または類似機能の要素を示す。さらに、参照番号の左端の数字は、その参照番号が最初に現れる図面を示す。
【図１】図１は、単一プロセッサシステムのブロック図である。
【図２】図２は、集中対称型共有記憶マルチプロセッサ（ＳＭＰ）システムのブロック図である。
【図３】図３は、分散共有記憶マルチプロセッサシステム（ＤＳＭ）のブロック図である。
【図４】図４は、図１から図３に示すプロセッサシステムのいずれでも使用可能なプロセッサとキャッシュのブロック図である。
【図５ａ】図５ａは、ＤＳＭマルチプロセッサシステムにおけるキャッシュコヒーレンスを保持するための方法の実施例のフローチャートである。
【図５ｂ】図５ｂは、ＤＳＭマルチプロセッサシステムにおけるキャッシュコヒーレンスを保持するための方法の別の実施例のフローチャートである。
【図６】図６は、本発明を適用できるＤＳＭマルチプロセッサシステムの実施例のブロック図である。

【図１】

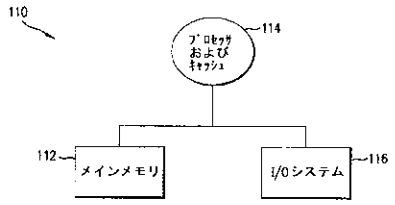


FIG. 1

【図２】

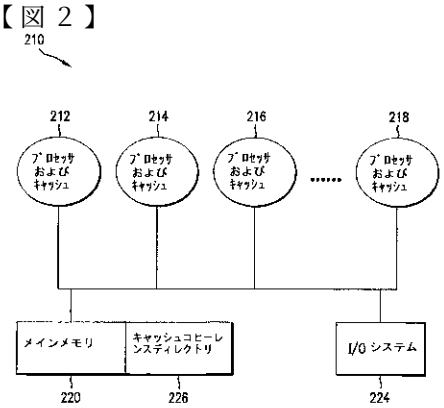
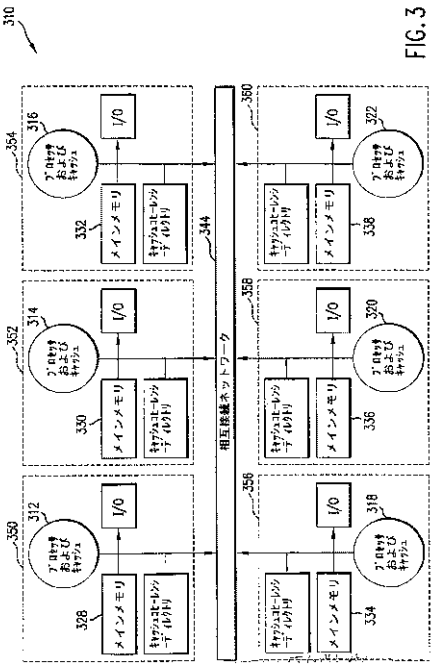


FIG. 2

【図３】



【図４】

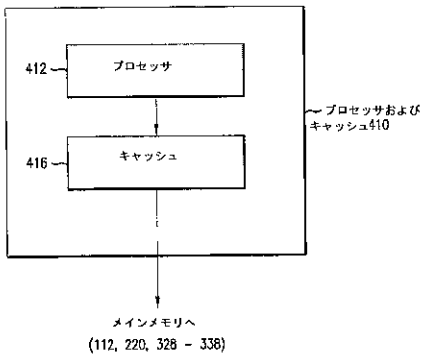


FIG. 4

【図５ａ】

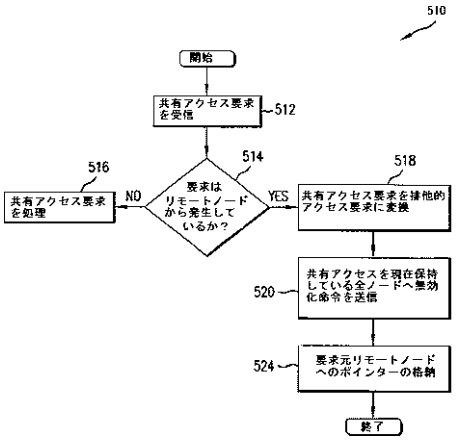


FIG. 5a

【図５ｂ】

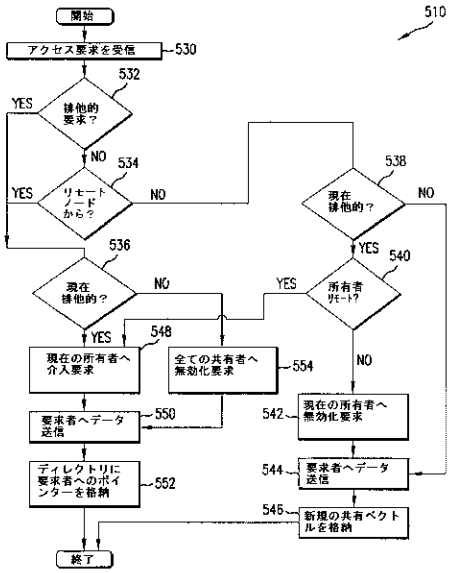


FIG. 5b

【図６】

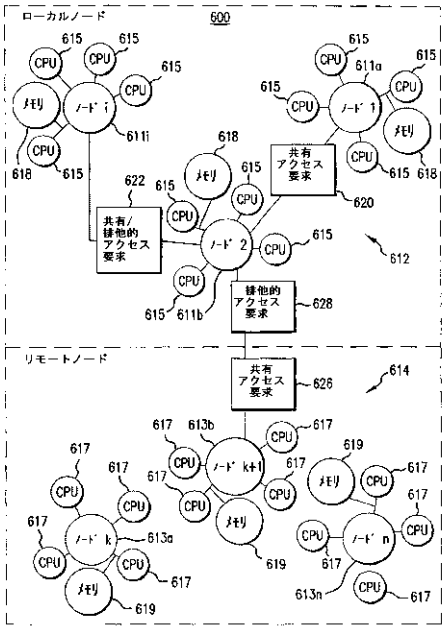


FIG. 6

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US02/05779

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 12/00
US CL : 711/147, 141, 148, 152, 153, 154
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 711/147, 141, 148, 152, 153, 154

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WEST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,P	US 6,370,622 B1 (CHIOU et al.) 9 APRIL 2002, entire	1-41
A	US 6,026,474 A (CARTER et al.) 15 February 2000, entire	1-41
A	US 5,940,860 A (HAGERSTEN et al.) 17 AUGUST 1999, entire	1-41
A	US 5,617,577 A (YAMADA et al.) 1, APRIL 1997, entire	1-41
A	US 5,592,625 A (SANDBERG) 7 JANUARY 1997, entire	1-41

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 03 JUNE 2002	Date of mailing of the international search report 25 JUN 2002
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer KIMBERLY MCLEAN Telephone No. (703) 308-3900

Form PCT/ISA/210 (second sheet) (July 1998)★

フロントページの続き

(72)発明者　デネロフ，　マーティン　エム.
アメリカ合衆国　ニュージャージー　０７７５５，　オークハースト，　オールド　ファーム　ロード　２４

Ｆターム(参考) 5B005 KK02　MM01　PP11
5B060 KA01　KA02　KA06