(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2009-211167
(P2009-211167A)
(43)公開日　平成21年9月17日(2009.9.17)

(51)Int.Cl.		F I		テーマコード（参考）
G06F 9/52	(2006.01)	G06F 9/46	472Z	5B045
G06F 15/167	(2006.01)	G06F 15/167	610B	

審査請求 未請求　請求項の数 5　ＯＬ　（全 22 頁）

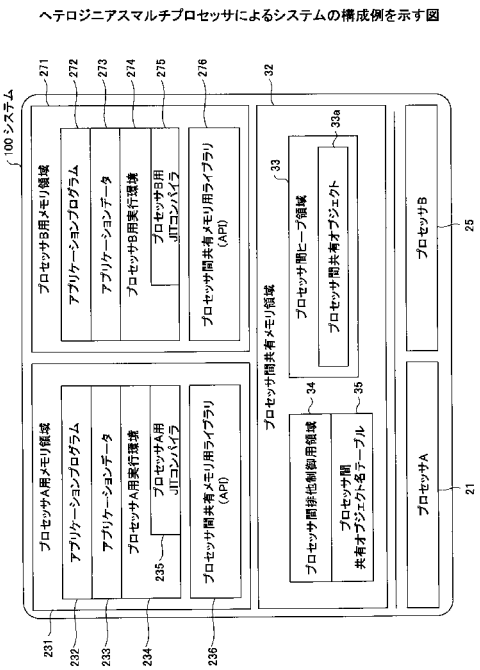
(21)出願番号　特願2008-50953(P2008-50953)
(22)出願日　平成20年2月29日(2008.2.29)

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100070150
弁理士　伊東　忠彦
(72)発明者　上方　輝彦
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

Ｆターム(参考)　5B045 DD03　EE03　EE36　JJ33

(54)【発明の名称】プログラム実行システム

(57)【要約】
【課題】本発明の課題は、組み込みヘテロマルチプロセッサシステム上でのプロセッサ間で連携動作する中間コードを用いるアプリケーションプログラムを高速に実行可能とすることを目的とする。
【解決手段】上記課題は、複数のプロセッサを含み、前記複数のプロセッサ上で動作可能な中間コードを実行するプログラム実行環境を備えたプログラム実行システムであって、前記複数のプロセッサ夫々に専用の複数のプロセッサ専用メモリと、前記複数のプロセッサ間で共有される前記中間コードによって操作される共有オブジェクトを格納するプロセッサ間共有メモリとを備え、前記複数のプロセッサの各々は、前記共有オブジェクトを指定する名前と前記プロセッサ間共有メモリ内の該共有オブジェクトの実体とを対応させることによって、前記複数のプロセッサ専用メモリの各々と該プロセッサ間共有メモリ間にて該共有オブジェクトを読み出し及び書き込みする読出書込機能を実行することにより達成される。
【選択図】図６



【特許請求の範囲】
【請求項１】
複数のプロセッサを含み、前記複数のプロセッサ上で動作可能な中間コードを実行するプログラム実行環境を備えたプログラム実行システムであって、
前記複数のプロセッサ夫々に専用の複数のプロセッサ専用メモリと、
前記複数のプロセッサ間で共有される前記中間コードによって操作される共有オブジェクトを格納するプロセッサ間共有メモリとを備え、
前記複数のプロセッサの各々は、
前記共有オブジェクトを指定する名前と前記プロセッサ間共有メモリ内の該共有オブジェクトの実体とを対応させることによって、前記複数のプロセッサ専用メモリの各々と該プロセッサ間共有メモリ間にて該共有オブジェクトを読み出し及び書き込みする読出書込機能を実行するようにしたプログラム実行システム。
【請求項２】
前記プロセッサ間共有メモリは、前記共有オブジェクトの名前と該共有オブジェクトの実体とを対応させた対応テーブルを備え、
前記複数のプロセッサの各々は、前記共有オブジェクトの名前と該共有オブジェクトの実体との対応を前記対応テーブルに書き込むことにより該共有オブジェクトを登録し、また該登録された共有オブジェクトを該対応テーブルから削除することにより該共有オブジェクトを削除する登録削除機能を実行するようにした請求項１記載のプログラム実行システム。
【請求項３】
前記複数のプロセッサの各々は、
前記プロセッサ間共有メモリのメモリ領域の割り当て及び開放を管理する共有メモリ管理機能と、
前記複数のプロセッサ間の前記プロセッサ間共有メモリへのアクセスに対してセマフォ又はロック機能を用いて排他制御を行うプロセッサ間排他制御機能と、
前記プロセッサ間共有メモリのヒープ領域に前記共有オブジェクトを格納するための共有メモリヒープ機能とを実行するようにした請求項１又は２記載のプログラム実行システム。
【請求項４】
前記プロセッサ間共有メモリのメモリ領域と前記複数のプロセッサ専用メモリの各々のメモリ領域との間のダイレクトメモリアクセスによるデータ転送を制御するＤＭＡコントローラを更に備え、
前記複数のプロセッサの各々において、前記ＤＭＡコントローラを用いて前記高速データ転送を行うルーチンと該ルーチンを前記プログラム実行環境から呼び出すためのインターフェイスを備えるようにした請求項１乃至３のいずれか一項記載のプログラム実行システム。
【請求項５】
前記プロセッサ間排他制御機能を有するハードウェアを備え、
前記複数のプロセッサの各々において、前記ハードウェアのプロセッサ間排他制御機能を用いて排他制御を行うルーチンと該ルーチンを前記プログラム実行環境から呼び出すためのインターフェイスを備えるようにした請求項１乃至３のいずれか一項記載のプログラム実行システム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、組み込みヘテロマルチプロセッサシステム上でのプロセッサ間で連携動作する中間コードを用いるアプリケーションプログラムを高速に実行可能とするプログラム実行システムに関する。
【背景技術】

【０００２】
近年、情報処理装置を組み込んだ電子機器の多様化及びそれら電子機器間を通信接続するネットワーク化に伴い、多数の異種プロセッサが開発されると共に、ネットワーク化もあらゆる電子機器間で実現されるようになってきた。多様化した電子機器の開発において、プロセッサに依存したプログラム開発を行うのでは開発者の負担が大きい。そのため、ハードウェアやＯＳ（Operation System）に依存しない仮想マシンを構築し、ハードウェアが解釈するネイティブコードより抽象度の高い中間コードで表現されるプログラムを仮想マシン上で実行することで、ネットワークを介してプログラムの授受を可能とし、かつプロセッサ毎のプログラム開発の負担を低減することができるようになった。
【０００３】
更に、仮想マシンを備えないプロセッサへネットワークを介してプログラムを転送するために、仮想マシンを備えたプロセッサが仮想マシンを備えないプロセッサ向けのネイティブコードに変換することが提案されている（例えば、特許文献１及び特許文献２参照。）。また、プロセッサ毎のネイティブコードへ変換する負荷を低減するために、各プロセッサがネイティブトランスレータを備えることが提案されている（例えば、特許文献３参照。）。
【特許文献１】特開２０００－１７２５０９号公報
【特許文献２】特開２００５－１０８１２６号公報
【特許文献３】特表２００３－５０８８４４号公報
【発明の開示】
【発明が解決しようとする課題】
【０００４】
従来の疎結合のヘテロジニアスマルチプロセッサによるシステム１は、例えば図１に示すように、異なるアーキテクチャを持つプロセッサＡ１１及びＢ１５を備え、プロセッサＡ１１にはプロセッサＡ専用ＬＡＮ－ＩＦ１２とプロセッサＡ専用メモリ１３とが各々バスＢで接続され、同様にプロセッサＢ１５にはプロセッサＢ専用ＬＡＮ－ＩＦ１６とプロセッサＢ専用メモリ１７とが各々バスで接続される。プロセッサＡ及びＢ間のデータ転送は、例えば、ＬＡＮ１９を介して行われる。
【０００５】
また、このようなシステム１は、例えば図２に示すような階層構造を成す。プロセッサＡ専用メモリ１３には、アプリケーションプログラム、アプリケーションデータ、プロセッサＡ用実行環境、ＲＭＩ（Remote Method Invocation）、プロセッサＡ用ＪＩＴ（Just In Time）コンパイラ、ＬＡＮドライバなどがプロセッサＡ用メモリ領域１３ａに格納されている。プロセッサＢ専用メモリ１７においても同様に、アプリケーションプログラム、アプリケーションデータ、プロセッサＢ用実行環境、ＲＭＩ、プロセッサＢ用ＪＩＴコンパイラ、ＬＡＮドライバなどがプロセッサＢ用メモリ領域１７ａに格納されている。アプリプログラムを上位層としＬＡＮドライバを下位層とした階層構造に従った機能構成を成す。各プロセッサＡ１１及びプロセッサＢ１５は、それぞれのプロセッサＡ専用メモリ１３及びプロセッサＢ専用メモリ１７からコード又はデータを読み出して所定機能を実行する。
【０００６】
アプリケーションプログラムがＪａｖａ（登録商標）言語によるプログラムである場合、図２に示すプロセッサＡ用実行環境及びプロセッサＢ用実行環境は、図３に示すようにＪａｖａ（登録商標）ＶＭ（Virtual Machine）として機能する。Ｊａｖａ（登録商標）ＶＭは、クラスローダー、クラス・キャッシュ機能、ヒープメモリ管理機能、ガーベジ・コレクション（ＧＣ）機能を備え、Ｊａｖａ（登録商標）プログラムの実行時には、ＰＣレジスタ、メソッド領域、ヒープ領域、Ｊａｖａ（登録商標）ＶＭスタック、実行時定数プール、Ｎａｔｉｖｅメソッドスタック等の領域がメモリ内に展開される。メソッド領域にはクラスローダーによってクラスがロードされ、ヒープ領域にはヒープメモリ管理機能によってクラスがインスタンス化したオブジェクトが格納される。Ｊａｖａ（登録商標）

ＶＭスタックには、関数が1つ呼ばれるたびにフレームがスタックされ、フレーム内にはローカル変数及びオペランドスタックの領域が構成される。このような実行環境にて実行されるプログラムを中間コードで表現されたプログラムという。

【０００７】

同種のプロセッサ間で、かつ対称型マルチプロセシング（Symmetric Multi-Processing(SMP)）ならば、POSIX（Portable Operating System Interface for UNIX（登録商標））スレッドによってスレッド化してメモリを共有できるが、図１に示すようなシステム１では、異種のプロセッサＡ１１及びＢ１５間ではスレッド化してメモリを共有することができない。

【０００８】

Ｊａｖａ（登録商標）言語、Ｃ＃言語などのプロセッサ非依存の中間コードによる処理では、複数のＪａｖａ（登録商標）ＶＭ間のメモリ空間は共有されない仕組みであるため、プロセッサＡがプロセッサＢのＪａｖａ（登録商標）ＶＭが管理するメモリにはアクセスできない。

【０００９】

ヘテロジニアスマルチプロセッサによるシステム１での負荷分散のために各プロセッサ用の実行形式モジュールを持たせると、アプリケーションプログラムのＲＯＭ容量が多く必要となる。また、異種プロセッサＡ１１及びＢ１５毎にプログラムを作成し管理するのは、開発者にとって負担となる。

【００１０】

ヘテロジニアスマルチプロセッサ間のデータ転送にＬＡＮ通信層を使うとデータコピーが多くなり内部バス間転送より時間を要する。また、Ｊａｖａ（登録商標）やＣ＃言語などで書かれたアプリケーションプログラムを用いた場合、ヘテロジニアスマルチプロセッサ間では、ＲＭＩなどのようにリモートオブジェクトとして登録して、ＬＡＮ等を介して別のＪａｖａ（登録商標）のオブジェクトのメソッドを呼ぶ仕組みであるため、ヘテロジニアスマルチプロセッサ間のＬＡＮ等を用いた（ＴＣＰ／ＩＰによる）データ転送には多くのデータ転送回数が必要になる。そのため、ヘテロジニアスマルチプロセッサを１チップ化したとしても、それ自身の性能を生かすことができないという問題があった。

【００１１】

更に、非対称型マルチプロセシング（Asymmetric Multi-Processing(AMP)）での異種プロセッサ上のＯＳからは、異種ＰＥ間での共有ファイルシステムをＮＦＳなどのように片方のＯＳに依頼するなどによって行われるため、高速にアクセスできない。

【００１２】

よって、本発明の目的は、ヘテロジニアスマルチプロセッサでアプリケーションプログラムを共通化してプログラム開発の負荷軽減を可能とし、ヘテロジニアスマルチプロセッサ上で負荷分散できるようにしつつ、ヘテロマルチプロセッサ間のデータ転送を削減することで、ヘテロジニアスマルチプロセッサ間で協調動作するプログラムを高速化することを提供することである。

【課題を解決するための手段】

【００１３】

上記課題を解決するため、本発明は、複数のプロセッサを含み、前記複数のプロセッサ上で動作可能な中間コードを実行するプログラム実行環境を備えたプログラム実行システムであって、前記複数のプロセッサ夫々に専用の複数のプロセッサ専用メモリと、前記複数のプロセッサ間で共有される前記中間コードによって操作される共有オブジェクトを格納するプロセッサ間共有メモリとを備え、前記複数のプロセッサの各々は、前記共有オブジェクトを指定する名前と前記プロセッサ間共有メモリ内の該共有オブジェクトの実体とを対応させることによって、前記複数のプロセッサ専用メモリの各々と該プロセッサ間共有メモリ間にて該共有オブジェクトを読み出し及び書き込みする読出書込機能を実行するように構成される。

【００１４】

このようなプログラム実行システムにおいて、前記プロセッサ間共有メモリは、前記共有オブジェクトの名前と該共有オブジェクトの実体とを対応させた対応テーブルを備え、前記各プロセッサは、前記共有オブジェクトの名前と該共有オブジェクトの実体との対応を前記対応テーブルに書き込むことにより該共有オブジェクトを登録し、また該登録された共有オブジェクトを該対応テーブルから削除することにより該共有オブジェクトを削除する登録削除機能を実行するように構成してもよい。

【発明の効果】

【００１５】

本願発明は、ヘテロジニアスマルチプロセッサでアプリケーションプログラムを共通化してプログラム開発の負荷軽減を可能とし、ヘテロジニアスマルチプロセッサ上で負荷分散できるようにしつつ、ヘテロマルチプロセッサ間のデータ転送を削減することで、ヘテロジニアスマルチプロセッサ間で協調動作するプログラムを高速化することができる。

【発明を実施するための最良の形態】

【００１６】

以下、本発明の実施の形態を図面に基づいて説明する。

【００１７】

図４は、本発明の第一実施例に係る密結合のヘテロジニアスマルチプロセッサによるシステム例を示す図である。図４において、システム１００は、密結合されたヘテロジニアスマルチプロセッサのシステムであり、プロセッサＡ２１と、プロセッサＡ専用ＬＡＮ－ＩＦ２２と、プロセッサＡ専用メモリ２３と、プロセッサＢ２５と、プロセッサＢ専用ＬＡＮ－ＩＦ２６と、プロセッサＢ専用メモリ２７と、プロセッサ間共有メモリ３１とを有し、それらはバス３０で互いに接続される。プロセッサＡ２１とプロセッサＢ２５とは異種プロセッサであり、ハードウェアに依存しない中間コードで書かれたアプリケーションプログラムを実行できる実行環境を提供する。例えば、中間コードはＪａｖａ（登録商標）言語であり、実行環境はＪａｖａ（登録商標）ＶＭである。

【００１８】

プロセッサＡ専用ＬＡＮ－ＩＦ２２とプロセッサＡ専用メモリ２３とは、プロセッサＡ２１によってのみ使用されるデバイスである。同様に、プロセッサＢ専用ＬＡＮ－ＩＦ２６と、プロセッサＢ専用メモリ２７とは、プロセッサＢ２５によってのみ使用されるデバイスである。

【００１９】

プロセッサ間共有メモリ３１は、プロセッサＡ２１とプロセッサＢ２５との間でアクセス可能な共有メモリである。プロセッサＡ２１は、プロセッサ間共有メモリ３１をアクセスすることによって、プロセッサＡ２１とプロセッサＢ２５とで共有に使用されるオブジェクトをプロセッサ間共有メモリ３１からプロセッサＡ専用メモリ２３へと読み込んだり、プロセッサＡ専用メモリ２３からプロセッサ間共有メモリ３１へと書き込んだりする操作を高速に行うことができる。同様に、プロセッサＢ２５は、プロセッサ間共有メモリ３１をアクセスすることによって、プロセッサＡ２１とプロセッサＢ２５とで共有に使用されるオブジェクトをプロセッサ間共有メモリ３１からプロセッサＢ専用メモリ２７へと読み込んだり、プロセッサＢ専用メモリ２７からプロセッサ間共有メモリ３１へと書き込んだりする操作を高速に行うことができる。

【００２０】

プロセッサ間共有メモリ３１は、図５に示すようにメモリ領域を構成する。図５は、プロセッサ間共有メモリの領域構成例を示す図である。

【００２１】

図５において、プロセッサ間共有メモリ３１は、主に、プロセッサ間共有メモリ領域３２とを備える。更に、プロセッサ間共有メモリ領域３２は、プロセッサ間ヒープ領域３３と、プロセッサ間排他制御用領域３４と、プロセッサ間共有オブジェクト名テーブル３５とを備える。

【００２２】

プロセッサ間ヒープ領域３３は、プロセッサ間共有オブジェクト名テーブル３５に登録されているオブジェクトが格納され、プロセッサＡ２１又はプロセッサＢ２５によってオブジェクトが書き込まれたり、読み出されたりする領域である。プロセッサ間排他制御用メモリ領域３４では、プロセッサ間ヒープ領域３３へのアクセスに係る排他制御を行うためのプロセッサ間セマフォ３４ａ又はプロセッサ間ロック３４ｂが管理される。

【００２３】

プロセッサ間共有メモリ３１に、プロセッサ間排他制御用領域３４を備えて、各プロセッサＡ２１及びプロセッサＢ２５からアトミックアクセスさせることで、ＬＡＮ－ＩＦやＬＡＮドライバを経由しないで、排他制御を行うことでプロセス間共有メモリ領域３２を使用するときの排他制御を高速化することができる。

【００２４】

図６は、ヘテロジニアスマルチプロセッサによるシステムの構成例を示す図である。図６に示すシステム１００では、プロセッサＡ専用メモリ２３には、アプリケーションプログラム２３２と、アプリケーションデータ２３３と、プロセッサＡ用実行環境２３４と、プロセッサＡ用ＪＩＴ（Just In Time）コンパイラ２３５と、プロセッサ間共有メモリ用ライブラリ２３６とがプロセッサＡ用メモリ領域２３１に格納され、プロセッサＢ専用メモリ２７には、アプリケーションプログラム２７２と、アプリケーションデータ２７３と、プロセッサＢ用実行環境２７４と、プロセッサＢ用ＪＩＴコンパイラ２７５と、プロセッサ間共有メモリ用ライブラリ２７６とがプロセッサＢ用メモリ領域２７１に格納されている。アプリプログラムを上位層とし実行環境の一部及びＪＩＴコンパイラを下位層とした階層構造に従った機能構成を成す。

【００２５】

各プロセッサＡ２１及びプロセッサＢ２５は、それぞれのプロセッサＡ専用メモリ２３及びプロセッサＢ専用メモリ２７からコード又はデータを読み出して所定機能を実行する。

【００２６】

プロセッサ間共有メモリ用ライブラリ２３６とプロセッサ間共有メモリライブラリ２７６とは、図７に示すようにプロセッサ間共有メモリ管理機能３６及びプロセッサ間排他制御機能３７とを備え、それらを用いて共有メモリ管理及び共有メモリ排他制御とを提供する種々ルーチンを備えると共に、それらルーチンを呼び出すためのＡＰＩ（Application Program Interface）を提供する。

【００２７】

プロセッサ間共有メモリ管理機能３６は、自身のプロセッサＡ２１又はプロセッサＢ２５からのアクセスに応じて、プロセッサ間共有メモリ３１のプロセッサ間ヒープ領域３３への割り当て及び開放等のメモリ管理を行う。プロセッサ間排他制御機能３７は、自身のプロセッサＡ２１又はプロセッサＢ２５からのプロセッサ間共有メモリ３１のプロセッサ間ヒープ領域３３へのアクセスに対して排他制御を行うためには、プロセッサ間ヒープ領域３３の全体に対して１以上のプロセッサ間セマフォ３４ａ又はプロセッサ間ロック３４ｂをプロセッサ間排他制御用領域３４に設ける。

【００２８】

プロセッサ間共有メモリ用ライブラリ２３６はプロセッサＡ用実行環境２３４にＡＰＩを提供し、プロセッサＡ用実行環境２３４はプロセッサ間共有メモリ用ライブラリ２３６から提供される共有メモリ管理及び共有メモリ排他制御を実行する種々ルーチンを用いて、プロセッサＡ２１によるプロセッサ間共有メモリ３１へのアクセスを高速に行うことを可能とする。同様に、プロセッサ間共有メモリライブラリ２７６はプロセッサＢ用実行環境２７４にＡＰＩを提供し、プロセッサＢ用実行環境２７４はプロセッサ間共有メモリライブラリ２７６から提供される共有メモリ管理及び共有メモリ排他制御を実行する種々ルーチンを用いて、プロセッサＢ２１によるプロセッサ間共有メモリ３１へのアクセスを高速に行うことを可能とする。

【００２９】

プロセッサ間共有メモリ領域３２では、プロセッサ間共有メモリ用ライブラリ２３６とプロセッサ間共有メモリライブラリ２７６とが夫々のプロセッサＡ用実行環境２３４とプロセッサＢ用実行環境２７４とに提供する種々ルーチンによって、プロセッサ間ヒープ領域３３と、プロセッサ間排他制御用領域３４と、プロセッサ間共有オブジェクト名テーブル３５とがプロセッサＡ２１とプロセッサＢ２５とに共有可能となる。

【００３０】

図８は、各プロセッサ用実行環境とプロセッサ間共有メモリ用ライブラリとによる機能構成を示す図である。図８において、例えば、各プロセッサ用実行環境２３４及び２７４がＪａｖａ（登録商標）ＶＭ５４を提供する場合、Ｊａｖａ（登録商標）ＶＭ５４は、クラスファイルをロードしてメソッド領域６３に格納するクラスローダー５４ａと、クラスをキャッシュして高速に実行するクラス・キャッシュ機能５４ｂと、ヒープメモリ管理機能５４ｃと、ヒープ領域６４の割当及び開放を管理するガーベジ・コレクション（ＧＣ）機能５４ｄとを備える。Ｊａｖａ（登録商標）ＶＭ５４は、実行時データ領域６１に示される。

【００３１】

実行時データ領域６１には、Ｊａｖａ（登録商標）ＶＭ５４によって、実行中の命令アドレスを示すプログラムカウンタとして使用されるＰＣレジスタ６２と、クラスファイルが格納されるメソッド領域６３と、クラスがインスタンス化されたオブジェクトを格納するヒープ領域６４と、一つのスレッドを実行するための各メソッドに対応するフレームを格納するＪａｖａ（登録商標）ＶＭスタック６５と、各クラスで使用される定数を格納する実行時定数プール６６と、ネイティブコードのメソッドを実行するためのＮａｔｉｖｅメソッドスタック６７とが展開される。

【００３２】

このように、システム１００の構成において、プロセッサＡ２１側でプロセッサＡ２１専用のＪａｖａ（登録商標）プログラムを実行する場合には、プロセッサＡ用実行環境２３４は、プロセッサＡ専用メモリ２３を用いてＪａｖａ（登録商標）ＶＭ５４を提供する。プロセッサＢ２５側でプロセッサＢ２５専用のＪａｖａ（登録商標）プログラムを実行する場合においても同様である。

【００３３】

プロセッサＡ２１側のアプリケーションプログラム２３２及びプロセッサＢ２５側のアプリケーションプログラム２７２がプロセッサ間で共有される共有クラスからインスタンス化したプロセッサ間共有オブジェクトを実行するために、各プロセッサ用実行環境２３４及び２７４は、夫々のプロセッサ間共有メモリ用ライブラリ２３６及び２３７を用いて、更にプロセッサ間中間コード実行環境７０が提供される。プロセッサ間中間コード実行環境７０において、共有メモリ３１と専用メモリ２３１又は２７１（以下、単に専用メモリと言う）間のプロセッサ間データ転送機能７９が行われる。

【００３４】

プロセッサ間中間コード実行環境７０は、プロセッサ間共有メモリ・クラス７１と、プロセッサ間共有中間コードのヒープメモリ７２と、プロセッサ間共有オブジェクト操作機能７３と、プロセッサ間共有クラスローダー７４と、プロセッサ間共有クラス・キャッシュ機能７５とを有する。

【００３５】

プロセッサ間共有メモリ・クラス７１は、インスタンス化されるとプロセッサ間共有オブジェクトとなるプロセッサ間共有クラスである。プロセッサ間共有中間コードのヒープメモリ７２は、プロセッサ間共有メモリ３１のプロセッサ間共有メモリ領域３２内にあるプロセッサ間ヒープ領域３３をプロセッサ間共有メモリ管理機能３６を用いて制御するための機能を備え、Ｊａｖａ（登録商標）ＶＭ５４のヒープメモリ管理機能５４ｃ及びガーベジ・コレクション（ＧＣ）機能５４ｄとは別にプロセッサ間共有メモリ３１用として、ヒープメモリ操作・管理機能７２ａとプロセッサ間共有ガーベジ・コレクション（ＧＣ）機能７２ｂとを有する。

【００３６】
プロセッサ間共有オブジェクト操作機能７３は、プロセッサ間共有メモリ３１が備えるプロセッサ間共有メモリ管理機能３６及びプロセッサ間排他制御機能３７と、プロセッサ間データ転送機能７９とを呼び出すルーチンを使用して共有オブジェクトの操作を行う。これらルーチンは、プロセッサ間共有ライブラリ２３６及び２７６によって提供される。プロセッサ間共有オブジェクト操作機能７３は、専用メモリ又はプロセッサ間共有メモリ３１から読み出すためのリード７３ａと、専用メモリ又はプロセッサ間共有メモリ３１へと書き込むためのライト７３ｂと、プロセッサ間共有オブジェクトを登録するための登録７３ｃと、登録したプロセッサ間共有オブジェクトを削除するための削除７３ｄと、図５に示すプロセッサ間共有オブジェクト名テーブル３５を用いてプロセッサ間共有オブジェクトとその名前とを対応させるオブジェクト対応機能７３ｆとを有する。
【００３７】
プロセッサ間共有クラスローダー７４は、プロセッサ間共有クラスをロードするための機能であり、Ｊａｖａ（登録商標）ＶＭ５４のクラスローダー５４ａとは別に備えられる。プロセッサ間共有クラス・キャッシュ機能７５とは、プロセッサ間共有クラスをキャッシュするための機能であり、Ｊａｖａ（登録商標）ＶＭ５４のクラス・キャッシュ機能５４ｂとは別に備えられる。
【００３８】
プロセッサ間共有メモリ３１には、図５及び図６で説明したように、プロセッサ間共有メモリ管理機能３６と、プロセッサ間排他制御機能３７とを有する。プロセッサ間共有メモリ管理機能３６にはプロセッサ間共有オブジェクト３３ａに対するプロセッサ間ヒープ領域３３のメモリ割り当て／開放部３６ａが備えられる。
【００３９】
プロセッサ間データ転送機能７９は、プロセッサ間共有メモリ３１と専用メモリ間のデータ転送をＬＡＮ－ＩＦやＬＡＮドライバを用いることなく実現する機能であり、引数／戻り値のデータ転送部７９ａ及びデータ転送を高速に行うための高速データ転送部７９ｂとを備える。
【００４０】
上述したようなプロセッサ間中間コード実行環境７０及びプロセッサ間データ転送機能７９が実現する種々の機能は、各プロセッサ間共有メモリライブラリ２３６及び２７６においてルーチン化され提供される。後述されるようにＤＭＡ（Direct Memory Access）コントローラ（ＤＭＡＣ）を備えるシステム構成の場合、ＤＭＡＣのＤＭＡ転送機能を用いたより高速なデータ転送を行うルーチンが提供される。
【００４１】
図９は、本発明の第二実施例に係る密結合のヘテロジニアスマルチプロセッサによるシステム構成例を示す図である。図９に示すシステム１００２は、図４に示すシステム１０００における構成に加えてＤＭＡＣ４１を備える。システム１００２は、更に排他制御機構４２を備えるようにしてもよい。
【００４２】
ＤＭＡＣ４１を使用して高速にデータ転送を行う場合、高速データ転送部７９ｂ（図８）によってプロセッサ間共有メモリ３１からプロセッサＡ専用メモリ２３又はプロセッサＢ専用メモリ２７へと、又は、プロセッサＡ専用メモリ２３又はプロセッサＢ専用メモリ２７からプロセッサ間共有メモリ３１へとプロセッサ間共有オブジェクトに係る情報が転送される。プロセッサ間共有オブジェクトに係る情報は、アプリケーションプログラム２３２又は２７２のプロセッサ間共有オブジェクトへの入力データ、プロセッサ間共有オブジェクトを実行した結果データ等である。高速データ転送部７９ｂは、プロセッサ間共有メモリ用ライブラリ２３６及び２７６が提供するプロセッサ間データ転送機能７９にＤＭＡＣ４１にＤＭＡデータ転送を実行させるためのルーチンとして組み込まれる。
【００４３】
また、図５に示すプロセッサ間共有メモリ３１のプロセッサ間排他制御機能３７とプロ

セッサ間排他制御用領域３４とを、別のハードウェアとして構成される排他制御機構４２に備えるようにしてもよい。この場合、プロセッサ間共有メモリ用ライブラリ２３６及び２７６が提供するプロセッサ間共有オブジェクト操作機能７３（図８）に、排他制御機構４２に備えられたプロセッサ間排他制御機能を実行するルーチンが組み込まれる。従って、プロセッサ間共有メモリ３１には、プロセッサ間共有メモリ管理機能３６とプロセッサ間共有メモリ領域３２とを備えるのみでよい。

【００４４】

プロセッサ間排他制御を専用に行う排他制御機構４２を備えることによって、プロセッサ間共有メモリ３１への操作に係る排他制御を、ハードウェアのプロセッサ間のセマフォ又はロック機能を用いて高速に行うことができる。

【００４５】

次に、図８に示すプロセッサ間共有オブジェクト操作機能７３での処理について説明する。先ず、リード７３ａが行うプロセッサ間共有オブジェクト３３ａに係る情報の読み出し操作の処理について図10で説明する。図10中、プロセッサＡ２１が行う処理として説明するが、異種プロセッサであるプロセッサＢ２５においても同様に実行可能である。

【００４６】

図10は、プロセッサ間共有オブジェクトに係る情報の読み出し操作の処理フローを示す図である。図10（Ａ）において、プロセッサ間中間コード実行環境７０において、プロセッサＡ２１は、参照用プロセッサ間共有オブジェクトを作成する（ステップＳ１０ｒ）。そして、プロセッサＡ２１は、アプリケーションプログラム２３２からプロセッサ間共有オブジェクトを参照可能とする処理を行って（ステップＳ２０ｒ）、プロセッサ間共有オブジェクトに係る情報の読み出し操作を行う（ステップＳ３０ｒ）。

【００４７】

図10（Ｂ）では、図10（Ａ）のステップＳ２０ｒでのプロセッサ間共有オブジェクト参照の処理フローを説明する。図10（Ｂ）において、プロセッサＡ２１は、参照用プロセッサ間共有オブジェクトを呼び出して（ステップＳ２１ｒ）、オブジェクト対応機能７３ｆによって、アプリケーションプログラム２３２が指定した名前に対応付けられるプロセッサ間共有オブジェクト３３ａの実体をプロセッサ間共有メモリ３１のプロセッサ間共有オブジェクト名テーブル３５を参照して特定する（ステップＳ２２ｒ）。そして、対応付けられるプロセッサ間共有オブジェクト３３ａを戻り値として取得し（ステップＳ２３ｒ）、参照用プロセッサ間共有オブジェクトにプロセッサ間共有オブジェクト３３ａを返却する（ステップＳ２４ｒ）。参照用プロセッサ間共有オブジェクトがプロセッサ間共有オブジェクト３３ａとして振る舞うようになる。

【００４８】

図10（Ｃ）では、図10（Ａ）のステップＳ３０ｒでのプロセッサ間共有オブジェクトの読み出し操作の処理フローを説明する。図10（Ｃ）において、プロセッサＡ２１は、アプリケーションプログラム２３２からのリード７３ａの呼び出しによって引数を受け付けてその引数を解析し（ステップＳ３１ｒ及びＳ３２ｒ）、プロセッサ間共有メモリ３１又は排他制御機能４２に備えられるプロセッサ間排他制御機能３７を使用して、プロセッサＡ２１がプロセッサ間共有メモリ３１に格納されているプロセッサ間共有オブジェクト３３ａに係る情報を読み出すためのセマフォ又はロックを取得する（ステップＳ３３ｒ）。

【００４９】

そして、高速データ転送部７９ｂを使用して、プロセッサ間共有メモリ領域３２のプロセッサ間ヒープ領域３３に格納されている、プロセッサ間共有オブジェクト名テーブル３５を参照して名前と対応付けしたプロセッサ間共有オブジェクト３３ａから、プロセッサＡ２１の専用メモリ領域（プロセッサＡ用メモリ領域２３１に展開されたヒープ領域６４）の参照用プロセッサ間共有オブジェクトへと、プロセッサ間共有オブジェクト３３ａに係る情報をデータ転送によってデータコピーする（ステップＳ３４ｒ）。

【００５０】

データコピーが終了すると、プロセッサ間共有メモリ排他制御によるセマフォ又はロックを解放する（ステップＳ３５ｒ）。そして、戻り値を作成して（ステップＳ３６ｒ）、戻り値をアプリケーションプログラム２３２へ返却することによって（ステップＳ３７ｒ）、リード７３ａの処理を終了する。

【００５１】

図11は、読み出しプログラムの例を示す図である。図11に示す読み出しプログラム３１０は、アプリケーションプログラム２３２及び２７２においてプロセッサ間共有オブジェクト３３ａに係る情報の読み出し操作を行うためのプログラムの例である。読み出しプログラム３１０において、ｐｒｇ３１２「CPUSharedMemory_xxx obj ;」の実行によって、図10（Ａ）のステップＳ１０ｒによる参照用プロセッサ間共有オブジェクトが作成される。

【００５２】

また、ｐｒｇ３１４「obj = (CPUSharedMemory_xxx) CPUShaedMemoryNaming.lookup(“SharedObjectName_XXX”);」の実行によって、図10（Ｂ）のプロセッサ間共有オブジェクトへの参照処理が行われ、指定した名前（SharedObjectName_XXX）と参照用プロセッサ間共有オブジェクト（CPUSharedMemory_xxx）との対応がなされる。指定した名前によって、プロセッサ間共有オブジェクト名テーブル３５に登録されているプロセッサ間共有オブジェクト３３ａの実体が参照用プロセッサ間共有オブジェクトから参照可能となり、プロセッサ間共有オブジェクト３３ａとして振る舞うことができる。

【００５３】

更に、ｐｒｇ３１６「data = obj.read();」の実行によって、図10（Ｃ）のプロセッサ間共有オブジェクトの読み出し操作が行われ、プロセッサ間共有メモリ３１から読み出したプロセッサ間共有オブジェクト３３ａに係る情報が値dataに設定される。

【００５４】

次に、ライト７３ｂが行うプロセッサ間共有オブジェクト３３ａに係る情報の書き込み操作の処理について図12で説明する。図12中、プロセッサＡ２１が行う処理として説明するが、異種プロセッサであるプロセッサＢ２５においても同様に実行可能である。

【００５５】

図12は、プロセッサ間共有オブジェクトに係る情報の書き込み操作の処理フローを示す図である。図12（Ａ）において、プロセッサ間中間コード実行環境７０において、プロセッサＡ２１は、参照用プロセッサ間共有オブジェクトを作成する（ステップＳ１０ｗ）。そして、プロセッサＡ２１は、アプリケーションプログラム２３２からプロセッサ間共有オブジェクトを参照可能とする処理を行って（ステップＳ２０ｗ）、プロセッサ間共有オブジェクトに係る情報の読み出し操作を行う（ステップＳ３０ｗ）。

【００５６】

図12（Ｂ）では、図12（Ａ）のステップＳ２０ｗでのプロセッサ間共有オブジェクト参照の処理フローを説明する。図12（Ｂ）において、プロセッサＡ２１は、参照用プロセッサ間共有オブジェクトを呼び出して（ステップＳ２１ｗ）、オブジェクト対応機能７３ｆによって、アプリケーションプログラム２３２が指定した名前に対応付けられるプロセッサ間共有オブジェクト３３ａの実体をプロセッサ間共有メモリ３１のプロセッサ間共有オブジェクト名テーブル３５を参照して特定する（ステップＳ２２ｗ）。そして、対応付けられるプロセッサ間共有オブジェクト３３ａを戻り値として取得し（ステップＳ２３ｗ）、参照用プロセッサ間共有オブジェクトにプロセッサ間共有オブジェクト３３ａを返却する（ステップＳ２４ｗ）。参照用プロセッサ間共有オブジェクトがプロセッサ間共有オブジェクト３３ａとして振る舞うようになる。

【００５７】

図12（Ｃ）では、図12（Ａ）のステップＳ３０ｗでのプロセッサ間共有オブジェクトの読み出し操作の処理フローを説明する。図12（Ｃ）において、プロセッサＡ２１は、アプリケーションプログラム２３２からのライト７３ｂの呼び出しによって引数を受け付けてその引数を解析し（ステップＳ３１ｗ及びＳ３２ｗ）、プロセッサ間共有メモリ３１

又は排他制御機能４２に備えられるプロセッサ間排他制御機能３７を使用して、プロセッサＡ２１がプロセッサ間共有メモリ３１に格納されているプロセッサ間共有オブジェクト３３ａに係る情報を書き出すためのセマフォ又はロックを取得する（ステップＳ３３ｗ）。

【００５８】

そして、高速データ転送部７９ｂを使用して、プロセッサＡ２１の専用メモリ領域（プロセッサＡ用メモリ領域２３１に展開されたヒープ領域６４）の参照用プロセッサ間共有オブジェクトから、プロセッサ間共有メモリ領域３２のプロセッサ間ヒープ領域３３に格納されている、プロセッサ間共有オブジェクト名テーブル３５を参照して名前と対応付けしたプロセッサ間共有オブジェクト３３ａへと、プロセッサ間共有オブジェクト３３ａに係る情報をデータ転送によってデータコピーする（ステップＳ３４ｗ）。

【００５９】

データコピーが終了すると、プロセッサ間共有メモリ排他制御によるセマフォ又はロックを解放する（ステップＳ３５ｗ）。そして、戻り値を作成して（ステップＳ３６ｗ）、戻り値をアプリケーションプログラム２３２へ返却することによって（ステップＳ３７ｗ）、ライト７３ｂの処理を終了する。

【００６０】

図13は、書き込みプログラムの例を示す図である。図13に示す書き込みプログラム３２０は、アプリケーションプログラム２３２及び２７２においてプロセッサ間共有オブジェクト３３ａに係る情報の書き込み操作を行うためのプログラムの例である。書き込みプログラム３２０において、ｐｒｇ３２２「CPUSharedMemory_xxx　obj　;」の実行によって、図12（Ａ）のステップＳ１０ｗによる参照用プロセッサ間共有オブジェクトが作成される。

【００６１】

また、ｐｒｇ３２４「obj = (CPUSharedMemory_xxx) CPUShaedMemoryNaming.lookup(“SharedObjectName_XXX”);」の実行によって、図12（Ｂ）のプロセッサ間共有オブジェクト参照の処理が行われ、指定した名前（SharedObjectName_XXX）と参照用プロセッサ間共有オブジェクト（CPUSharedMemory_xxx）との対応がなされる。指定した名前によって、プロセッサ間共有オブジェクト名テーブル３５に登録されているプロセッサ間共有オブジェクト３３ａの実体が参照用プロセッサ間共有オブジェクトから参照可能となり、プロセッサ間共有オブジェクト３３ａとして振る舞うことができる。

【００６２】

更に、ｐｒｇ３２６「obj.write(data);」の実行によって、図12（Ｃ）のプロセッサ間共有オブジェクトの書き込み操作が行われ、プロセッサ間共有メモリ３１のプロセッサ間共有オブジェクト３３ａにプロセッサ間共有オブジェクト３３ａに係る情報が書き込まれる。

【００６３】

図14は、プロセッサ間共有オブジェクトを登録するための処理フローを示す図である。図14中、プロセッサＡ２１が行う処理として説明するが、異種プロセッサであるプロセッサＢ２５においても同様に実行可能である。

【００６４】

図14において、プロセッサ間中間コード実行環境７０において登録７３ｃが実行されると、プロセッサＡ２１は、プロセッサ間共有メモリ・クラス７１（以下、単にクラスと言う）が登録されているかを確認し（ステップＳ４１）、登録済みであるか否かを判断する（ステップＳ４２）。登録済みである場合、ステップＳ４９へと進む。一方、登録済みでない場合、プロセッサＡ専用メモリ１３上のメソッド領域６３（図８）にクラスを生成して（ステップＳ４３）、生成したクラスを登録する（ステップＳ４４）。

【００６５】

そして、プロセッサＡ２１は、登録したクラスを関連するクラスとリンクした後（ステップＳ４５）、準備処理として初期化する（ステップＳ４６）。更に、プロセッサＡ２１

は、アクセス制御をクラスに設定し（ステップＳ４７）、クラスを初期化する（ステップＳ４７）。

【００６６】

次に、プロセッサＡ２１は、メソッド領域６３に生成したクラスのインスタンスをプロセッサ間共有メモリ３１のプロセッサ間ヒープ領域３３（図６）にプロセッサ間共有オブジェクト３３ａとして生成する（ステップＳ４９）。プロセッサ間共有オブジェクト３３ａがプロセッサ間ヒープ領域３３（図６）内に生成される際には、プロセッサ間共有メモリ管理機能３６を用いたヒープメモリ操作・管理機能７２ａ（図８）が実行され、メモリ割り当て／開放部３６ａによってプロセッサ間共有メモリ３１のプロセッサ間ヒープ領域３３へのメモリ割り当てが行われる。

【００６７】

更に、プロセッサＡ２１は、オブジェクト対応機能７３ｆによって、プロセッサ間共有オブジェクト３３ａの名前と実体との対応をプロセッサ間共有メモリ３１のプロセッサ間共有オブジェクト名テーブル３５に登録する（ステップＳ５０）。

【００６８】

図15は、登録プログラムの例を示す図である。図15に示す登録プログラム３３０は、プロセッサ間共有メモリ・クラス７１をインスタンス化してプロセッサ間共有オブジェクト３３ａを生成し、登録するためのプログラムの例である。登録プログラム３３０において、ｐｒｇ３３２「CPUSharedMemory_xxx obj = new CPUSharedMemory_xxx ()」の実行によって、新たにプロセッサ間共有オブジェクト３３ａがプロセッサ間共有メモリ３１上に生成される。そして、ｐｒｇ３３４「CPUShaedMemoryNaming.rebind(“SharedObjectName_XXX”, obj);」の実行によって、プロセッサ間共有オブジェクト３３ａの名前と実体との対応がプロセッサ間共有オブジェクト名テーブル３５に登録される。

【００６９】

図16は、プロセッサ間共有オブジェクトを削除するための処理フローを示す図である。図16中、プロセッサＡ２１が行う処理として説明するが、異種プロセッサであるプロセッサＢ２５においても同様に実行可能である。

【００７０】

図16において、プロセッサ間中間コード実行環境７０において削除７３ｄが実行されると、プロセッサＡ２１は、オブジェクト対応機能７３ｆによって、プロセッサ間共有オブジェクト３３ａの名前と実体との対応をプロセッサ間共有メモリ３１のプロセッサ間共有オブジェクト名テーブル３５から削除する（ステップＳ６１）。

【００７１】

次に、プロセッサＡ２１は、プロセッサ間共有メモリ３１上にあるインスタンス（プロセッサ間共有オブジェクト３３ａ）を削除する（ステップＳ６２）。この場合、ヒープメモリ捜査・管理機能７２ａによって、プロセッサ間共有メモリ３１のメモリ割り当て／開放部３６ａが実行されてプロセッサ間共有オブジェクト３３ａの領域が開放される。

【００７２】

そして、プロセッサＡ２１は、プロセッサＡ専用メモリ２３のメソッド領域６３にロードされたプロセッサ間共有メモリ・クラス７１が削除可能な場合に削除する（ステップＳ６３）。

【００７３】

図１７は、削除プログラムの例を示す図である。図１７に示す削除プログラム３４０は、ｐｒｇ３１２「obj = (CPUSharedMemory_xxx) CPUShaedMemoryNaming.lookup(“SharedObjectName_XXX”);」の実行によって、オブジェクト対応機能７３ｆにより、プロセッサ間共有オブジェクト名テーブル３５が参照され、指定した名前（SharedObjectName_XXX）に対応するプロセッサ間共有オブジェクト３３ａ（CPUSharedMemory_xxx）の実体を取得する。

【００７４】

そして、ｐｒｇ３１４「CPUShaedMemoryNaming.finalize (“SharedObjectName_XXX,

obj”);」の実行によって、オブジェクト対応機能７３ｆにより、指定した名前（SharedObjectName_XXX）と対応するプロセッサ間共有オブジェクト３３ａ（CPUSharedMemory_xxx）の実体とがプロセッサ間共有オブジェクト名テーブル３５から削除される。更に、ｐｒｇ３１６「System.runFinalize();」の実行によって、クラスを削除する。

【００７５】

上述したように、本発明では、下位にあるＯＳやハードウェアの違いを吸収し、異種プロセッサ上であっても上位にあるアプリケーションプログラムのソースをコンパイルした中間コードを共通の実行環境で動作可能とするプログラム実行環境を備えたシステムにおいて、２以上の異種プロセッサによってアクセス可能とするプロセッサ間共有メモリを備えるようにする。

【００７６】

本発明に係るプログラム実行環境は、一のプロセッサのみで実行されるアプリケーションプログラムを実行可能する他、プロセッサ間共有メモリを介して中間コードによって実行されるオブジェクトをプロセッサ間で共有するために、各プロセッサ専用メモリに作成した共有オブジェクトを、その共有オブジェクトを指定する名前とプロセッサ間共有メモリ内に作成した共有オブジェクトの実体とを対応させることによって、各プロセッサは読み出し及び書き込みの操作をプロセッサ間共有メモリ内の共有オブジェクトの実体に対して行うことができる。従って、各プロセッサ上で動作するアプリケーションプログラムを高速に連携動作させることができる。

【００７７】

また、このようなプログラム実行環境では、プロセッサ間共有メモリに対して共有オブジェクトの実体を格納するためのメモリ領域の割り当て／開放する管理機能と、プロセッサ間のプロセッサ間共有メモリに対する排他制御のためのセマフォ又はロック機能と、プロセッサ間共有メモリ内の共有オブジェクトの実体をプロセッサ専用メモリ上に割り当てたメモリ領域へ読み出し、また、プロセッサ専用メモリ内の共有オブジェクトの実体をプロセッサ間共有メモリ上に割り当てたメモリ領域へ書き出す機能と、プロセッサ間共有メモリに対するヒープ機能とを備えることによって、異種プロセッサ間でプロセッサ間共有メモリへのアクセスをプロセッサ専用メモリへのアクセスと同様に制御することができる。

【００７８】

よって、本発明に係るシステムでは、ＬＡＮ－ＩＦを介することなく、またＬＡＮドライバを不要とし、高速に異種プロセッサ間でオブジェクトを共有することができる。

【００７９】

本発明は、複数の異種プロセッサ（ヘテロジニアスマルチプロセッサ）で成るシステムに限定されることなく、例えば、中間コードを実行可能なプログラム実行環境を備えた非対称型マルチプロセシング方式の複数の同種のマルチプロセッサで成るシステムにも適応可能である。

【００８０】

更に、複数の同種のマルチプロセッサで成るシステムが、均一メモリアクセス（Uniform Memory Access/Architecture: UMA）又は非均一メモリアクセス（Non-Uniform Memory Access/Architecture : NUMA）を実装、集中共有メモリ(Centralized Shared Memory: CSM)又は分散共有メモリ(Distributed Shared Memory: DSM)を搭載等による共有メモリ型マルチプロセッサシステムであっても、非リモートメモリアクセス(No Remote memory Access: NORA)を実装、分散非共有メモリ(Distributed Non-Shared Memory)を搭載等による非共有メモリ型マルチプロセッサシステムであっても、すべてのプロセッサからアプリケーションプログラムが格納されているメモリを参照できない場合に、プロセッサ間共有メモリを備え、本発明に係るプログラム実行環境を適応することによってアプリケーションプログラムを高速に連携動作させることができる。

【００８１】

均一メモリアクセスは、すべてのプロセッサが同じスピードでメインメモリにアクセス

できる方式であり、メモリアクセスコストの均一性を保つことにより、対称型マルチプロセシング（Symmetric Multi-Processing(SMP)）システムに実装される。非均一メモリアクセスは、すべてのプロセッサが全てのメモリにアクセスできる方式であるが、プロセッサからメインメモリへのアクセスのコストがアクセスするメモリ領域とプロセッサに依存して均一ではないメモリ・アーキテクチャである。また各ノードのメモリを全プロセッサに共有の物理アドレス空間にマップできなければならない等の制約がある。また、非リモートメモリアクセスは、各ノードのメモリを全プロセッサに共通の物理アドレスにマップしない。

【００８２】

上記実施例において、オブジェクト指向プログラミング言語のＪａｖａ（登録商標）言語を一例として説明した。Ｊａｖａ（登録商標）言語（プログラムソース）は、中間コード（バイトコード）にコンパイルされ、Ｊａｖａ（登録商標）仮想マシンとしてのＯＳやハードウェアに依存しないプログラム実行環境で実行される。Ｊａｖａ（登録商標）仮想マシンでは、Ｊａｖａ（登録商標）の中間コード（バイトコード）をＪＩＴコンパイラ等によってハードウェアに依存するネイティブコードに変換して実行される。

【００８３】

本発明は、このようなプログラム実行環境を備えるシステムであれば適用可能であり、例えば、オブジェクト指向プログラミング言語のＣ＃言語、共通中間言語（Common Intermediate Language : CIL）等によるアプリケーションプログラムを実行するシステムであってもよい。例えば、この場合、共通言語基盤（Common Language Infrastructure : CLI）がプログラム実行環境に相当する。

【００８４】

以上の説明に関し、更に以下の項を開示する。

（付記１）

複数のプロセッサを含み、前記複数のプロセッサ上で動作可能な中間コードを実行するプログラム実行環境を備えたプログラム実行システムであって、

前記複数のプロセッサ夫々に専用の複数のプロセッサ専用メモリと、

前記複数のプロセッサ間で共有される前記中間コードによって操作される共有オブジェクトを格納するプロセッサ間共有メモリとを備え、

前記複数のプロセッサの各々は、

前記共有オブジェクトを指定する名前と前記プロセッサ間共有メモリ内の該共有オブジェクトの実体とを対応させることによって、前記複数のプロセッサ専用メモリの各々と該プロセッサ間共有メモリ間にて該共有オブジェクトを読み出し及び書き込みする読出書込機能を実行するようにしたプログラム実行システム。

（付記２）

前記プロセッサ間共有メモリは、前記共有オブジェクトの名前と該共有オブジェクトの実体とを対応させた対応テーブルを備え、

前記複数のプロセッサの各々は、前記共有オブジェクトの名前と該共有オブジェクトの実体との対応を前記対応テーブルに書き込むことにより該共有オブジェクトを登録し、また該登録された共有オブジェクトを該対応テーブルから削除することにより該共有オブジェクトを削除する登録削除機能を実行するようにした付記１記載のプログラム実行システム。

（付記３）

前記複数のプロセッサの各々は、

前記プロセッサ間共有メモリのメモリ領域の割り当て及び開放を管理する共有メモリ管理機能と、

前記複数のプロセッサ間の前記プロセッサ間共有メモリへのアクセスに対してセマフォ又はロック機能を用いて排他制御を行うプロセッサ間排他制御機能と、

前記プロセッサ間共有メモリのヒープ領域に前記共有オブジェクトを格納するための共有メモリヒープ機能とを実行するようにした付記１又は２記載のプログラム実行システム

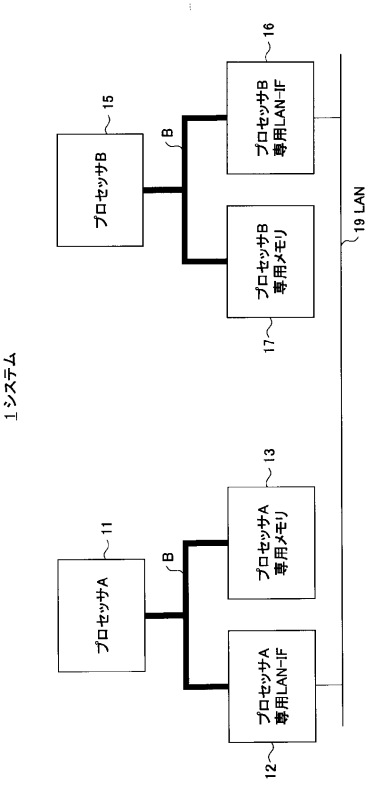
。
（付記４）
前記プロセッサ間共有メモリのメモリ領域と前記複数のプロセッサ専用メモリの各々のメモリ領域との間のダイレクトメモリアクセスによるデータ転送を制御するＤＭＡコントローラを更に備え、
前記複数のプロセッサの各々において、前記ＤＭＡコントローラを用いて前記高速データ転送を行うルーチンと該ルーチンを前記プログラム実行環境から呼び出すためのインターフェイスを備えるようにした付記１乃至３のいずれか一項記載のプログラム実行システム。
（付記５）
前記プロセッサ間排他制御機能を有するハードウェアを備え、
前記複数のプロセッサの各々において、前記ハードウェアのプロセッサ間排他制御機能を用いて排他制御を行うルーチンと該ルーチンを前記プログラム実行環境から呼び出すためのインターフェイスを備えるようにした付記１乃至３のいずれか一項記載のプログラム実行システム。
（付記６）
複数の異種プロセッサ毎に専用の専用メモリを備えたヘテロジニアスマルチプロセッサであって、
前記複数の異種プロセッサ間で共有される該各異種プロセッサ上で動作可能な中間コードによって操作される共有オブジェクトを格納するプロセッサ間共有メモリを備え、
前記各専用メモリは、
前記共有オブジェクトを指定する名前と前記プロセッサ間共有メモリ内の該共有オブジェクトの実体とを対応させることによって、前記各プロセッサ専用メモリと該プロセッサ間共有メモリ間にて該共有オブジェクトを読み出し及び書き込みする読出書込機能を前記異種プロセッサに実行させるプログラムを格納するようにしたヘテロジニアスマルチプロセッサ。
（付記７）
前記各専用メモリは、
前記共有オブジェクトの名前と該共有オブジェクトの実体との対応を前記プロセッサ間共有メモリが保持する対応テーブルに書き込むことにより該共有オブジェクトを登録し、また該登録された共有オブジェクトを該対応テーブルから削除することにより該共有オブジェクトを削除する登録削除機能を前記異種プロセッサに実行させるプログラムを格納するようにした付記６記載のヘテロジニアスマルチプロセッサ。
（付記８）
前記各専用メモリは、
前記プロセッサ間共有メモリのメモリ領域の割り当て及び開放を管理する共有メモリ管理機能と、
前記複数のプロセッサ間の前記プロセッサ間共有メモリへのアクセスに対してセマフォ又はロック機能を用いて排他制御を行うプロセッサ間排他制御機能と、
前記プロセッサ間共有メモリのヒープ領域に前記共有オブジェクトを格納するための共有メモリヒープ機能とを前記異種プロセッサに実行させるプログラムを格納するようにした付記６又は７記載のヘテロジニアスマルチプロセッサ。
【００８５】
本発明は、具体的に開示された実施例に限定されるものではなく、特許請求の範囲から逸脱することなく、種々の変形や変更が可能である。
【図面の簡単な説明】
【００８６】
【図１】疎結合のヘテロジニアスマルチプロセッサのシステム例を示す図である。
【図２】従来のヘテロジニアスマルチプロセッサによるシステムの構成例を示す図である。

【図３】従来の中間コード実行環境の構造例を示す図である。
【図４】本発明の第一実施例に係る密結合のヘテロジニアスマルチプロセッサによるシステム例を示す図である。
【図５】プロセッサ間共有メモリの領域構成例を示す図である。
【図６】ヘテロジニアスマルチプロセッサによるシステムの構成例を示す図である。
【図７】プロセッサ間共有メモリ用ライブラリの構成例を示す図である。
【図８】各プロセッサ用実行環境とプロセッサ間共有メモリ用ライブラリとによる機能構成を示す図である。
【図９】本発明の第二実施例に係る密結合のヘテロジニアスマルチプロセッサによるシステム構成例を示す図である。
【図１０】プロセッサ間共有オブジェクトに係る情報の読み出し操作の処理フローを示す図である。
【図１１】読み出しプログラムの例を示す図である。
【図１２】プロセッサ間共有オブジェクトに係る情報の書き込み操作の処理フローを示す図である。
【図１３】書き込みプログラムの例を示す図である。
【図１４】プロセッサ間共有オブジェクトを登録するための処理フローを示す図である。
【図１５】登録プログラムの例を示す図である。
【図１６】プロセッサ間共有オブジェクトを削除するための処理フローを示す図である。
【図１７】削除プログラムの例を示す図である。
【符号の説明】
【００８７】
２１　プロセッサＡ
２２　プロセッサＡ専用ＬＡＮ－ＩＦ
２３　プロセッサＡ専用メモリ
２５　プロセッサＢ
２６　プロセッサＢ専用ＬＡＮ－ＩＦ
２７　プロセッサＢ専用メモリ
３１　プロセッサ間共有メモリ
３２　プロセッサ間共有メモリ領域
３３　プロセッサ間ヒープ領域
３４　プロセッサ間排他制御用領域
３４ａ　プロセッサ間セマフォ
３４ｂ　プロセッサ間ロック
３５　プロセッサ間共有オブジェクト名テーブル
３６　プロセッサ間共有メモリ管理機能
３６ａ　メモリ割り当て／開放部
３７　プロセッサ間排他制御機能
５４　Ｊａｖａ（登録商標）ＶＭ
５４ａ　クラスローダー
５４ｂ　クラス・キャッシュ機能
５４ｃ　ヒープメモリ管理機能
５４ｄ　ガーベジ・コレクション（ＧＣ）機能
６１　実行時データ領域
６２　ＰＣレジスタ
６３　メソッド領域
６４　ヒープ領域
６５　Ｊａｖａ（登録商標）ＶＭスタック
６５２　フレーム
６５３　ローカル変数

６５４　オペランドスタック
６６　実行時定数プール
６７　Ｎａｔｉｖｅメソッドスタック
７０　プロセッサ間中間コード実行環境
７１　プロセッサ間共有メモリ・クラス
７２　プロセッサ間共有中間コードのヒープメモリ
７２ａ　ヒープメモリ操作・管理機能
７２ｂ　プロセッサ間共有ガーベジ・コレクション（ＧＣ）機能
７３　プロセッサ間共有オブジェクト操作機能
７３ａ　リード
７３ｂ　ライト
７３ｃ　登録
７３ｄ　削除
７３ｆ　オブジェクト対応機能
７４　プロセッサ間共有クラスローダー
７５　プロセッサ間共有クラス・キャッシュ機能
７９　プロセッサ間データ転送機能
７９ａ　引数／戻り値のデータ転送部
７９ｂ　高速データ転送部
１００　システム
２３１　プロセッサＡ用メモリ領域
２３２　アプリケーションプログラム
２３３　アプリケーションデータ
２３４　プロセッサＡ用実行環境
２３５　プロセッサＡ用ＪＩＴコンパイラ
２３６　プロセッサ間共有メモリ用ライブラリ
２７１　プロセッサＢ用メモリ用域
２７２　アプリケーションプログラム
２７３　アプリケーションデータ
２７４　プロセッサＢ用実行環境
２７５　プロセッサＢ用ＪＩＴコンパイラ
２７６　プロセッサ間共有メモリ用ライブラリ

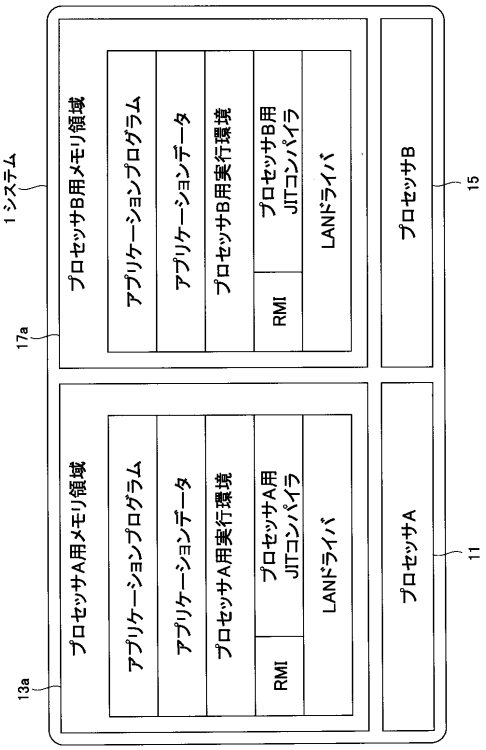
【図１】

疎結合のヘテロジニアスマルチプロセッサのシステム例を示す図



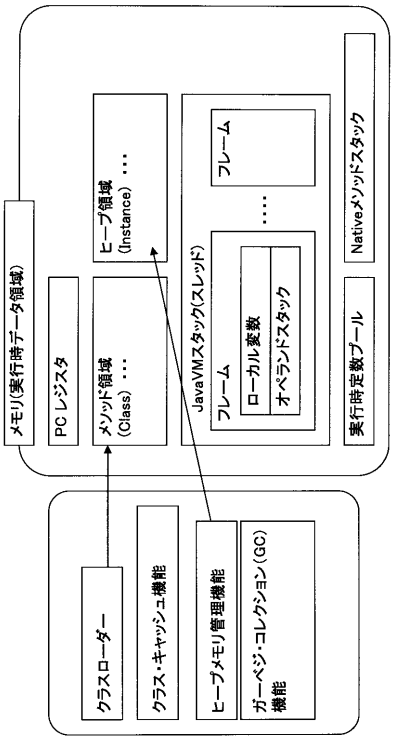
【図２】

従来のヘテロジニアスマルチプロセッサによるシステムの構成例を示す図



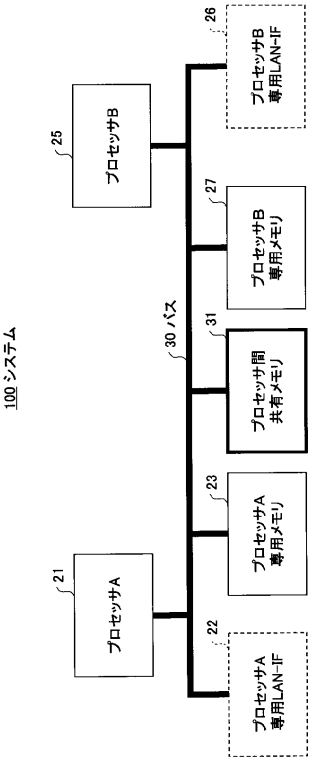
【図３】

従来の中間コード実行環境の構造例を示す図



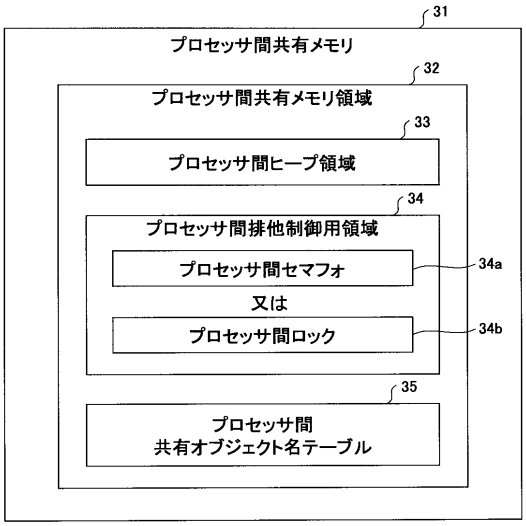
【図４】

本発明の第一実施例に係る
密結合のヘテロジニアスマルチプロセッサによるシステム例を示す図



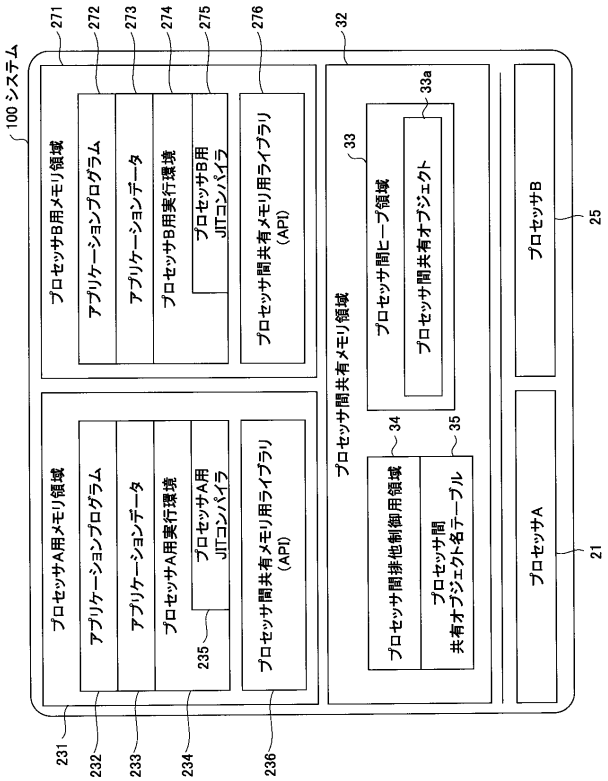
【図５】

プロセッサ間共有メモリの領域構成例を示す図



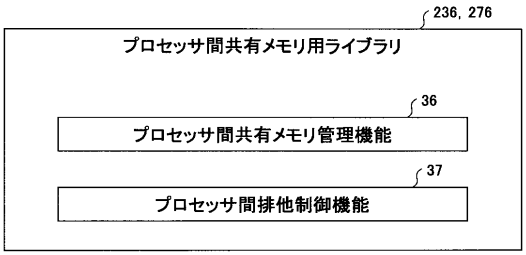
【図６】

ヘテロジニアスマルチプロセッサによるシステムの構成例を示す図



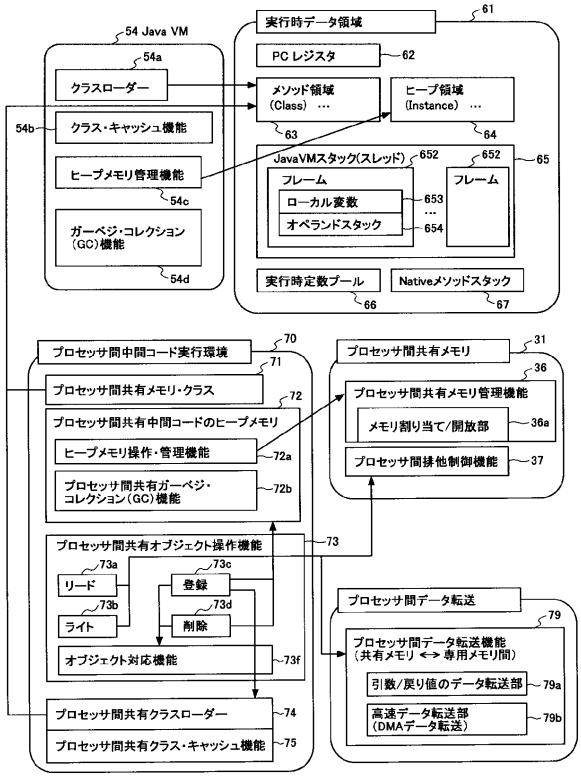
【図７】

プロセッサ間共有メモリ用ライブラリの構成例を示す図



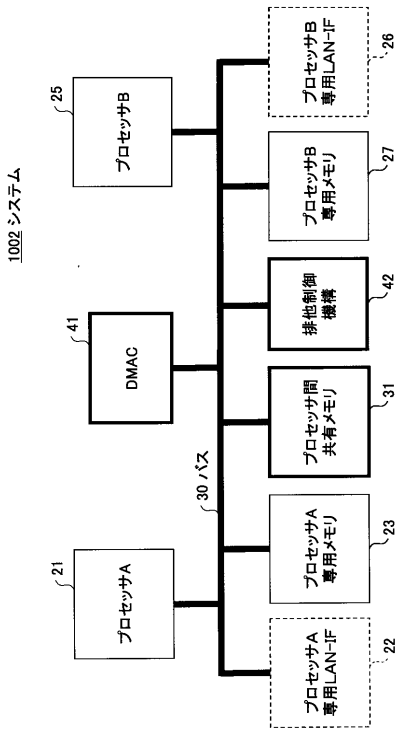
【図８】

各プロセッサ用実行環境と
プロセッサ間共有メモリ用ライブラリとによる機能構成を示す図



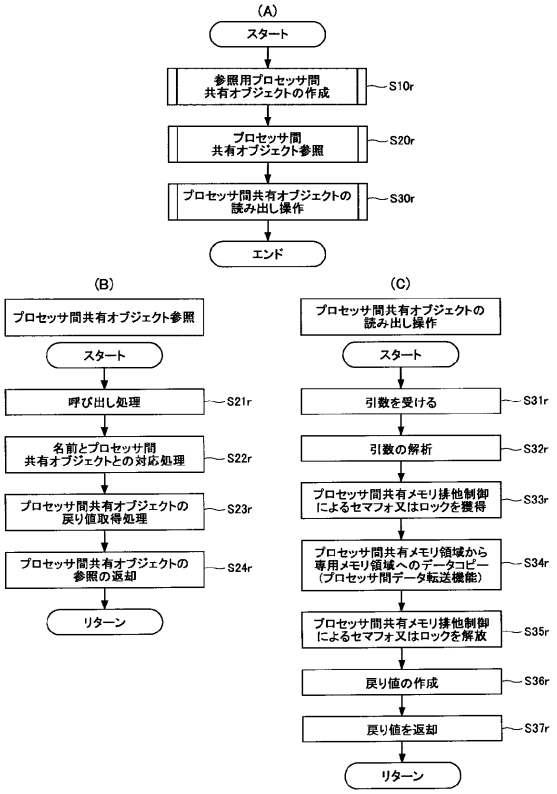
【図９】

本発明の第二実施例に係る
密結合のヘテロジニアスマルチプロセッサによるシステム構成例を示す図



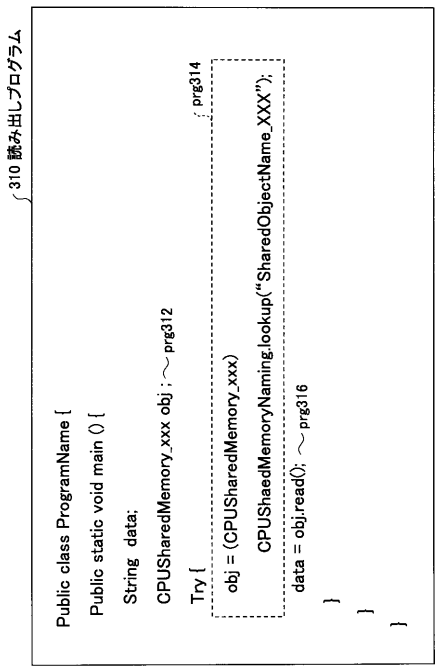
【図１０】

プロセッサ間共有オブジェクトに係る
情報の読み出し操作の処理フローを示す図



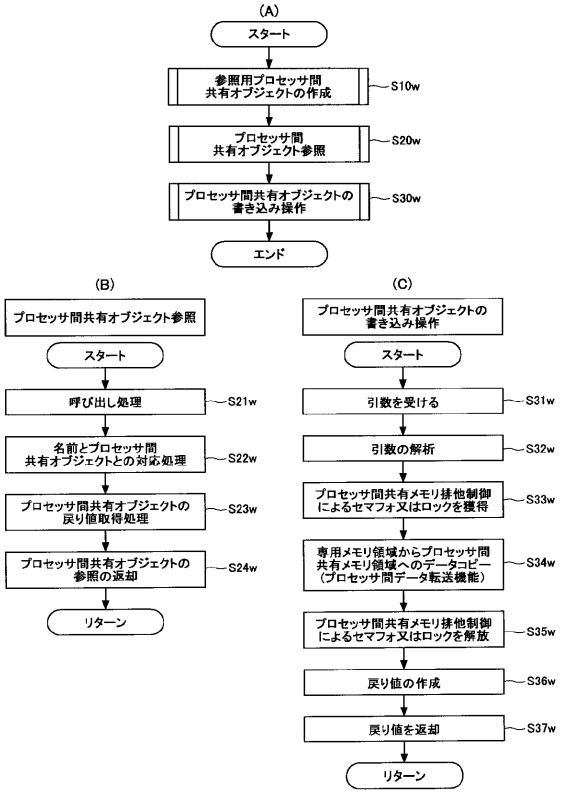
【図１１】

読み出しプログラムの例を示す図



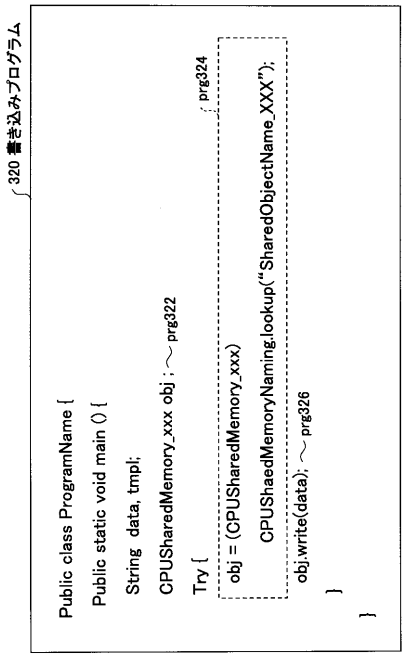
【図１２】

プロセッサ間共有オブジェクトに係る
情報の書き込み操作の処理フローを示す図



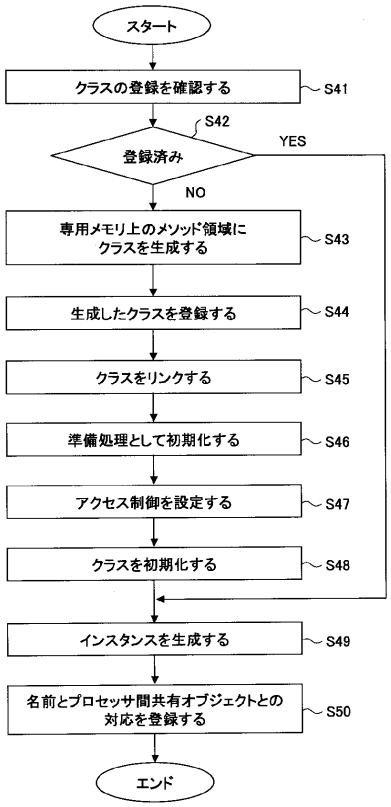
【図１３】

書き込みプログラムの例を示す図



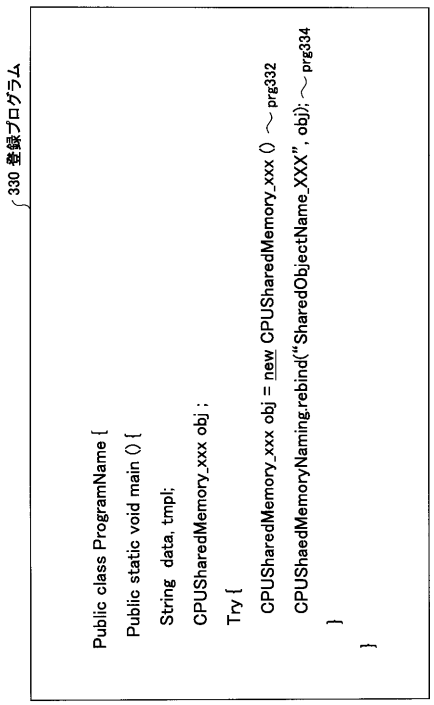
【図１４】

プロセッサ間共有オブジェクトを登録するための処理フローを示す図



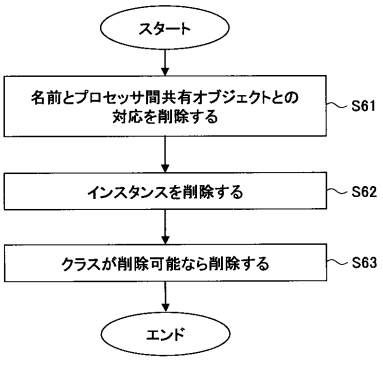
【図１５】

登録プログラムの例を示す図



【図１６】

プロセッサ間共有オブジェクトを削除するための処理フローを示す図



【図１７】

削除プログラムの例を示す図