(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2019-200698
(P2019-200698A)
(43)公開日　令和1年11月21日(2019.11.21)

(51)Int.Cl.			F I			テーマコード（参考）
G06F	12/06	(2006.01)	G06F	12/06	550B	5B160
H04L	12/771	(2013.01)	H04L	12/771		5K030

審査請求 未請求　請求項の数 8　ＯＬ　（全 25 頁）

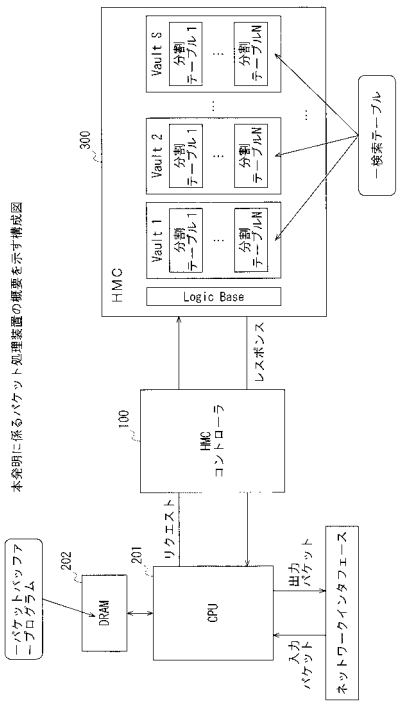
(21)出願番号　特願2018-96227(P2018-96227)
(22)出願日　平成30年5月18日(2018.5.18)

(71)出願人　000004226
日本電信電話株式会社
東京都千代田区大手町一丁目５番１号
(71)出願人　504132272
国立大学法人京都大学
京都府京都市左京区吉田本町３６番地１
(74)代理人　110001863
特許業務法人アテンダ国際特許事務所
(72)発明者　郡川　智洋
東京都千代田区大手町一丁目５番１号　日本電信電話株式会社内
(72)発明者　川端　明生
東京都千代田区大手町一丁目５番１号　日本電信電話株式会社内

最終頁に続く

(54)【発明の名称】パケット処理装置及びそのメモリアクセス制御方法

(57)【要約】
【課題】仮想化環境での使用を前提とした、汎用デバイスから構成される汎用装置で、検索テーブルへの高いメモリアクセス性能を有するパケット処理装置を提供する。
【解決手段】パケット処理においてＣＰＵ２０１からアクセスされるテーブルを記憶したHMC３００と、ＣＰＵ２０１からのHMC３００３００の前記テーブルへのメモリアクセスを制御するHMCコントローラ１００とを備えたパケット処理装置であって、HMCは並列アクセス可能なＳ個のブロック（Ｖａｕｌｔ）に区画されており、前記各ブロックは並列アクセス可能なＮ個のバンクに区画されており、前記テーブルは分割されてHMC３００の前記バンクに分散して記憶されており、HMCコントローラ３００は、前記アクセスリクエストに対してアクセス対象データが格納されているブロック及びバンクを特定して前記HMC３００にアクセスする。
【選択図】図１



【特許請求の範囲】
【請求項１】
パケット処理において演算装置からアクセスされるテーブルを記憶した記憶装置と、前記演算装置からの前記記憶装置の前記テーブルへのアクセスリクエストに基づき前記記憶装置へのメモリアクセスを制御する制御装置とを備えたパケット処理装置であって、
前記記憶装置の記憶領域は互いに並列アクセス可能なＳ個（Ｓは２以上の自然数）のブロックに区画されており、
前記各ブロックの記憶領域は互いに並列アクセス可能なＮ個（Ｎは２以上の自然数）のバンクに区画されており、
前記テーブルは分割されて前記記憶装置の前記バンクに分散して記憶されており、
前記制御装置は、前記アクセスリクエストに対してアクセス対象データが格納されているブロック及びバンクを特定して前記記憶装置にアクセスする
ことを特徴とするパケット処理装置。
【請求項２】
前記テーブルをＮ個の分割テーブルに等分割し、前記Ｓ個のブロックのそれぞれにおいて、前記Ｎ個の分割テーブルを前記Ｎ個のバンクに対応させて記憶した
ことを特徴とする請求項１記載のパケット処理装置。
【請求項３】
前記制御装置は、さらに、ブロック識別子及びバンク識別子の組により特定される記憶領域へのアクセス状態を管理するアクセス状態管理部と、同一データが分散記憶されたブロック識別子及びバンク識別子の組により特定される複数の記憶領域からアクセス状態がアクセス可能である記憶領域を選択し、選択した記憶領域へのアクセスを行うアクセス制御部とを備えた
ことを特徴とする請求項１又は２記載のパケット処理装置。
【請求項４】
前記制御装置は、さらに、前記演算装置からのアクセスリクエストの種別を識別するリクエスト識別部と、前記リクエスト識別部により識別された種別が前記テーブルの更新リクエストの場合に当該更新リクエストに基づき前記Ｓ個のブロックのそれぞれにおいてテーブルの更新処理を行うデータ更新制御部とを備えた
ことを特徴とする請求項１乃至３何れか１項記載のパケット処理装置。
【請求項５】
前記制御装置は、さらに、前記演算装置からのアクセスリクエストによる前記記憶装置の負荷を監視する負荷監視部と、前記負荷監視部により計測された負荷に基づき１つ以上のブロックにおいて第１のバンクに記憶されている分割テーブルを第２のバンクにコピーする分割変動制御部とを備えた
ことを特徴とする請求項１乃至４何れか１項記載のパケット処理装置。
【請求項６】
前記アクセスリクエストはパケットに付随するパケット付随情報を含み、
前記テーブルは前記パケット付随情報に基づき分割されている
ことを特徴とする請求項１乃至５何れか１項記載のパケット処理装置。
【請求項７】
前記記憶装置は、複数のデータ記憶素子層とメモリコントロール機能層とを互いに接続するように積層するとともに、各データ記憶素子層を平面上においてＳ個の区画に分割するとともに各データ記憶素子層の同一区画間を互いに接続することによりブロックを形成した
ことを特徴とする請求項１乃至６何れか１項記載のパケット処理装置。
【請求項８】
パケット処理において演算装置からアクセスされるテーブルを記憶した記憶装置と、前記演算装置からの前記記憶装置の前記テーブルへのアクセスリクエストに基づき前記記憶装置へのメモリアクセスを制御する制御装置とを備えたパケット処理装置におけるメモリ

アクセス制御方法であって、
前記記憶装置の記憶領域は互いに並列アクセス可能なＳ個（Ｓは２以上の自然数）のブロックに区画されており、
前記各ブロックの記憶領域は互いに並列アクセス可能なＮ個（Ｎは２以上の自然数）のバンクに区画されており、
前記テーブルは分割されて前記記憶装置の前記バンクに分散して記憶されており、
前記制御装置は、前記アクセスリクエストに対してアクセス対象データが格納されているブロック及びバンクを特定して前記記憶装置にアクセスする
ことを特徴とするパケット処理装置のメモリアクセス制御方法。

【発明の詳細な説明】

【技術分野】

【０００１】
本発明は、通信ネットワークにおける大規模トラヒックフローを対象とするパケット処理装置及びそのメモリアクセス制御方法に関する。

【背景技術】

【０００２】
近年のＩｎｔｅｒｎｅｔ ｏｆ Ｔｈｉｎｇｓ（ＩｏＴ）やエッジコンピューティング、第５世代モバイルネットワーク（５Ｇ）の登場により、ネットワークを流れるトラヒック量や遅延低減化の要求、ネットワークに接続されるデバイス数、さらには通信の多様性は急速に増加している。通信事業者やサービスプロバイダのネットワークは、その規模や信頼性由来の要件から、従来は用途に特化した専用デバイスや独自のアーキテクチャからなる装置により構成されてきた。

【０００３】
しかし、近年の急激なトラヒック需要変動に対する柔軟かつ迅速な装置増減設やネットワーク機能の容易な追加実装を可能にするために、通信事業者ネットワークやサービスプロバイダネットワークのような大規模ネットワークにおいても、ネットワーク仮想化（Network Function Virtualization；ＮＦＶ）やソフトウェア定義ネットワーク（Software Defined Networking；ＳＤＮ）などの仮想化技術の活用が期待されている。

【０００４】
このような仮想化技術活用の機運到来の背景には、従来に比べてより汎用的なデバイスの性能向上がある。ＣＰＵやＤｙｎａｍｉｃ Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ（ＤＲＡＭ）といった、汎用的で安価なデバイスからなる汎用コンピュータの性能が向上したことにより、従来は専用装置を用いないと実現困難であった数十ギガビット毎秒級のパケット処理が汎用コンピュータ上のソフトウェアにより実現可能になってきている。したがって、今後、大規模ネットワークにおいても、汎用コンピュータを活用したネットワーク構築により、急激な需要変動や新サービスのための機能追加実装を柔軟・迅速・安価に実現することが可能になると期待される。

【０００５】
しかし、このような大規模ネットワークにおいては、以降で議論するように、パケット処理のためのテーブル検索等の処理で、現在の汎用コンピュータアーキテクチャではメモリアクセス性能が支配的な性能ボトルネックとなり、これが大規模ネットワークにおける仮想化技術導入の性能観点での障壁になる。

【０００６】
従来は検索に特化した専用デバイスであるＴｅｒｎａｒｙ Ｃｏｎｔｅｎｔ－Ａｄｄｒｅｓｓａｂｌｅ Ｍｅｍｏｒｙ（ＴＣＡＭ。以降、本表記を使用）を活用することで、パケット処理テーブル検索処理の性能を担保できた。しかし、これは専用デバイスで高価・高消費電力・小容量という課題もあるため、仮想化技術を用いた柔軟かつ低コストな大規模ネットワーク実現に向けて汎用コンピュータに専用デバイスを組み込むというアプローチは望ましくない。

【０００７】

一方、このメモリアクセス性能高めるデバイスとしてＨｙｂｒｉｄ　Ｍｅｍｏｒｙ　Ｃｕｂｅ（ＨＭＣ。以降、本表記を使用）が２０１３年４月に仕様が開示され、既にスーパーコンピュータ等の領域で使用されている。ＨＭＣは、３次元形状を持つ半導体の層が４～８枚積層され、各層がシリコン貫通電極によって接続されている。その積層した縦の列を“Ｖａｕｌｔ”と呼び、各Ｖａｕｌｔは、独立したＤＲＡＭベースのメモリであり独立にアクセス可能で並列動作が可能である。また、Ｖａｕｌｔ内には、各層ごとに数個のＢａｎｋと呼ばれる領域がある。同一Ｖａｕｌｔ内でこれらＢａｎｋは、共有バスにより接続されているが、共有バス衝突が発生しない範囲内で並列に動作（Ｂａｎｋ間ｉｎｔｅｒｌｅａｖｉｎｇ。以降、本表記を使用）可能。このため、汎用メモリデバイスながらきわめて高い性能を実現できる可能性を有している。

【０００８】

前述のように、パケット処理におけるルーティングやフィルタリング等のためのテーブル検索処理は、特に高いメモリアクセス性能を要求するため、従来のネットワーク装置においては、ＴＣＡＭのような専用の高速なメモリを使用されている。しかし、上記した仮想化技術を用いた柔軟かつ低コストな大規模ネットワーク実現に向けてＴＣＡＭのような専用デバイスを汎用コンピュータに組み込むというアプローチから望ましくないとともに、検索処理に限らず今後より多くのネットワーク機能が仮想化されていくうえでは、汎用コンピュータにおけるメモリアクセスの高性能化が必要である。

【０００９】

一方、新しいメモリのアーキテクチャを持つＨＭＣについては、ＨＭＣ内に検索テーブルを配置して、ＨＭＣのもつ広帯域を活用した高速な読み出しに関する検討も存在するが（非特許文献１参照）、ＴＣＡＭのようなテーブル検索の専用メモリでないため、検索処理に要求されるメモリアクセス性能に達していないとともに、後述する本発明のようにＨＭＣのもつ並列構造を積極的に活用する方式はまだ検討されていない。

【００１０】

ＮＦＶを考慮した汎用コンピュータを適用した従来の技術のパケット処理装置構成には、図２０に示すような（１）のＤＤＲｘ　ＤＲＡＭ及び（２）のＨＭＣを使用したアーキテクチャがある。

【００１１】

図２０の（１）では、上記したようにＤｏｕｂｌｅ－Ｄａｔａ－Ｒａｔｅ　３（ＤＤＲ３）ＤＲＡＭや速度がこの２倍となるＤｏｕｂｌｅ－Ｄａｔａ－Ｒａｔｅ　４（ＤＤＲ４）ＤＲＡＭを採用している。最近は、更にＤＤＲ４の２倍程度高速なＤｏｕｂｌｅ－Ｄａｔａ－Ｒａｔｅ　５（ＤＤＲ５）等が次世代メモリとして登場してきている。このような、Ｄｏｕｂｌｅ－Ｄａｔａ－Ｒａｔｅ　ｘ　ＤＲＡＭ（ＤＤＲｘ　ＤＲＡＭ。以降、本表記を使用）は、パケット処理においてパケットバッファやアドレス検索テーブル等に使用される。ＣＰＵは、マルチコア化されたマルチスレッドでの処理技術が一般化しており、並列処理が可能となっている。また、マルチコアＣＰＵは、各ＣＰＵコア内や各ＣＰＵコアで共通に使用する低容量で高速動作可能なキャッシュメモリを内蔵しており、キャッシュメモリに納まる範囲内の処理であれば高い処理性能を発揮する。しかしながら、これらキャッシュメモリは、容量が小さく容量不足によりメインメモリであるＤＤＲｘ　ＤＲＡＭへのアクセスが頻発した場合、性能のボトルネックが生じる。これは、ＤＤＲｘ　ＤＲＡＭは、アクセス速度がキャッシュメモリと比較して遅いとともに、アクセスの並列度がないかもしくは並列度があっても低いため、複数のＣＰＵコア側が同時に多くのアクセス要求を出す場合、ＤＤＲｘ　ＤＲＡＭ側がアクセス中でビジー状態となり、ＣＰＵコア側で待ち合わせ状態となるためである。

【００１２】

図２０の（２）では、メモリとしてＨＭＣを使用し、これをパケットバッファや検索テーブルとして使用している例を示している。ＨＭＣアクセス速度は、ＤＤＲｘ　ＤＲＡＭより高速ではあるが、ＴＣＡＭのようなテーブル検索の専用メモリでないため、検索処理に要求されるメモリアクセス性能に達してないとともに、後述する本発明のようにＨＭＣ

のもつ並列構造を積極的に活用する方式はまだ検討されていないため、前述したように、パケット処理装置の仮想化及び将来的にさらなるパケット通信速度の高速化、トラヒックの爆発的な増加、低遅延化等によるメモリアクセス性能不足による性能劣化が主要な性能ボトルネックとなることが予想される。

【先行技術文献】

【非特許文献】

【００１３】

【非特許文献１】Packet Matching on FPGAs Using HMC Memory: Towards One Million Rules, Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays

【発明の概要】

【発明が解決しようとする課題】

【００１４】

大規模な通信事業者ネットワークを汎用コンピュータにより実現し、将来的な大容量トラヒックに対応するため、上述した従来アーキテクチャの延長によるパケット処理方式では、いずれは限界がくると想定される。これは、パケット処理の中でも特に、ルーティング、ＱｏＳ（Quality of Service）、パケットフィルタリングのようなテーブル検索処理を伴う処理においてメモリアクセス性能不足が顕在化するためである。図２０の従来アーキテクチャでのＤＤＲｘ　ＤＲＡＭやＨＭＣ、また専用メモリであるＴＣＡＭでは、具体的には、以下が問題となってくる。

【００１５】

（１）ＤＤＲｘ　ＤＲＡＭを使用したアーキテクチャでは、メモリのアクセス並列度がないもしくは低い。マルチコアＣＰＵの複数のＣＰＵコアからＤＤＲｘ　ＤＲＡＭへのアクセスが頻発した場合、アクセス待ち状態によりパケット処理性能のボトルネックになる。

【００１６】

（２）ＨＭＣは、高速アクセスが可能であるが、単純にＨＭＣを従来のＤＲＡＭの代わりに接続したとしてもＣＰＵとのリンク帯域が向上するが、メモリアクセスの並列度が向上するわけではないため、メモリ内のテーブルへのアクセス性能はテーブル検索処理等に求められる水準までは向上しない。このため、従来はＴＣＡＭのような専用デバイスを用いる必要があったが、下記のような仮想化における課題が生じてくる。

【００１７】

（３）仮想化適用による柔軟な運用や低コスト化のメリットを享受するためには汎用コンピュータ等汎用装置でネットワークが作れることが重要だが、専用デバイスであるＴＣＡＭを使わないといけなかった高速テーブル検索などの領域の汎用デバイス化が課題となってくる。また、ＴＣＡＭは、高価・高消費電力・小容量という課題もある。

【００１８】

これらの問題を解決するためには、従来のパケット処理装置アーキテクチャでなく、新しいパケット処理装置アーキテクチャが必要となる。特に、仮想化環境での使用を前提とした、汎用デバイスから構成される汎用装置で、検索テーブルへの高いメモリアクセス性能を実現するパケット処理の具体的な方式の考案が必要である。

【課題を解決するための手段】

【００１９】

上記目的を達成するために、本願発明は、パケット処理において演算装置からアクセスされるテーブルを記憶した記憶装置と、前記演算装置からの前記記憶装置の前記テーブルへのアクセスリクエストに基づき前記記憶装置へのメモリアクセスを制御する制御装置とを備えたパケット処理装置であって、前記記憶装置の記憶領域は互いに並列アクセス可能なＳ個（Ｓは２以上の自然数）のブロックに区画されており、前記各ブロックの記憶領域は互いに並列アクセス可能なＮ個（Ｎは２以上の自然数）のバンクに区画されており、前記テーブルは分割されて前記記憶装置の前記バンクに分散して記憶されており、前記制御

装置は、前記アクセスリクエストに対してアクセス対象データが格納されているブロック及びバンクを特定して前記記憶装置にアクセスすることを特徴とする。

【発明の効果】

【００２０】

本発明によれば、記憶装置に記憶されたテーブルに対する演算装置からのアクセスを制御装置が制御するので、記憶装置として並列アクセス可能な複数のブロック及びバンクに区画された汎用的なものを用いることができる。これにより、仮想化環境での使用を前提とした、汎用デバイスから構成される汎用装置で、検索テーブルへの高いメモリアクセス性能を有するパケット処理を実現できる。

【図面の簡単な説明】

【００２１】

【図１】本発明に係るパケット処理装置の概要を示す構成図
【図２】ＨＭＣ内のテーブル分散配置方式を説明する図
【図３】第１の実施の形態に係るＨＭＣコントローラの機能ブロック図
【図４】（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部の構成例
【図５】ＨＭＣコントローラ内の振り分け機構の処理フロー例
【図６】ＨＭＣコントローラ内の振り分け機構の処理フロー例
【図７】第２の実施の形態に係るＨＭＣコントローラの機能ブロック図
【図８】ＨＭＣコントローラ内の振り分け機構およびテーブル更新制御機構の処理フロー例
【図９】ＨＭＣコントローラ内の振り分け機構およびテーブル更新制御機構の処理フロー例
【図１０】ＨＭＣコントローラ内のテーブル更新制御機構の処理フロー例
【図１１】テーブル更新制御機構におけるリクエスト識別部の状態遷移図
【図１２】ＨＭＣ内の負荷追従型テーブル分散配置方式を説明する図
【図１３】第３の実施の形態に係るＨＭＣコントローラの機能ブロック図
【図１４】分割テーブル配置管理部がもつ配置管理表の一例
【図１５】ＨＭＣコントローラ内の振り分け機構および負荷追従型テーブル分割変動機構の処理フロー例
【図１６】ＨＭＣコントローラ内の振り分け機構および負荷追従型テーブル分割変動機構の処理フロー例
【図１７】分割変動実施時の処理フロー例
【図１８】分割変動実施時における分割テーブルコピー処理の処理フロー例
【図１９】分割変動実施時における分割変動リセット時の処理フロー例
【図２０】従来のパケット処理装置の構成図

【発明を実施するための形態】

【００２２】

まず、本発明の概要について図面を参照して説明する。図１は本発明に係るパケット処理装置の概要を示す構成図である。

【００２３】

本発明では、上記の課題を解決するため、図１に示すように、検索テーブルデータの保存に、並列アクセス可能なＨｙｂｒｉｄ　Ｍｅｍｏｒｙ　Ｃｕｂｅ（ＨＭＣ）３００を用いるとともに、ＣＰＵ２０１＋ＤＲＡＭ２０２とＨＭＣ３００との間に、ＨＭＣ３００への並列アクセスを可能とするためのＨＭＣコントローラ１００を配置するアーキテクチャを提案する。このＨＭＣコントローラ１００は、Ｆｉｅｌｄ　Ｐｒｏｇｒａｍｍａｂｌｅ　Ｇａｔｅ　Ａｒｒａｙ（ＦＰＧＡ。以降、本表記を使用）等の再プログラム可能な汎用デバイスで実装する。これにより通信事業者ネットワークのような大規模ネットワークにおけるパケット処理等、高いメモリアクセス性能が求められるアプリケーションにおいて仮想化環境下での使用を想定した汎用デバイスからなる汎用システムにより高性能を実現可能となる。

【００２４】
図１において、ＣＰＵ２０１は、複数のＣＰＵコアを有するマルチコアＣＰＵで構成され、内部にキャッシュメモリを内蔵し、これと主メモリ用のＤＲＡＭ２０２と接続している。
【００２５】
ＨＭＣ３００は、前述したように、Ｖａｕｌｔを複数有し（Ｖａｕｌｔ　１～Ｖａｕｌｔ　ＳのＳ個）、各Ｖａｕｌｔは、ＣＰＵ２０１側から並列アクセス可能な構造をもつ。より具体的には、ＨＭＣは、データ記憶素子層である複数のＤＲＡＭ層と、メモリコントロール機能を実装した層であるロジックベースとを、Ｔｈｒｏｕｇｈ－Ｓｉｌｉｃｏｎ Ｖｉａ（ＴＳＶ／シリコン貫通電極）と呼ばれる層間接続導体により互いに接続するように積層したものである。ＨＭＣは、各データ記憶素子層を平面上において複数の区画に分割するとともに各データ記憶素子層の同一区画間を互いに接続することによりＶａｕｌｔが形成されている。また、一つのＶａｕｌｔは複数（Ｎ個）のＢａｎｋにより構成され、Ｂａｎｋ間共有バスにおいて衝突が発生しない範囲で、各Ｂａｎｋは並列アクセスが可能（Ｂａｎｋ ｉｎｔｅｒｌｅａｖｉｎｇ。以下、本表記を使用）である。なお、Ｖａｕｌｔは、特許請求の範囲の「ブロック」に相当する。
【００２６】
パケット処理においては、パケット処理プログラム及びパケットバッファは、ＣＰＵ２０１に接続されたＤＲＡＭ２０２内に設け、パケット処理時間に特に影響する検索テーブルをＨＭＣ３００内に設ける方式を示している。なお、本発明においては「パケット」とは、例えばＩｎｔｅｒｎｅｔ Ｐｒｏｔｏｃｏｌ（ＩＰ）パケットなどＯｐｅｎ Ｓｙｓｔｅｍｓ Ｉｎｔｅｒｃｏｎｎｅｃｔｉｏｎ（ＯＳＩ）参照モデルのレイヤー３のパケットを意味するものとする。
【００２７】
本発明では、以下に述べるように、検索テーブルデータをＨＭＣ３００内で並列アクセス可能な単位である複数のＶａｕｌｔとＢａｎｋに分割・分散して配置し、さらに、それらの分散配置された検索テーブルへのＣＰＵ２０１からのメモリアクセスを振り分けるためのＨＭＣコントローラ１００をＣＰＵ２０１とＨＭＣ３００間に設けることにより高速パケット処理方式を実現する。
【００２８】
（１）ＨＭＣ３００内の複数あるＶａｕｌｔの１つに、検索テーブル全体を当該Ｖａｕｌｔ内のＮ個のＢａｎｋ毎にテーブル１～テーブルＮに等分割（分割テーブル。以降、本表記を使用）して配置する。この１つのＶａｕｌｔに配置したものと同一内容かつ同一分割のテーブルを残りの全てのＳ－１個のＶａｕｌｔにコピーする。したがって、各（Ｖａｕｌｔ，Ｂａｎｋ）の組み合わせ（（Ｖａｕｌｔ，Ｂａｎｋ）対。以降、本表記を使用）それぞれが分割テーブルを持ち、Ｖａｕｌｔ間では独立に、同一Ｖａｕｌｔ内のＢａｎｋ間ではＢａｎｋ ｉｎｔｅｒｌｅａｖｉｎｇとして並列アクセスにアクセスが可能である。また、後述の第２の実施の形態では、アプリケーションにおけるルーティングプロトコル等のやりとりにより、ＨＭＣ３００内のテーブルの内容に更新が発生した場合は、テーブル更新処理を実施する。また、Ｖａｕｌｔ内のＢａｎｋ間でのテーブル分割方法は、等分割を基本とするが、後述する第３の実施の形態では、各分割テーブルへのメモリアクセスの集中度（負荷。以下、本表記を使用）に応じて動的に変動させる。
【００２９】
（２）ＨＭＣコントローラ１００において、ＣＰＵ２０１からのアクセスリクエストに基づいて、上記（１）のＨＭＣ３００内に分散配置された分割テーブル番号の抽出とこれに該当するアクセス先Ｂａｎｋ番号を特定し、さらにこのアクセス先Ｂａｎｋ番号をもとにアイドル状態（メモリアクセス中でない状態）のＶａｕｌｔを見つけてアクセス先の（Ｖａｕｌｔ，Ｂａｎｋ）対を決定し、当該（Ｖａｕｌｔ，Ｂａｎｋ）対へアクセスを振り分ける。また、後述の第２の実施の形態では、ＨＭＣコントローラ１００において、上記（１）のテーブル更新処理を制御する。また、後述の第３の実施の形態では、ＨＭＣコン

トローラ１００において、上記（１）の負荷に応じたＢａｎｋ間テーブル分割方法の動的変動を制御する。

【００３０】

上記（１）及び（２）により、ＨＭＣ３００の各（Ｖａｕｌｔ，Ｂａｎｋ）対に分散配置された分割テーブルに並列アクセスが可能となり、また後述の第２の実施の形態では動作中のテーブル更新に対応し、さらに第３の後述の実施の形態ではバーストトラヒックのような特定の分割テーブルへの負荷増大に対しても動的な対処が可能となるため、仮想化環境下での使用に適した汎用デバイスを用いて高速なパケット処理が可能となる。

【００３１】

以下に本発明の第１～第３の実施の形態について詳述する。

【００３２】

（第１の実施の形態）

本発明の第１の実施の形態に係るパケット処理装置について図面を参照して説明する。図２は、図１のＨＭＣ内のテーブル分散配置方式を具体化したものである。

【００３３】

図２に示すように、ＨＭＣ３００が、１つのＶａｕｌｔ内のＮ個のＢａｎｋにおいて、ルーティングテーブルやフローテーブル等の検索テーブル全体をテーブル１からテーブルＮまで等分割して配置し、この一つのＶａｕｌｔに配置した検索テーブルを、さらに残りの全てのＶａｕｌｔにコピーして配置する。これにより、同一テーブル番号のアクセスが競合しても複数のＶａｕｌｔに同一内容の検索テーブルがあるため、Ｖａｕｌｔ間の並列動作が可能となる。また、１つのＶａｕｌｔ内では、Ｂａｎｋ間のｉｎｔｅｒｌｅａｖｉｎｇによる並列動作が可能である。これら並列動作機能を高めた方式の採用により、ＣＰＵからの検索テーブルアクセス頻度が増大する、より高いレートでのパケット処理が期待できる。

【００３４】

図３は、図１におけるＨＭＣコントローラ１００内の振り分け機構構成を示したものである。この振り分け機構は、図２のＨＭＣ内テーブル分散配置方式により分散配置された各分割テーブルへのＣＰＵ２０１からのメモリリクエストを振り分けることにより、検索テーブルへのメモリアクセス並列動作を実現する。

【００３５】

図３に示すように、本発明は、ＣＰＵ２０１及びＤＲＡＭ２０２等から構成するプロセッサ部２００と、ＨＭＣコントローラ１００と、ＨＭＣ３００の３つの主要部分から構成される。

【００３６】

プロセッサ部２００は、ＣＰＵ２０１とこれに接続されるプログラムやパケットバッファを有するＤＲＡＭ２０２を備える。ＣＰＵ２０１には数個～数十個のオーダの複数のＣＰＵコアとこれに内蔵される複数個のキャッシュがある。

【００３７】

ＨＭＣ３００は、並列動作できる３２個程度のＶａｕｌｔから構成され、それぞれのＶａｕｌｔには１６個程度のＢａｎｋがある。ＨＭＣ３００内のこれらのＶａｕｌｔおよびＢａｎｋに、図１及び図２を参照して上述した検索テーブルが配置されている。

【００３８】

このプロセッサ部２００とＨＭＣ３００間に振り分け機構であるＨＭＣコントローラ１００を設ける。ＨＭＣコントローラ１００は、ＦＰＧＡ等の再プログラム可能な汎用デバイスにより構成可能である。

【００３９】

ＨＭＣコントローラ１００には、ＣＰＵ２０１からＨＭＣ３００への検索テーブルアクセスによるメモリリクエストを受け付け、アクセス結果を返すＣＰＵインタフェース部１０１と、これと接続してメモリリクエストから検索テーブル処理するために必要な情報を抽出するパケット付随情報抽出部１０２と、この抽出した宛先アドレス等からハッシュ計

算によりＨＭＣ３００の検索テーブルの分割テーブル番号（１～Ｎ）を特定する分割テーブル特定部１０３と、分割テーブル番号からこれと対応するＨＭＣ３００のＢａｎｋ番号を特定するＢａｎｋ番号特定部１０４と、Ｂａｎｋ番号からＨＭＣ３００のアクセスするＶａｕｌｔを決定するＶａｕｌｔ決定部１０５と、このアクセスするＶａｕｌｔを決定する際に（Ｖａｕｌｔ，Ｂａｎｋ）対がアイドル状態（メモリアクセス中でない状態）なのかビジー状態（メモリアクセス中状態）なのかを表示している（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６と、決定したアクセス先（Ｖａｕｌｔ，Ｂａｎｋ）対アドレスをもとにＨＭＣ３００を実際にアクセスするインタフェース部となるＨＭＣアクセスコントローラ部１０７とを備えている。

【００４０】

図４は、図３の（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６におけるＨＭＣ３００内の（Ｖａｕｌｔ，Ｂａｎｋ）対が現在、アイドル状態なのかビジー状態なのかを表示するアクセス表示フラグ構成を示す。

【００４１】

図４に示すように、マトリックス構成（Ｂａｎｋ番号，Ｖａｕｌｔ番号）で行がＢａｎｋ番号を示し、Ｂａｎｋ　１からＢａｎｋ　Ｎまであり、列がＶａｕｌｔ番号を示し、Ｖａｕｌｔ　１からＶａｕｌｔ　Ｓまである。現状のＨＭＣ３００では、前述したように最大でも１６×３２程度の簡易なマトリックスであり、アイドル状態時が“０”でビジー状態が“１”のフラグ表示構成となっている。本フラグは、ＨＭＣアクセス開始時に“１”を立て、ＨＭＣアクセス完了時に“０”リセットする。図４では、例として、マトリックス（３，２）においてＢａｎｋ　３がアクセス該当部となった場合、Ｖａｕｌｔ　２が“０”でアイドル状態であり、アクセス可能な状態にあることを示す。

【００４２】

以下、図２～図４の構成をもとに、図５及び図６を参照して、検索テーブル処理の流れについて、検索テーブルのメモリアクセスリクエストの入力から検索結果の出力までについて、図４のＨＭＣコントローラとの処理部位の関連を含めて説明する。図５及び図６は本発明のＨＭＣコントローラ１００内の振り分け機構の処理フロー例である。

【００４３】

図５及び図６において、図３のプロセッサ部２００内のＣＰＵ２０１からＨＭＣ３００の検索テーブルへのアクセスに伴うメモリリクエストを受け付け、振り分け機構の処理を開始する（ステップＳ１０１）。

【００４４】

ＣＰＵインタフェース部１０１では、受け付けた検索テーブルへのメモリアクセスリクエストをパケット付随情報抽出部１０２に転送する（ステップＳ１０２）。

【００４５】

これを受信したパケット付随情報抽出部１０２では、メモリアクセスリクエスト内容に応じて検索テーブル処理に必要な情報を抽出する（ステップＳ１０３）。例えば、ルーティングテーブル検索では、メモリアクセスリクエスト内での宛先ＩＰアドレス情報を抽出する。

【００４６】

この抽出した宛先アドレス等の情報をもとに分割テーブル特定部１０３では、ハッシュ計算によりＨＭＣの検索テーブルの分割テーブル番号（１～Ｎ）を特定する（ステップＳ１０４）。

【００４７】

この分割テーブル特定部１０３で特定した分割テーブル番号からＢａｎｋ番号特定部１０４では、これと対応するＨＭＣ３００内のアクセスするＢａｎｋ番号を特定する（ステップＳ１０５）。

【００４８】

次に、Ｂａｎｋ番号を受信したＶａｕｌｔ決定部１０５では、Ｂａｎｋ番号からＨＭＣ３００のアクセスするＶａｕｌｔを決定するために該当するＢａｎｋ番号をアクセスして

いないアイドル状態のＶａｕｌｔを見つけるため、（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６にアイドル状態の参照要求を出す（ステップＳ１０６）。

【００４９】

この参照要求を受信した（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６では、図４に示す（Ｖａｕｌｔ，Ｂａｎｋ）対のメモリアクセスしてないアイドル状態かメモリアクセス中のビジー状態かを表示するアクセス表示フラグを該当アクセスＢａｎｋ部分について順次確認する（ステップＳ１０７，Ｓ１０８）。全部アクセス中でビジー状態である場合（全てフラグ“１”）、一定時間Ｗ（１～数クロック程度）待機し（ステップＳ１１０）、再びフラグを順次確認する。そして、アイドル状態を最初に見つけたＶａｕｌｔ番号を参照番号結果としてＶａｕｌｔ決定部１０５に返送する（ステップＳ１０９）。この返送直後に、このフラグを“１”としてビジー状態にする（ステップＳ１１１）。

【００５０】

アクセスするＶａｕｌｔ番号を参照結果として受け取ったＶａｕｌｔ決定部１０５では、アクセスするＢａｎｋ番号とＶａｕｌｔ番号の対番号をＨＭＣアクセスコントローラ部１０７にアクセス要求する（ステップＳ１１２）。

【００５１】

これを受信したＨＭＣアクセスコントローラ部１０７では、この（Ｖａｕｌｔ，Ｂａｎｋ）対番号よりＨＭＣ３００の該当アドレスを割り出して、ＨＭＣに対してアクセス要求を出す（ステップＳ１１３）。このアクセスにおいてＨＭＣ３００から検索アクセス応答の状態を監視し（ステップＳ１１４）、アクセス応答が正常である場合には、アクセス結果をＶａｕｌｔ決定部１０５に返却転送する（ステップＳ１１５）。

【００５２】

これを受信したＶａｕｌｔ決定部１０５では、（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６の該当アクセス表示フラグにアクセス完了リセット指示するとともに、アクセス検索結果をＣＰＵインタフェース部１０１側に返送する（ステップＳ１１６）。

【００５３】

Ｖａｕｌｔ決定部１０５からのアクセス完了リセット指示により（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６では、該当アクセス表示フラグを“０”リセットし、アイドル状態とする（ステップＳ１１７）。

【００５４】

ＨＭＣ３００からのアクセス応答が異常でエラー状態であった場合（ステップＳ１１４）には、アクセス結果をエラーとしてＶａｕｌｔ決定部１０５に返却する（ステップＳ１１８）。Ｖａｕｌｔ決定部１０５では、これをアクセスエラーとしてＣＰＵ２０１側に返送する（ステップＳ１１９）。ＣＰＵ２０１では、エラー内容に応じてアプリケーションレベルで適宜エラー処理を行う。

【００５５】

このようなパケット処理装置によれば以下の効果が生じる。

【００５６】

（１）高速パケットの検索テーブルの分散配置によりＨＭＣ３００のもつメモリ容量を効率的に利用できるとともに、並列処理により汎用的なデバイスのみを活用してサーバ上のパケット処理性能向上を飛躍的に大きく拡大できる。

【００５７】

（２）ＨＭＣコントローラおよび振り分け機構をＦＰＧＡ等の再プログラム可能な汎用デバイスで実現することによりＣＰＵ２０１やＨＭＣ３００などのデバイス自体の変更は不要である。

【００５８】

（３）ＨＭＣ３００を含め汎用デバイスから成る汎用コンピュータによるシステム構成であるため、幅広い既存パケット処理ソフトウェアをより高速に動作させることが可能である。

【００５９】

（４）ＴＣＡＭに比べて低消費電力なＤＲＡＭベースのＨＭＣ３００の採用により、システム全体の消費電力削減や実装面積削減によるコンパクト化が可能となる。

【００６０】

（第２の実施の形態）

本発明の第２の実施の形態に係るパケット処理装置について図面を参照して説明する。本実施の形態が第１の実施の形態と異なる点は、第１の実施の形態に係るＨＭＣコントローラ１００に対して、さらに、ＨＭＣ３００に記憶されたテーブルを更新するテーブル更新制御機能を設けた点にある。他の点については第１の実施の形態と同様なので、ここでは主として相違点について説明する。なお、第１の実施の形態と同様の構成については同一の符号を付した。

【００６１】

図７は、図１におけるＨＭＣコントローラ１００内の振り分け機構およびテーブル更新制御機構構成を示したものである。この振り分け機構については、第１の実施の形態と同様である。一方、テーブル更新制御機構は、振り分け機構と連携し、アプリケーション側からの要求に応じてＨＭＣ３００内のテーブル更新処理を制御することにより、動作中のテーブル更新を可能とするものである。

【００６２】

図７に示すように、ＨＭＣコントローラ１００は、ＣＰＵ２０１からＨＭＣ３００への検索テーブルアクセスによるメモリリクエストを受け付け、アクセス結果を返すＣＰＵインタフェース部１０１と、これと接続してメモリリクエストが、通常のＨＭＣ３００内のテーブルへのメモリアクセスリクエスト（リードアクセス）であるのかテーブル更新のためのリクエストであるのかを識別するリクエスト識別部１１１と、リクエストがＨＭＣ３００内のテーブルへのメモリアクセスリクエストの場合にメモリリクエストからテーブル検索処理するために必要な情報を抽出するパケット付随情報抽出部１０２と、この抽出した宛先アドレス等からハッシュ計算によりＨＭＣ３００の検索テーブルの分割テーブル番号（１～Ｎ）を特定する分割テーブル特定部１０３と、分割テーブル番号からこれと対応するＨＭＣ３００のＢａｎｋ番号を特定するＢａｎｋ番号特定部１０４と、Ｂａｎｋ番号からＨＭＣ３００のアクセスするＶａｕｌｔを決定するＶａｕｌｔ決定部１０５と、このアクセスするＶａｕｌｔを決定する際に（Ｖａｕｌｔ，Ｂａｎｋ）対がアイドル状態（メモリアクセス中でない状態）なのかビジー状態（メモリアクセス中状態）なのかを表示している（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６と、決定したアクセス先（Ｖａｕｌｔ，Ｂａｎｋ）対アドレスをもとにＨＭＣ３００を実際にアクセスするインタフェース部となるＨＭＣアクセスコントローラ部１０７と、リクエストがＨＭＣ３００内のテーブル更新である場合にそのテーブル更新処理を制御するテーブル更新制御部１１２とを備えている。

【００６３】

以下、図２及び図４並びに図７の構成をもとに、図８～図１０を参照して、検索テーブル処理の流れについて、検索テーブルのメモリアクセスリクエストの入力から検索結果の出力までおよびテーブル更新の動作について、図７のＨＭＣコントローラとの処理部位の関連を含めて説明する。図８及び図９は本発明のＨＭＣコントローラ１００内の振り分け機構およびテーブル更新制御機構の処理フロー例である。また、図１０は、本発明のテーブル更新制御機構の処理フロー例（詳細）である。

【００６４】

本実施の形態では、ＨＭＣコントローラ１００がＣＰＵ２０１から受け付けるメモリリクエストは、テーブルへのアクセスリクエスト（リードリクエスト）と、テーブル更新に係るメモリリクエストとに大別される。さらに、後者のテーブル更新に係るメモリリクエストは、更新開始リクエストと、更新内容を含むテーブル更新内容リクエスト、更新終了リクエストとに別れる。テーブル更新の際には、ＨＭＣコントローラ１００は、更新開始リクエスト、更新内容を含む１つ又は複数のテーブル更新内容リクエスト、更新終了リクエストを順に受信する。

【００６５】
図８及び図９において、図７のプロセッサ部２００内のＣＰＵ２０１からＨＭＣ３００の検索テーブルへのアクセスに伴うメモリリクエストを受け付け、振り分け機構の処理を開始する（ステップＳ２０１）。
【００６６】
ＣＰＵインタフェース部１０１では、受け付けた検索テーブルへのメモリアクセスリクエストをリクエスト識別部１１１に送付し（ステップＳ２０２）、リクエスト識別部１１１ではリクエストがＨＭＣ３００内のテーブルへのアクセス（リードアクセス）であるのかテーブル更新に関するものであるのかを識別する（ステップＳ２０３）。識別の結果、ＨＭＣ３００内のテーブルへのアクセス（リードアクセス）である場合は、リクエストをパケット付随情報抽出部１０２に転送（ステップＳ２０４）し、テーブル更新に関するものである場合は図１０の本発明のテーブル更新制御機構の処理フロー例（詳細）により説明するフロー例に従って処理を実施する。
【００６７】
以降の処理（図８のＳ２０５以降及び図９の処理）は、ＨＭＣ３００内のテーブルへのアクセス（リードアクセス）についての処理であり、当該処理については第１の実施の形態と同様なのでここでは説明は省略する。
【００６８】
テーブル更新処理のフロー例について、図１０および図１１の本発明のテーブル更新制御機構におけるリクエスト識別部の状態遷移図を用いて説明する。
【００６９】
リクエスト識別部１１１において、ＨＭＣ３００内のテーブルへのメモリアクセス（リードアクセス）以外と判断されたリクエストについて、さらに、テーブル更新開始を示すリクエストであるかどうか識別する（ステップＳ２５１）。テーブル更新開始のリクエストであると判断された場合は、リクエスト識別部１１１の状態を図１１のようにテーブル更新モードに遷移させ、テーブル更新をテーブル更新制御部１１２に指示（ステップＳ２５２）、また、テーブル更新開始のリクエストでないと判断された場合は、テーブル更新内容のリクエストであるかテーブル更新終了のリクエストであるのかを識別する（ステップＳ２５７）。
【００７０】
テーブル更新モードに遷移した場合、テーブル更新制御部１１２では、テーブル更新を実施するためにＨＭＣ３００内のテーブルへのメモリアクセス状況の参照を要求する（ステップＳ２５３）。（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６では、全Ｖａｕｌｔのメモリアクセス状況を順次確認（ステップＳ２５４）し、全（Ｖａｕｌｔ，Ｂａｎｋ）対がアイドル状態かどうか確認（ステップＳ２５５）し、すべてアイドル状態であれば当該リクエストの処理を完了、そうでなければ一定時間W_2だけ待機（ステップＳ２５６）し、再度アクセス状況を確認する。
【００７１】
前記確認（ステップＳ２５７）において、テーブル更新内容のリクエストであると判断された場合は、テーブル更新制御部１１２において、テーブル更新のためのＨＭＣアクセスを指示（ステップＳ２５８）し、ＨＭＣアクセスコントローラ部１０７において指定されたアドレスへのＨＭＣアクセスを実施してテーブルを更新する（ステップＳ２５９）。テーブル更新制御部１１２では、アクセス結果をＣＰＵ２０１へ通知し、当該リクエストの処理を完了する（ステップＳ２６０）。
【００７２】
また、前記確認（ステップＳ２５７）において、テーブル更新終了のリクエストであると判断された場合は、リクエスト識別部１１１の状態を図１１のようにＶａｕｌｔ間コピーモードに遷移させ、Ｖａｕｌｔ間コピーをテーブル更新制御部１１２に指示する（ステップＳ２６１）。テーブル更新制御部１１２では、Ｖａｕｌｔ間コピーを指示（ステップＳ２６２）し、ＨＭＣアクセスコントローラ部１０７において指定されたアドレスへのＨ

ＭＣアクセスを実施して、Ｖａｕｌｔ間のテーブルデータを同期する（ステップＳ２６３）。テーブル更新制御部１１２では、アクセス結果をＣＰＵ２０１へ通知し、リクエスト識別部１１１の状態を図１１のように通常モードへ遷移させる（ステップＳ２６４，Ｓ２６５）。これにより、テーブル更新処理が完了し、当該リクエストの処理を完了する。

【００７３】

本実施の形態に係るパケット処理装置では、ルーティングプロトコルなどのアプリケーションによって発生するテーブル更新処理を、ＨＭＣ３００内に分散配置された検索テーブルについてもシステム動作中に実施することが可能となる。他の効果については第１の実施の形態と同様である。

【００７４】

（第３の実施の形態）

本発明の第３の実施の形態に係るパケット処理装置について図面を参照して説明する。本実施の形態が第１の実施の形態と異なる点は、メモリアクセスの負荷に追従してＨＭＣ３００内におけるテーブルの分割形態を動的に変化させる点にある。他の点については第１の実施の形態と同様なので、ここでは主として相違点について説明する。なお、第１の実施の形態と同様の構成については同一の符号を付した。

【００７５】

図１２は、その初期状態を示した、ＨＭＣ３００内の負荷追従型テーブル分散配置方式を具体化したものである。ＨＭＣ３００内へのテーブルの分散配置は、負荷に応じて動的に変動させるが、初期状態では、１つのＶａｕｌｔ内のＮ個のＢａｎｋにおいてルーティングテーブルやフローテーブル等の検索テーブル全体をテーブル１からテーブルＮまで等分割して配置し、この一つのＶａｕｌｔに配置した検索テーブルを、さらに残りの全てのＶａｕｌｔにコピーして配置する。これにより、同一テーブル番号のアクセスが競合しても複数のＶａｕｌｔに同一内容の検索テーブルがあるため、Ｖａｕｌｔ間の並列動作が可能となる。また、１つのＶａｕｌｔ内では、Ｂａｎｋ間のｉｎｔｅｒｌｅａｖｉｎｇによる並列動作が可能である。これら並列動作機能を高めた方式の採用により、ＣＰＵからの検索テーブルアクセス頻度が増大する、より高いレートでのパケット処理が期待できる。

【００７６】

本実施の形態では、このようにして分散配置した分割テーブルのうち、負荷が最大のものについて、その負荷があらかじめプログラムされた閾値を超えていることが検出された場合、同一Ｖａｕｌｔ内で最大負荷の分割テーブルを最小負荷の分割テーブルが配置されているＢａｎｋへ上書きコピー配置（Ｖａｕｌｔ内分割変動。以降、本表記を使用）する。このＶａｕｌｔ内分割変動は、１つのＨＭＣ内のＶａｕｌｔ　１から実施し、全Ｖａｕｌｔ数Ｓに対して、あらかじめプログラムされた、負荷に応じてＶａｕｌｔ内分割変動を許容するＶａｕｌｔ数S_{var}分だけ実施するまで繰り返す。Ｖａｕｌｔ（$S_{var}+1$）からＶａｕｌｔ　Ｓまでの（$S-S_{var}$）個のＶａｕｌｔについては、Ｖａｕｌｔ内分割変動を行わない。また、負荷検出は、ＣＰＵからのメモリアクセスをＲ個受信するごとに実施する。ここでＲはあらかじめ定められた値であり、且つ、プログラマブルである。

【００７７】

図１３は、図２におけるＨＭＣコントローラ内の振り分け機構および負荷追従型テーブル分割変動機構構成を示したものである。この振り分け機構は、図１２のＨＭＣ内テーブル分散配置方式により分散配置された各分割テーブルへのＣＰＵ２０１からのメモリリクエストを振り分けることにより、検索テーブルへのメモリアクセス並列動作を実現する。また、負荷追従型テーブル分割変動機構は、振り分け機構と連携し、分割テーブルへのメモリアクセス負荷を監視し、負荷の集中を検知した場合は、ＨＭＣ３００内のテーブル分割を変動させることにより、特定の分割テーブルへの負荷集中に対処することを可能とするものである。

【００７８】

図１３に示すように、ＨＭＣコントローラ１００は、ＣＰＵ２０１からＨＭＣ３００へ

の検索テーブルアクセスによるメモリリクエストを受け付け、アクセス結果を返すＣＰＵインタフェース部１０１と、これと接続してメモリリクエストからテーブル検索処理するために必要な情報を抽出するパケット付随情報抽出部１０２と、この抽出した宛先アドレス等からハッシュ計算によりＨＭＣ３００の検索テーブルの分割テーブル番号（１～Ｎ）を特定する分割テーブル特定部１０３と、分割テーブル番号をもとに各分割テーブルの負荷を監視する負荷監視部１２１と、負荷情報からテーブル分割を変動するべきか判断しテーブル分割実施時にはその制御を行う分割変動制御部１２２と、この負荷情報からテーブル分割を変動するべきか判断を行う際に必要な閾値をあらかじめプログラムしておき必要に応じて参照する負荷閾値部１２３と、ＨＭＣ３００内における分割テーブルの配置状況を示す配置管理表を管理する分割テーブル配置管理部１２４と、分割テーブル番号及び配置管理表からアクセス対象候補となるＨＭＣ３００の一以上のＢａｎｋ番号及びＶａｕｌｔ番号を特定するＢａｎｋ番号特定部１０４と、アクセス候補である一以上のＢａｎｋ番号及びＶａｕｌｔ番号からＨＭＣ３００のアクセスするＶａｕｌｔを決定するＶａｕｌｔ決定部１０５と、このアクセスするＶａｕｌｔを決定する際に（Ｖａｕｌｔ，Ｂａｎｋ）対がアイドル状態（メモリアクセス中でない状態）なのかビジー状態（メモリアクセス中状態）なのかを表示している（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６と、決定したアクセス先（Ｖａｕｌｔ，Ｂａｎｋ）対アドレスをもとにＨＭＣ３００を実際にアクセスするインタフェース部となるＨＭＣアクセスコントローラ部１０７とから構成される。

【００７９】

図１４に分割テーブル配置管理部１２４の有する配置管理表の一例を示す。配置管理表は、図１４に示すように、（Ｖａｕｌｔ，Ｂａｎｋ）対で特定される記憶領域と当該記憶領域に記憶されている分割テーブルの番号及び分割変動の実施状況との対応関係を示す。本実施の形態では、分割変動の実施状況は「分割変動未実施」「分割変動実施済み」の２つの状況を含む。また、「分割変動実施済み」の場合は、付随状態として、分割変動時の負荷情報を含む。負荷情報は、「負荷最小」「負荷最大」を含む。図１４の例は、分割変動によりｂａｎｋ　ｉの分割テーブルをｂａｎｋ　ｋに上書きコピーした場合を示している。

【００８０】

以下、図４、図１２～図１４の構成をもとに、検索テーブル処理の流れについて図１５～図１９を用いて、検索テーブルへのメモリアクセスリクエストの入力から検索結果の出力およびテーブル分割変動の動作について図１３のＨＭＣコントローラとの処理部位の関連を含めて説明する。図１５及び図１６は本発明のＨＭＣコントローラ内の振り分け機構および負荷追従型テーブル分割変動機構の処理フロー例であり、図１７及び図１８は分割変動実施時の処理フロー例、図１９は分割変動リセット時の処理フロー例である。

【００８１】

図１５及び図１６において、図１３のプロセッサ部２００内のＣＰＵ２０１からＨＭＣ３００の検索テーブルへのアクセスに伴うメモリリクエストを受け付け、振り分け機構部の処理を開始する（ステップＳ３０１）。

【００８２】

ＣＰＵインタフェース部１０１では、受け付けた検索テーブルへのメモリアクセスリクエストをパケット付随情報抽出部１０２に転送する（ステップＳ３０２）。

【００８３】

これを受信したパケット付随情報抽出部１０２では、メモリアクセスリクエスト内容に応じて検索テーブル処理に必要な情報を抽出する（ステップＳ３０３）。例えば、ルーティングテーブル検索では、メモリアクセスリクエスト内での宛先ＩＰアドレス情報を抽出する。

【００８４】

この抽出した宛先アドレス等の情報をもとに分割テーブル特定部１０３では、ハッシュ計算によりＨＭＣの検索テーブルの分割テーブル番号（１～Ｎ）を特定する（ステップＳ

３０４）。
【００８５】
負荷監視部１２１及び分割変動制御部１２２では、図１７～図１９を用いて後述する分割変動処理を負荷閾値部１２３、分割テーブル配置管理部１２４、（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６、ＨＭＣアクセスコントローラ部１０７と連携して実施する（ステップＳ３０５）。
【００８６】
分割テーブル特定部１０３で特定した分割テーブル番号及び分割テーブル配置管理部１２４が有する配置管理表に基づき、Ｂａｎｋ番号特定部１０４では、アクセス候補となるＨＭＣ内のＢａｎｋ番号及びＶａｕｌｔ番号を特定する（ステップＳ３０６）。
【００８７】
次に、アクセス候補となるＢａｎｋ番号及びＶａｕｌｔ番号を受信したＶａｕｌｔ決定部１０５では、アクセス候補のうちＢａｎｋ番号及びＶａｕｌｔ番号をアクセスしていないアイドル状態の（Ｖａｕｌｔ，Ｂａｎｋ）対を見つけるため、（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６にアイドル状態の参照要求を出す（ステップＳ３０７）。
【００８８】
この参照要求を受信した（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６では、図４に示す（Ｖａｕｌｔ，Ｂａｎｋ）対のメモリアクセスしてないアイドル状態かメモリアクセス中のビジー状態かを表示するアクセス表示フラグを該当アクセスＢａｎｋ部分について順次確認する（ステップＳ３０８，Ｓ３０９）。全部アクセス中でビジー状態である場合（全てフラグ“１”）、一定時間W_1（１～数クロック程度）待機し（ステップＳ３１１）、再びフラグを順次確認する。アイドル状態を最初に見つけたＢａｎｋ番号及びＶａｕｌｔ番号を参照番号結果としてＶａｕｌｔ決定部１０５に返送する（ステップＳ３１０）。この返送直後に、このフラグを“１”としてビジー状態にする（ステップＳ３１２）。
【００８９】
アクセスするＢａｎｋ番号及びＶａｕｌｔ番号を参照結果として受け取ったＶａｕｌｔ決定部１０５では、アクセスするＢａｎｋ番号とＶａｕｌｔ番号の対番号をＨＭＣアクセスコントローラ部１０７にアクセス要求する（ステップＳ３１３）。
【００９０】
これを受信したＨＭＣアクセスコントローラ部１０７では、この（Ｖａｕｌｔ，Ｂａｎｋ）対番号よりＨＭＣ３００の該当アドレスを割り出して、ＨＭＣ３００に対してアクセス要求を出す（ステップＳ３１４）。このアクセスにおいてＨＭＣ３００から検索アクセス応答の状態を監視し（ステップＳ３１５）、アクセス応答が正常である場合には、アクセス結果をＶａｕｌｔ決定部１０５に返却転送する（ステップＳ３１６）。
【００９１】
これを受信したＶａｕｌｔ決定部１０５では、（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６の該当アクセス表示フラグにアクセス完了リセット指示するとともに、アクセス検索結果をＣＰＵインタフェース部１０１側に返送する（ステップＳ３１８）。
【００９２】
Ｖａｕｌｔ決定部１０５からのアクセス完了リセット指示により（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６では、該当アクセス表示フラグを“０”リセットし、アイドル状態とする（ステップＳ３１９）。
【００９３】
ＨＭＣ３００からのアクセス応答が異常でエラー状態であった場合（ステップＳ３１５）には、アクセス結果をエラーとしてＶａｕｌｔ決定部１０５に返却する。Ｖａｕｌｔ決定部１０５では、これをアクセスエラーとしてＣＰＵ２０１側に返送する（ステップＳ３２０）。ＣＰＵ２０１では、エラー内容に応じてアプリケーションレベルで適宜エラー処理を行う。
【００９４】

前記分割変動処理（ステップＳ３０５）の詳細について図１７～図１９を参照して説明する。なお、図１７において処理主体が特に明示していない処理は、分割変動制御部１２２が行うものとする。

【００９５】

図１７に示すように、負荷監視部１２１は、分割テーブルごとの到着リクエスト数（負荷）を常時カウントし、全分割テーブル合計の累計到着リクエスト数がＲに到達した場合、負荷最大値と最小値、対応する分割テーブル番号（負荷情報）を出力するとともに、到着リクエスト数のカウンタをリセットする（ステップＳ４０１～Ｓ４０３）。一方、累計到着リクエスト数がＲに到達しない場合は、処理を終了する。すなわち、分割変動処理は、Ｒ個のリクエスト到着ごとに実施する。

【００９６】

分割変動制御部１２２は、負荷最大値が所定の負荷閾値を超えている否かにより分割変動の実施要否を判定する（ステップＳ４０４）。

【００９７】

分割変動制御部１２２は、負荷最大値が所定の負荷閾値を超えていない場合、分割変動実施済みか確認するために分割テーブル配置管理部１２４へ参照要求を行う（ステップＳ４０５）。分割テーブル配置管理部１２４は、配置管理表を参照して分割変動実施済みか否かの結果を分割変動制御部１２２に返却する（ステップＳ４０６）。分割変動制御部１２２は、返却結果が分割変動実施済みであった場合、後述する分割変動リセット処理を実施して処理を終了する（ステップＳ４０７）。分割変動制御部１２２は、返却結果が分割変動実施済みでない場合、処理を終了する。

【００９８】

一方、分割変動制御部１２２は、負荷最大値が所定の負荷閾値を超えている場合、分割変動実施済みか確認するために分割テーブル配置管理部１２４へ参照要求を行う（ステップＳ４０８）。分割テーブル配置管理部１２４は、配置管理表を参照して分割変動実施済みか否かの結果を分割変動制御部１２２に返却する（ステップＳ４０９）。ここで、分割変動実施済みの場合には、分割テーブル配置管理部１２４は、前回分割変動実施した分割テーブルが今回の負荷最大となる分割テーブルか否かの判定結果を分割変動制御部１２２に返却する（ステップＳ４１０）。分割変動制御部１２２は、分割変動実施済みであり且つ前回分割変動実施した分割テーブルが今回の負荷最大となる分割テーブルである場合には、処理を終了する。

【００９９】

一方、分割変動制御部１２２は、分割変動実施済みであり且つ前回分割変動実施した分割テーブルが今回の負荷最大となる分割テーブルでない場合には、分割変動リセット処理を実施し（ステップＳ４１１）、分割テーブルのコピー処理を実施する（ステップＳ４１２）。また、分割変動制御部１２２は、分割変動実施済みでない場合、分割テーブルのコピー処理を実施する（ステップＳ４１２）。

【０１００】

分割テーブルのコピー処理（ステップＳ４１２）は、Ｖａｕｌｔ　１から順にＶａｕｌｔ　S_{var}まで、負荷最大となる分割テーブルの内容を負荷最小分割テーブルが配置されているｂａｎｋへ上書きコピーする処理である。

【０１０１】

具体的には、図１８に示すように、まず、分割変動制御部１２２において、コピー処理が初回かどうかの確認（ステップＳ５０１）により初回である場合はコピー処理を行うＶａｕｌｔ番号ｍをｍ＝１としてＶａｕｌｔ　１から処理を開始（ステップＳ５０２）、初回ではない場合は$m < S_{var}$であるか確認（ステップＳ５０３）し、$m < S_{var}$である場合は、ｍを１増やして別のＶａｕｌｔに対してコピー処理を開始（ステップＳ５０４）、$m < S_{var}$でない場合は、コピー処理が完了したと判断し、本フローを終了する。

【０１０２】

Ｖａｕｌｔ　ｍにおいてコピー処理を実施するために、Ｖａｕｌｔ　ｍのアクセス状況

参照指示を出す（ステップＳ５０５）。（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６において、Ｖａｕｌｔ　ｍのアクセス状況を確認する（ステップＳ５０６）。次に、Ｖａｕｌｔ　ｍの全Ｂａｎｋがアイドル状態かどうか確認し（ステップＳ５０７）、アイドル状態でない場合は一定時間W_2待機（ステップＳ５０８）したのち、再度アクセス状態から確認する（ステップＳ５０７）。

【０１０３】

Ｖａｕｌｔ　ｍがアイドル状態である場合は、分割変動制御部１２２において、Ｖａｕｌｔ　ｍのアクセス抑止を指示する（ステップＳ５０９）。（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６において、Ｖａｕｌｔ　ｍの全Ｂａｎｋのアクセス状況フラグを“１”にし、アクセスを抑止する（ステップＳ５１０）。分割変動制御部１２２から、Ｖａｕｌｔ　ｍの内の最大負荷分割テーブルの内容を最小負荷分割テーブルが配置されているｂａｎｋへコピーして配置する指示を出す（ステップＳ５１１）。

【０１０４】

ＨＭＣアクセスコントローラ部１０７では、指示されたアドレスへのＨＭＣアクセスを要求する（ステップＳ５１２）。アクセス結果を確認し（ステップＳ５１３）、正常な場合は分割変動制御部１２２においてＶａｕｌｔ　ｍのアクセス抑止を解除し（ステップＳ５１４）、本処理フローの最初にもどる。確認（ステップＳ５１３）の結果が異常の場合は、本処理フローを終了する。

【０１０５】

前述の分割変動リセット処理（ステップＳ４０７，Ｓ４１１）について図１９を参照して説明する。分割変動制御部１２２において、分割変動リセット処理が初回かどうかの確認（ステップＳ６０１）により初回である場合は分割変動リセットを行うＶａｕｌｔ番号ｍをｍ＝１としてＶａｕｌｔ　１から処理を開始（ステップＳ６０２）、初回ではない場合は$m<S_{var}$であるか確認（ステップＳ６０３）し、$m<S_{var}$である場合は、ｍを１増やして別のＶａｕｌｔに対して分割変動リセット処理を開始（ステップＳ６０４）、$m<S_{var}$でない場合は、分割変動処理対象である全Ｖａｕｌｔについて分割変動リセット処理実施済みであると判断し、後述する分割変動リセット処理時の元データとするＶａｕｌｔ　ｐのアクセス抑止を解除（ステップＳ６１６）して本処理フローを終了する。

【０１０６】

Ｖａｕｌｔ　ｍにおいて分割変動リセット処理を実施するために、Ｖａｕｌｔ　ｍおよび分割変動処理対象外であるＶａｕｌｔ（$S_{var}+1$）～Ｓのアクセス状況参照指示を出す（ステップＳ６０５）。（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６において、Ｖａｕｌｔ　ｍのアクセス状況の確認およびＶａｕｌｔ（$S_{var}+1$）～Ｓのアクセス状況を順次確認し、ビジー状態のＢａｎｋ数が最小であるＶａｕｌｔ　ｐを選出する（ステップＳ６０６）。次に、Ｖａｕｌｔ　ｍおよびｐの全Ｂａｎｋがアイドル状態かどうか確認し（ステップＳ６０７）、アイドル状態でない場合は一定時間W_3待機（ステップＳ６０８）したのち、再度アクセス状況参照指示を出す（ステップＳ６０５）。

【０１０７】

Ｖａｕｌｔ　ｍおよびｐの全Ｂａｎｋがアイドル状態である場合は、分割変動制御部１２２において、Ｖａｕｌｔ　ｍおよびｐのアクセス抑止を指示する（ステップＳ６０９）。（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６において、Ｖａｕｌｔ　ｍおよびｐの全Ｂａｎｋのアクセス状況フラグを“１”にし、アクセスを抑止する（ステップＳ６１０）。分割変動制御部１２２から、Ｖａｕｌｔ　ｐの全Ｂａｎｋの内容をＶａｕｌｔ　ｍにコピーする指示を出す（ステップＳ６１１）。

【０１０８】

ＨＭＣアクセスコントローラ部１０７では、指示されたアドレスへのＨＭＣアクセスを要求する（ステップＳ６１２）。アクセス結果を確認し（ステップＳ６１３）、正常な場合は分割変動制御部１２２においてＶａｕｌｔ　ｍのアクセス抑止を解除し（ステップＳ６１４）、分割変動実施済みで未リセットのＶａｕｌｔがあるか確認する（ステップＳ６

１５）。確認（ステップＳ６１５）の結果、未リセットのＶａｕｌｔがある場合は、本処理フローの最初にもどり、無い場合は（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部１０６においてＶａｕｌｔ　ｐのアクセス状況フラグをすべて“０”にセットして、アクセス抑止を解除し（ステップＳ６１６）、本処理フローを終了する。確認（ステップＳ６１３）の結果が異常の場合は、本処理フローを終了する。

【０１０９】

本実施の形態に係るパケット処理装置では、バーストトラヒック等により特定の分割テーブルにメモリアクセスが集中する場合において局所的な対処を行っているので、システム全体のスループットを最大化することができる。他の効果については第１の実施の形態と同様である。

【０１１０】

なお、本実施の形態は第１の実施の形態の変形例として説明したが、第２の実施の形態の変形例とすることもできる点に留意されたい。

【０１１１】

以上、本発明の実施の形態について詳述したが本発明はこれに限定されるものではない。ＨＭＣコントローラ１００における各機能の実装形態やアルゴリズムは不問であり、他の実装形態やアルゴリズムであっても本発明を適用できる。例えば、上記各実施の形態では、アクセス対象とするアイドル状態のＢａｎｋを特定する際に、常にＶａｕｌｔ１から検索していたが、メモリリクエスト毎に検索開始Ｖａｕｌｔ番号を所定のルールで又はランダムで変更するようにしてもよい。

【０１１２】

また、上記第３の実施の形態では、分割変動処理時にはＳｖａｒ個のＶａｕｌｔについて負荷最大の分割テーブルを負荷最小の分割テーブルに上書きコピーしていた。ここでＳｖａｒは、あらかじめ定められたプログラマブルな値である。他の変形例としては、Ｓｖａｒを、負荷情報に応じて可変とするようにしてもよい。より具体的には、負荷が大きいほどＳｖａｒを大きく設定し、負荷が小さいほどＳｖａｒを小さくするよう動的に制御してもよい。

【０１１３】

また、上記各実施の形態では、記憶装置の一例としてＨＭＣについて説明したが、並列アクセス可能なブロック（Ｖａｕｌｔ）及びバンク構成を有する他の構造・規格の記憶装置であっても本発明を適用できる。

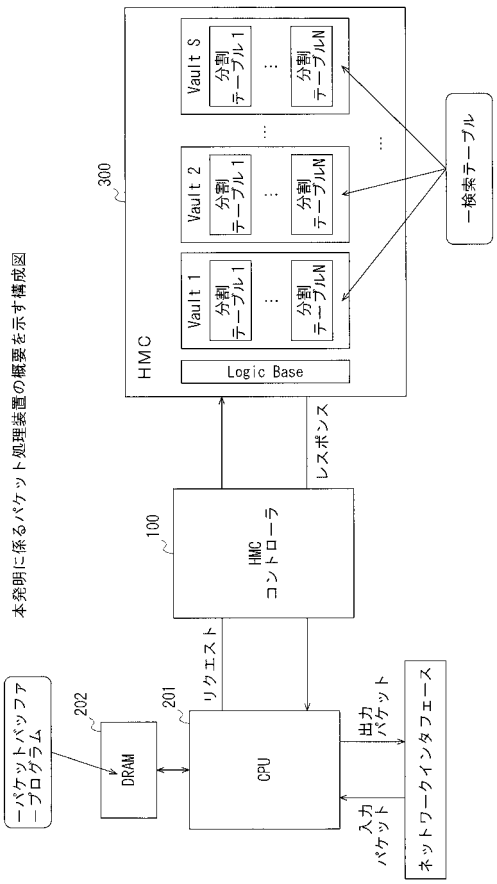
【符号の説明】

【０１１４】

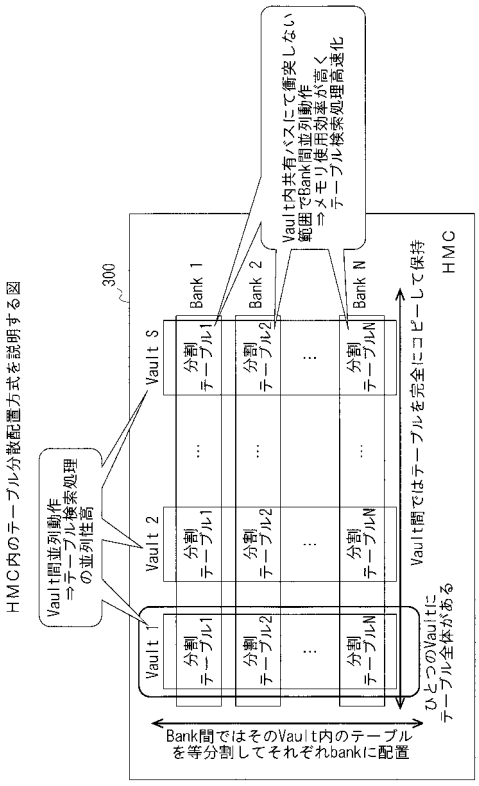
１００…ＨＭＣコントローラ
１０１…ＣＰＵインタフェース部
１０２…パケット付随情報抽出部
１０３…分割テーブル特定部
１０４…Ｂａｎｋ番号特定部
１０５…Ｖａｕｌ決定部
１０６…（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部
１０７…ＨＭＣアクセスコントローラ部
１１１…リクエスト識別部
１１２…テーブル更新制御部
１２１…負荷監視部
１２２…分割変動制御部
１２３…負荷閾値部
１２４…分割テーブル配置管理部
２００…プロセッサ部
２０１…ＣＰＵ
２０２…ＤＲＡＭ

３００…ＨＭＣ

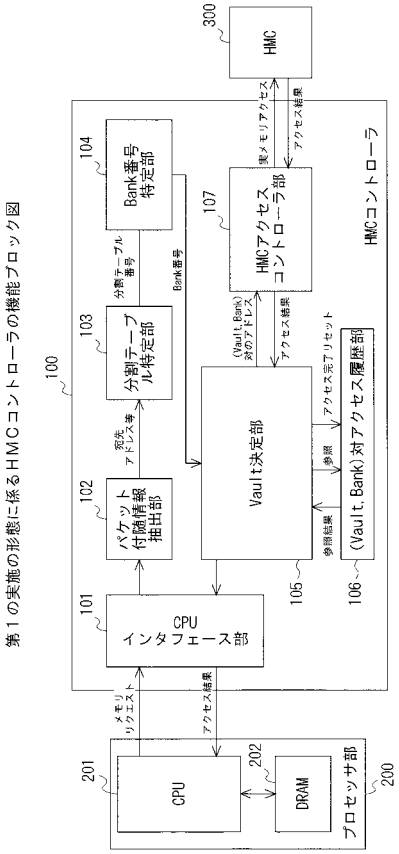
【図１】


本発明に係るパケット処理装置の概要を示す構成図
ーパケットバッファ
ープログラム
202
DRAM
201
CPU
リクエスト
100
HMC
コントローラ
レスポンス
300
HMC
Logic Base
Vault 1
Vault 2
Vault S
分割
テーブル1
分割
テーブルN
ー検索テーブル
出力
パケット
入力
パケット
ネットワークインタフェース

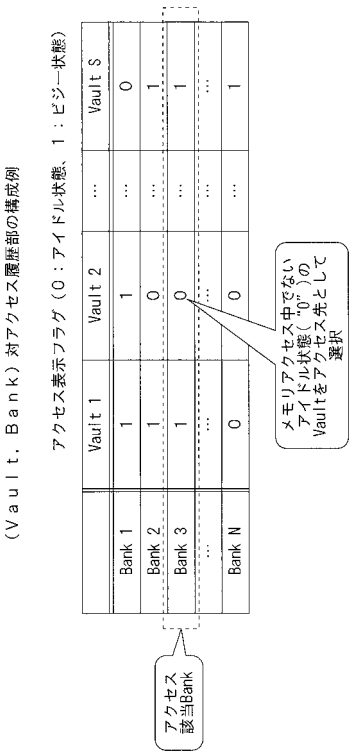
【図２】


HMC内のテーブル分散配置方式を説明する図
Vault間並列動作
⇒テーブル検索処理
の並列性高
300
Vault S
Vault 2
Vault 1
Bank 1
Bank 2
Bank N
分割
テーブル1
分割
テーブル2
分割
テーブルN
Vault内共有バスにて衝突しない
範囲でBank間並列動作が高く
⇒メモリ使用効率が高く
テーブル検索処理高速化
Vault間ではテーブルを完全にコピーして保持
HMC
ひとつのVaultに
テーブル全体がある
Bank間ではそのVault内のテーブル
を等分割してそれぞれbankに配置

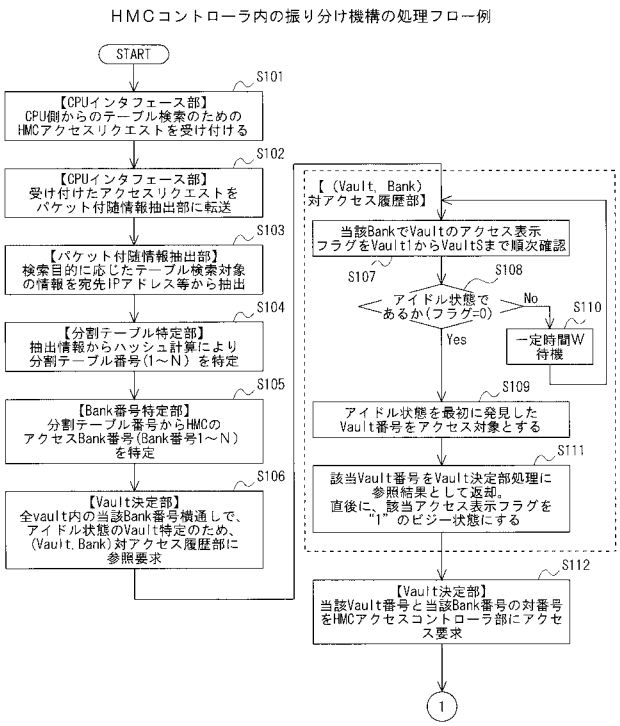
【図３】


第１の実施の形態に係るＨＭＣコントローラの機能ブロック図
HMC
実メモリアクセス
アクセス結果
HMCコントローラ
Bank番号特定部
分割テーブル特定部
パケット付随情報抽出部
HMCアクセスコントローラ部
Vault決定部
(Vault,Bank)対アクセス履歴部
アクセス完了リセット
参照
参照結果
(Vault,Bank)対のアドレス
分割テーブル番号
Bank番号
宛先アドレス等
CPU
インタフェース部
メモリリクエスト
アクセス結果
CPU
DRAM
プロセッサ部
100
101
102
103
104
105
106
107
200
201
202

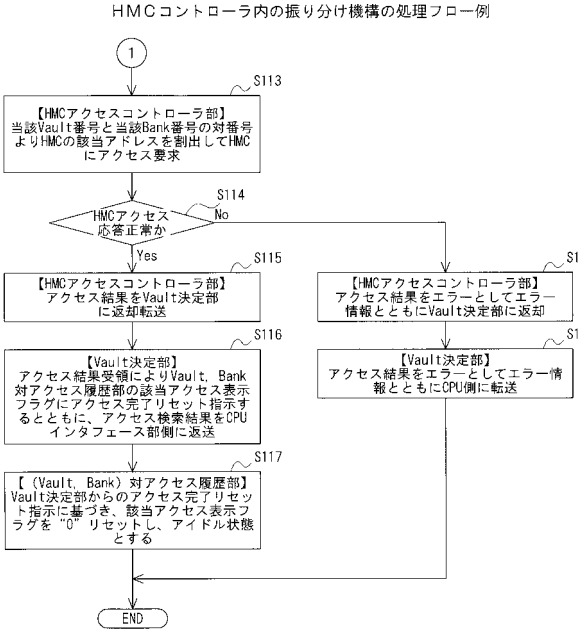
【図４】


（Ｖａｕｌｔ，Ｂａｎｋ）対アクセス履歴部の構成例
アクセス表示フラグ（０：アイドル状態、１：ビジー状態）
Vault 1
Vault 2
Vault S
Bank 1
Bank 2
Bank 3
Bank N
メモリアクセス中でないアイドル状態（"0"）のVaultをアクセス先として選択
アクセス該当Bank

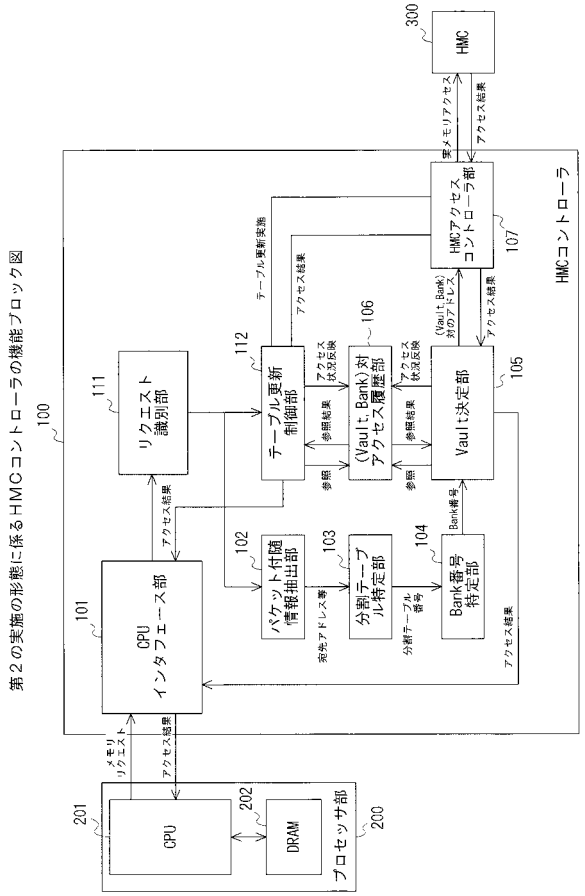
【図５】


ＨＭＣコントローラ内の振り分け機構の処理フロー例
START
S101
【CPUインタフェース部】
CPU側からのテーブル検索のための
HMCアクセスリクエストを受け付ける
S102
【CPUインタフェース部】
受け付けたアクセスリクエストを
パケット付随情報抽出部に転送
S103
【パケット付随情報抽出部】
検索目的に応じたテーブル検索対象
の情報を宛先IPアドレス等から抽出
S104
【分割テーブル特定部】
抽出情報からハッシュ計算により
分割テーブル番号（1～Ｎ）を特定
S105
【Bank番号特定部】
分割テーブル番号からHMCの
アクセスBank番号（Bank番号1～Ｎ）
を特定
S106
【Vault決定部】
全vault内の当該Bank番号横通しで、
アイドル状態のVault特定のため、
（Vault,Bank）対アクセス履歴部に
参照要求
【（Vault，Bank）
対アクセス履歴部】
当該BankでVaultのアクセス表示
フラグをVault1からVaultSまで順次確認
S107
S108
アイドル状態で
あるか（フラグ=0）
No
S110
一定時間W
待機
Yes
S109
アイドル状態を最初に発見した
Vault番号をアクセス対象とする
S111
該当Vault番号をVault決定部処理に
参照結果として返却。
直後に、該当アクセス表示フラグを
"1"のビジー状態にする
S112
【Vault決定部】
当該Vault番号と当該Bank番号の対番号
をHMCアクセスコントローラ部にアクセス
要求
1
1

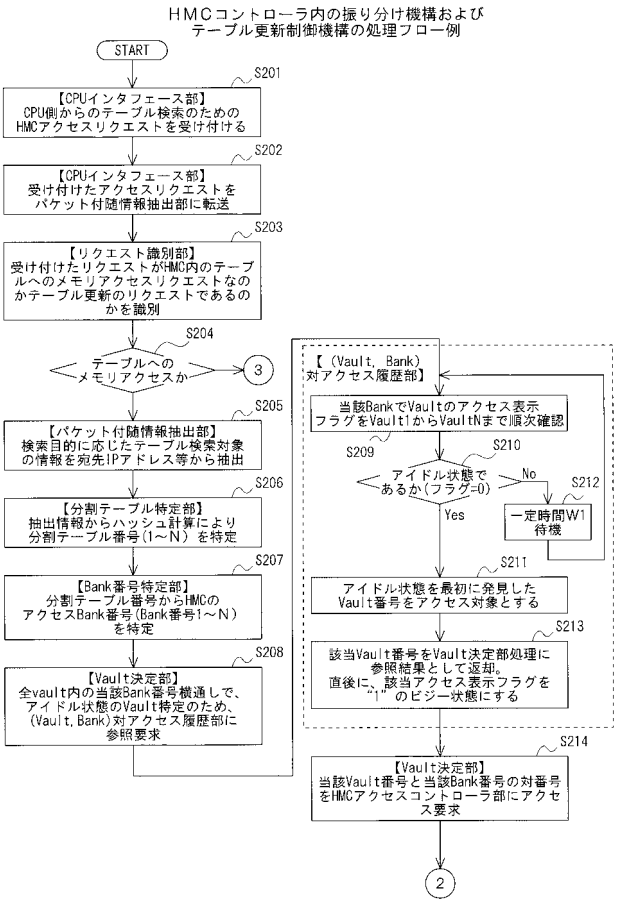
【図６】


ＨＭＣコントローラ内の振り分け機構の処理フロー例
1
S113
【HMCアクセスコントローラ部】
当該Vault番号と当該Bank番号の対番号
よりHMCの該当アドレスを割出してHMC
にアクセス要求
S114
HMCアクセス
応答正常か
No
Yes
S115
【HMCアクセスコントローラ部】
アクセス結果をVault決定部
に返却転送
S116
【Vault決定部】
アクセス結果受領によりVault，Bank
対アクセス履歴部の該当アクセス表示
フラグにアクセス完了リセット指示す
るとともに、アクセス検索結果をCPU
インタフェース部側に返送
S117
【（Vault，Bank）対アクセス履歴部】
Vault決定部からのアクセス完了リセッ
ト指示に基づき、該当アクセス表示フ
ラグを"0"リセットし、アイドル状態
とする
END
S118
【HMCアクセスコントローラ部】
アクセス結果をエラーとしてエラー
情報とともにVault決定部に返却
S119
【Vault決定部】
アクセス結果をエラーとしてエラー情
報とともにCPU側に転送

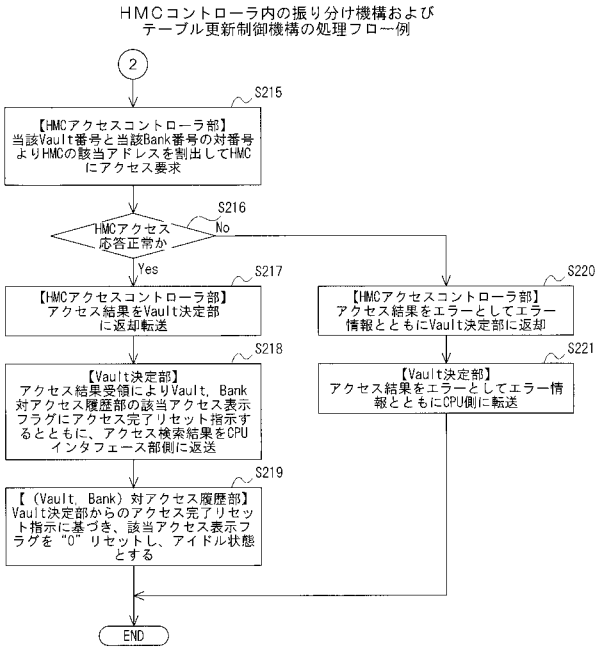
【図７】


第２の実施の形態に係るHMCコントローラの機能ブロック図
HMC
HMCアクセス
コントローラ部
HMCコントローラ
リクエスト
識別部
テーブル更新
制御部
(Vault, Bank)対
アクセス履歴部
Vault決定部
CPU
インタフェース部
パケット付随
情報抽出部
分割テーブル
特定部
Bank番号
特定部
CPU
DRAM
プロセッサ部

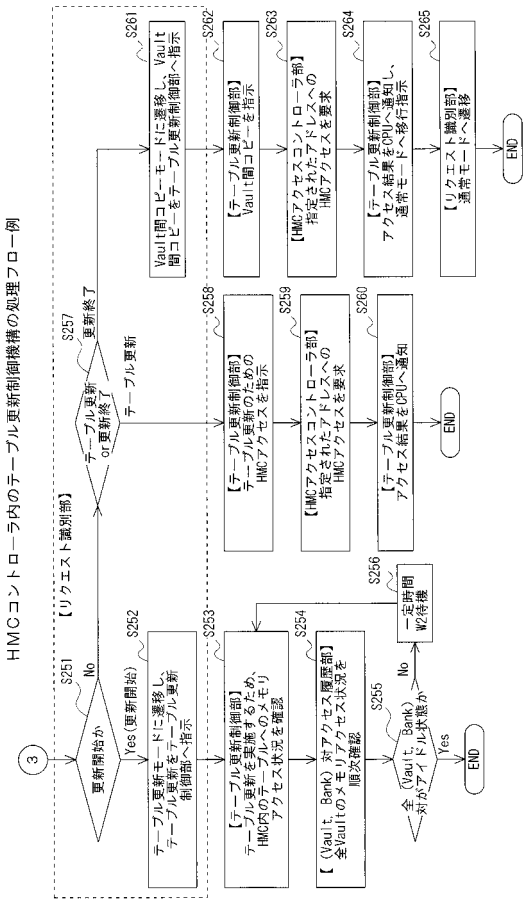
【図８】


HMCコントローラ内の振り分け機構および
テーブル更新制御機構の処理フロー例
START
【CPUインタフェース部】
CPU側からのテーブル検索のための
HMCアクセスリクエストを受け付ける
【CPUインタフェース部】
受け付けたアクセスリクエストを
パケット付随情報抽出部に転送
【リクエスト識別部】
受け付けたリクエストがHMC内のテーブル
へのメモリアクセスリクエストなの
かテーブル更新のリクエストであるの
かを識別
テーブルへの
メモリアクセスか
【パケット付随情報抽出部】
検索目的に応じたテーブル検索対象
の情報を宛先IPアドレス等から抽出
【分割テーブル特定部】
抽出情報からハッシュ計算により
分割テーブル番号(1～N) を特定
【Bank番号特定部】
分割テーブル番号からHMCの
アクセスBank番号(Bank番号1～N)
を特定
【Vault決定部】
全vault内の当該Bank番号横通しで、
アイドル状態のVault特定のため、
(Vault, Bank)対アクセス履歴部に
参照要求
【(Vault, Bank)
対アクセス履歴部】
当該BankでVaultのアクセス表示
フラグをVault1からVaultNまで順次確認
アイドル状態で
あるか(フラグ=0)
No
Yes
一定時間W1
待機
アイドル状態を最初に発見した
Vault番号をアクセス対象とする
該当Vault番号をVault決定部処理に
参照結果として返却。
直後に、該当アクセス表示フラグを
"1"のビジー状態にする
【Vault決定部】
当該Vault番号と当該Bank番号の対番号
をHMCアクセスコントローラ部にアクセス
要求

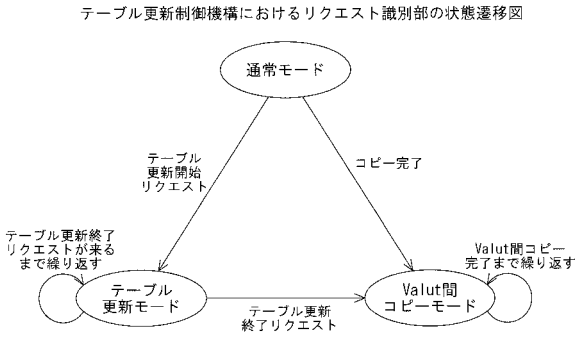
【図９】


HMCコントローラ内の振り分け機構および
テーブル更新制御機構の処理フロー例
【HMCアクセスコントローラ部】
当該Vault番号と当該Bank番号の対番号
よりHMCの該当アドレスを割出してHMC
にアクセス要求
HMCアクセス
応答正常か
No
Yes
【HMCアクセスコントローラ部】
アクセス結果をVault決定部
に返却転送
【Vault決定部】
アクセス結果受領によりVault, Bank
対アクセス履歴部の該当アクセス表示
フラグにアクセス完了リセット指示す
るとともに、アクセス検索結果をCPU
インタフェース部側に返送
【(Vault, Bank) 対アクセス履歴部】
Vault決定部からのアクセス完了リセッ
ト指示に基づき、該当アクセス表示フ
ラグを"0"リセットし、アイドル状態
とする
END
【HMCアクセスコントローラ部】
アクセス結果をエラーとしてエラー
情報とともにVault決定部に返却
【Vault決定部】
アクセス結果をエラーとしてエラー情
報とともにCPU側に転送

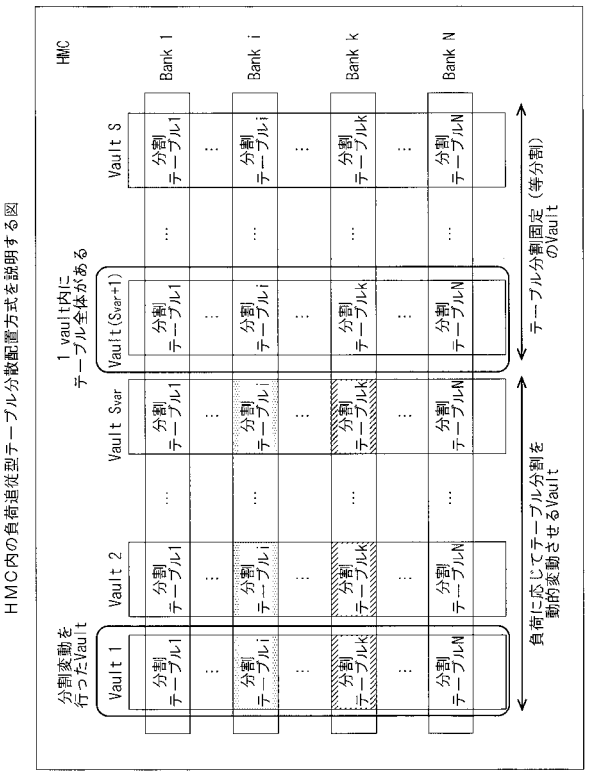
【図１０】


HMCコントローラ内のテーブル更新制御機構の処理フロー例
【リクエスト識別部】
更新開始か
No
Yes(更新開始)
テーブル更新モードに遷移し、
テーブル更新を
制御部へ指示
Vault間コピーモードに遷移し、Vault
間コピーをテーブル更新制御部へ指示
【テーブル更新制御部】
Vault間コピーを指示
【HMCアクセスコントローラ部】
指定されたアドレスへの
HMCアクセスを要求
【テーブル更新制御部】
アクセス結果をVault決定部へ通知し、
通常モードへ移行指示
【リクエスト識別部】
通常モードへ移行
END
テーブル更新
or更新終了
更新終了
【テーブル更新制御部】
テーブル更新のための
HMCアクセスを指示
【HMCアクセスコントローラ部】
指定されたアドレスへの
HMCアクセスを要求
【テーブル更新制御部】
アクセス結果をCPUへ通知
END
【テーブル更新制御部】
テーブル更新を実施するため、
HMC内のテーブルへのメモリ
アクセス状況を確認
【(Vault, Bank) 対アクセス履歴部】
全Vaultのメモリアクセス状況を
順次確認
全(Vault, Bank)
対がアイドル状態か
No
Yes
一定時間
W2待機
END

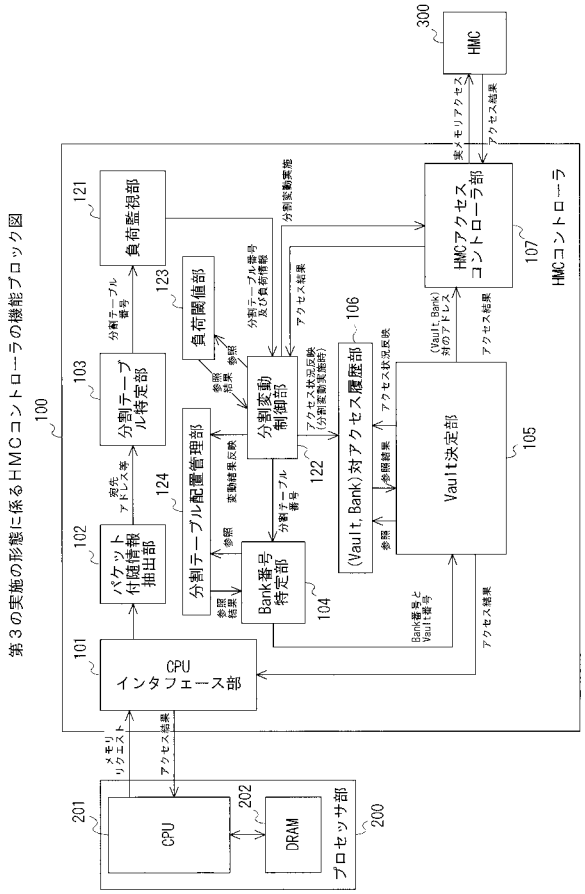
【図１１】



【図１２】



【図１３】



【図１４】

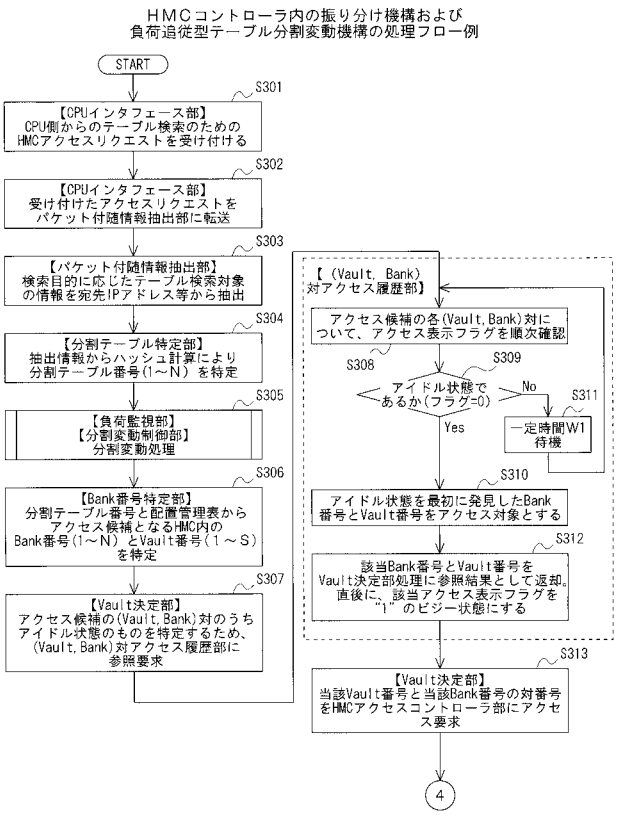
分割テーブル配置管理部がもつ配置管理表の一例

Bank＼Vault	1	…	Svar	Svar+1	…	S
1	1，0			1，0	1，0	1，0
…						
i	i，1			i，0	i，0	i，0
…						
k	k，2			k，0	k，0	k，0
…						
N	N，0			N，0	N，0	N，0

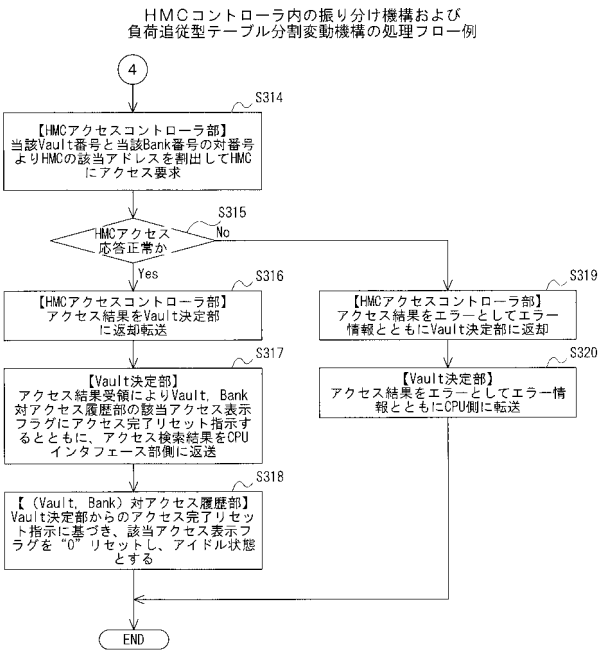
分割変動によりbank iの
分割テーブルをbank kに
上書きコピーした場合

(vault, bank)対ごとに、(配置する分割テーブル番号，分割変動フラグ)を保持
分割変動フラグは
0：変動未実施
1：変動実施・負荷最大
2：変動実施・負荷最小

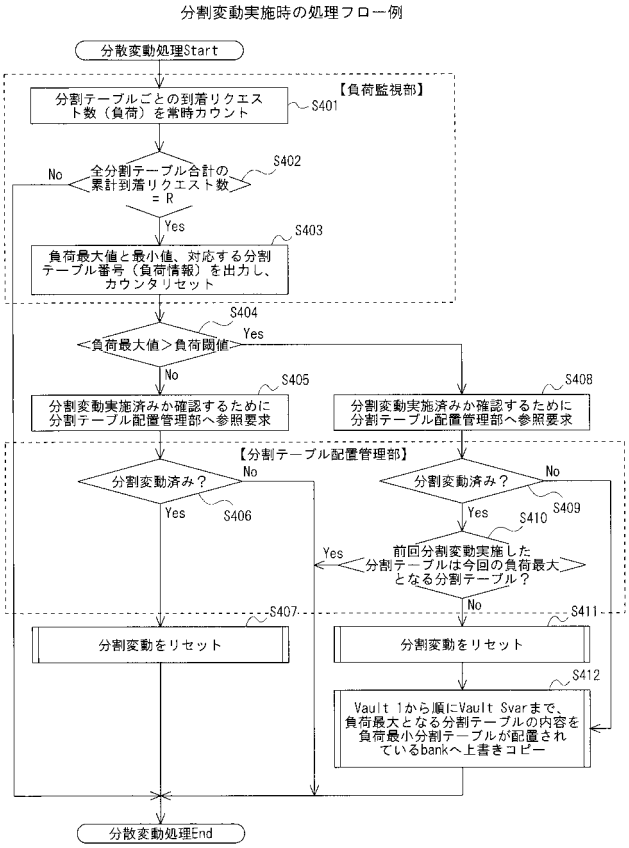
【図１５】



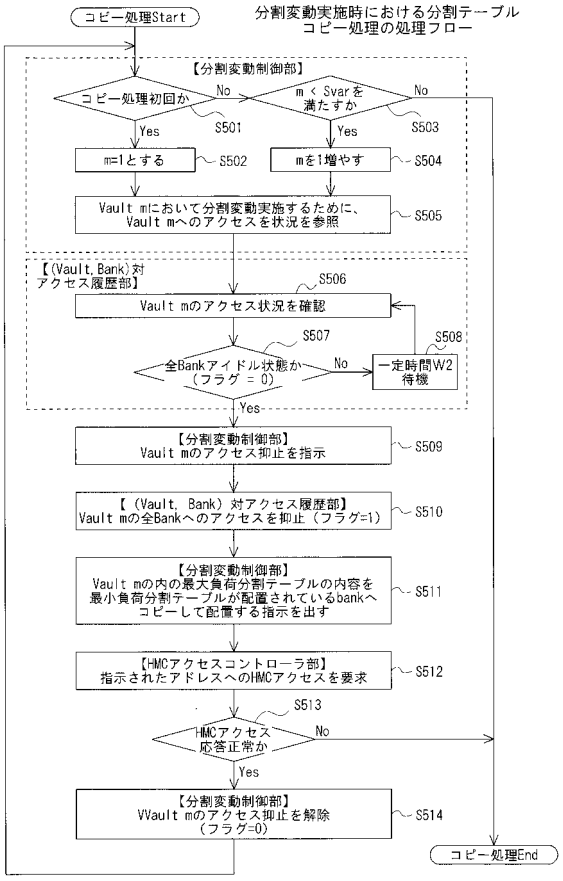
【図１６】



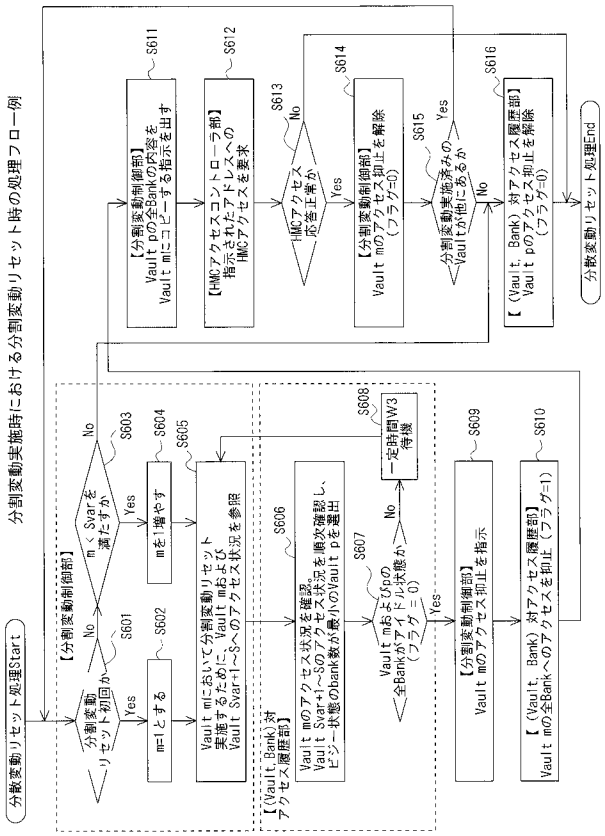
【図１７】



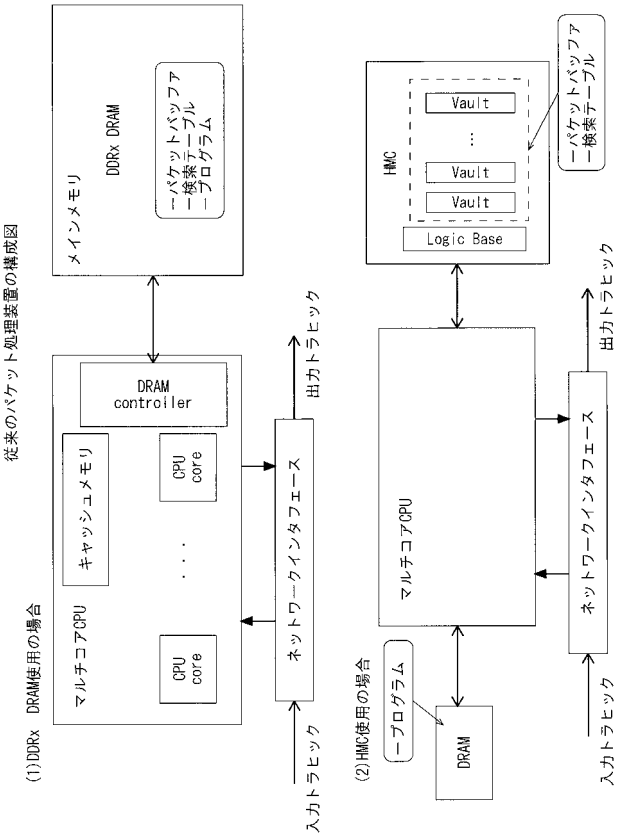
【図１８】



【図１９】


分散変動リセット処理Start
分割変動実施時における分割変動リセット時の処理フロー例
【分割変動制御部】
分割変動リセット初回か
No
Yes
S601
m=1とする
S602
m＜Svarを満たすか
No
Yes
S603
mを1増やす
S604
Vault mにおいて分割変動リセット実施するために、Vault mおよびVault Svar+1～Sへのアクセス状況を参照
S605
【(Vault, Bank)対アクセス履歴部】
Vault mのアクセス状況を確認。Vault Svar+1～Sのアクセス状況を順次確認し、ビジー状態のbank数が最小のVault pを選出
S606
Vault mおよびpの全Bankがアイドル状態か（フラグ＝0）
No
Yes
S607
一定時間W3待機
S608
【分割変動制御部】Vault mのアクセス抑止を指示
S609
【(Vault, Bank)対アクセス履歴部】Vault mの全Bank～のアクセスを抑止（フラグ=1）
S610
【分割変動制御部】Vault pの全Bankの内容をVault mにコピーする指示を出す
S611
【HMCアクセスコントローラ部】指示されたアドレスへのHMCアクセスを要求
S612
HMCアクセス応答正常か
No
Yes
S613
【分割変動制御部】Vault mのアクセス抑止を解除（フラグ=0）
S614
分割変動実施済みのVaultが他にあるか
Yes
No
S615
【(Vault, Bank)対アクセス履歴部】Vault pのアクセス抑止を解除（フラグ=0）
S616
分散変動リセット処理End

【図２０】


従来のパケット処理装置の構成図
(1)DDRx DRAM使用の場合
メインメモリ
DDRx DRAM
－パケットバッファ
－検索テーブル
－プログラム
マルチコアCPU
キャッシュメモリ
DRAM controller
CPU core
CPU core
ネットワークインタフェース
入力トラヒック
出力トラヒック
(2)HMC使用の場合
－プログラム
DRAM
マルチコアCPU
HMC
Vault
Vault
Vault
Logic Base
－パケットバッファ
－検索テーブル
ネットワークインタフェース
入力トラヒック
出力トラヒック

フロントページの続き

(72)発明者　大木　英司
京都府京都市左京区吉田本町３６番地１　国立大学法人京都大学内
(72)発明者　何　馥君
京都府京都市左京区吉田本町３６番地１　国立大学法人京都大学内

Ｆターム(参考) 5B160 CA12
5K030 GA01 HA08 KA01 KA02 KA11