

学位論文題名

Study on Memory Access Arbitrating Scheme for Vector Processor Type Supercomputer

(ベクトルプロセッサ型スーパーコンピュータに
おけるメモリ競合調停方式に関する研究)

学位論文内容の要旨

近年、ベクトルプロセッサ型のスーパーコンピュータへの要求性能は急速に高まっている。各スーパーコンピュータベンダーではこの要求に応えるために、1プロセッサの同時に動作するベクトル演算器の数を増やすことによって演算性能を高めるアプローチと、メモリを共有するプロセッサ数を増やして演算性能を高めるアプローチの両方を採用している。前者のアプローチによる並列処理を要素並列処理、後者のアプローチのメモリシステムをメモリ共有マルチプロセッサ(TCMP)と呼ぶ。

ユーザがあるプログラムをベクトルプロセッサ型のスーパーコンピュータ上でデバッグし、チューニングする場合、まず1台のベクトルプロセッサでデバッグを行う。その後でベクトル演算器を多く持つ上位モデルやTCMPモデルにプログラムを移行して実運用などに用いる場合が多い。また、一つのモデルで開発したプログラムを様々な種類のモデル上で運用を行ったりする場合もある。このとき、モデル間で主記憶性能に著しい差がないことが使いやすいスーパーコンピュータの条件となる。

このようなユーザニーズに応えるには、メモリ競合による性能低下を最小限に抑えることと、メモリ構成の異なるシステムの間で性能低下の生じるメモリアクセスストライドが一定であることである。

しかし各スーパーコンピュータベンダーが採っている上記いずれの性能向上アプローチにおいてもメモリ競合の発生確率は高くなることと、前者のアプローチではメモリバンク数の異なるシステム間で性能低下の発生するメモリアクセスストライドが異なるという問題があった。本論文では、ベクトル演算器を増やした場合の一つのメモリパイプラインにおけるメモリ競合と、複数のベクトルプロセッサ間のメモリ競合を別々に解析し、それぞれのメモリ競合を独立して解決する2つのメモリ競合調停方式を提案する。

一つのメモリアクセスパイプラインのメモリ競合は、単一ベクトルロード/ストア命令内のメモリ競合に、複数のベクトルプロセッサ間のメモリ競合は複数のベクトルロード/ストア命令間のメモリ競合に置き換えることができる。

単一ベクトルロード/ストア命令内のメモリ競合については、トランジェント集中による性能低下を抑えながら、パーマネント集中の発生しないメモリアクセスストライドをメモリバンク数に依存させない方式: Scalable Parallel Memory Architecture with a Skew Scheme (SPMA) を提案した。パーマネント集中とは、並列にアクセス可能な複数のメモリバンクのうちのいくつかのメモリバンクにアクセスが恒久的に集中することによって発生する競合であり、トランジェント集中とは、いくつかのメモリバンクに一時的にメモリアクセスが集中する競合である。従来のスキュースキームではメモリバンクへのアドレス割り

付けにおいて、アドレスシフトの周期（スキュー係数）をメモリバンク数に等しくした場合が最も効率がよいとしていた。スキュー係数をメモリバンク数よりも大きな値とすると、パーマネント集中が低下する代わりにトランジェント集中が増大するためである。

本論文ではスキュー係数をメモリバンク数より大きくしてパーマネント集中を削減し、メモリバンク入り口のバッファを増やしてトランジェント集中を吸収することによって、スキュー係数がメモリバンク数に依存しないスキュースキームを提案した。この新しいスキュースキームを用いることによって、メモリバンク数が異なる複数のシステム間でパーマネント集中の発生しないメモリアクセスストライドを等しくできることを示した。

一方複数のベクトルロード／ストア命令間のメモリ競合を解決する方式として、Instruction-Based Variable Priority Scheme (IBVPS) を提案する。一般にベクトルロード／ストア命令間のメモリ競合の調停には2つのアプローチ：リクエストレベルプライオリティ制御と固定プライオリティがある。しかし両調停方式とも著しい性能低下を生じる可能性がある。リクエストレベルプライオリティ制御ではMutual Obstructionが、固定プライオリティではMemory Bank Monopolyが性能低下を招く。ここでMutual Obstructionとは異なるベクトルロード／ストア命令間で互いにメモリアクセスの邪魔をし合うことを指し、Memory Bank Monopolyとは高いプライオリティを持つベクトルロード／ストア命令が特定のメモリバンク集中してアクセスするために、他の命令がアクセスできなくなる状態を指す。

IBVPSでは、(a)各ベクトルロード／ストア命令の処理終了毎に全メモリアクセスパイプラインのプライオリティを変更し、(b)メモリバンクを占有する可能性のある非連続アクセスベクトルロード／ストア命令の場合には、命令実行中に1回以上全メモリアクセスパイプラインのプライオリティを変更する。IBVPSを用いると、全メモリアクセスパイプラインのプライオリティを一度に変更することによって、Mutual Obstructionを大幅に削減できる。また、特定のベクトルロード／ストア命令によってメモリバンクを占有される可能性も低くなり、Memory Bank Monopolyを低下させることができる。

これによりTCMP型のベクトルスーパーコンピュータにおいて高い実効メモリ性能を達成できることを示した。

このようにベクトルプロセッサにおけるメモリアクセス競合を、一つのベクトルパイプラインにおけるメモリアクセス競合と、複数のプロセッサ間のメモリメモリアクセス競合とに分けて各々を解析し、ベクトルプロセッサにおけるメモリ競合を解決するメモリ制御方式を提案した。本方式により、同時に動作可能なベクトル演算器の数やベクトルプロセッサの数に依存せず高いスループットを得られるベクトルプロセッサ型スーパーコンピュータを開発可能とした。

学位論文審査の要旨

主査 教授 青木 由直
副査 教授 栃内 香次
副査 教授 北島 秀夫
副査 助教授 川嶋 稔夫

学位論文題名

Study on Memory Access Arbitrating Scheme for Vector Processor Type Supercomputer

(ベクトルプロセッサ型スーパーコンピュータに
おけるメモリ競合調停方式に関する研究)

ベクトルプロセッサ型スーパーコンピュータに対する要求性能は、近年急速に高まってきたおり、1プロセッサで同時に動作するベクトル演算器の数を増やすアプローチ（要素並列処理）と、メモリを共有するプロセッサ数を増やすアプローチ（メモリ共有マルチプロセッサ、TCMP）による演算性能の向上が図られている。

ところで、ユーザの一般的なプログラム開発過程をみると、まず1台のベクトルプロセッサでデバッグやチューニングをした後、ベクトル演算器を多く持つ上位モデルやTCMPモデルに移行して実運用するケースが多い。また、特定のモデル上で開発したプログラムを様々な種類のモデルに移行して運用するケースも多い。このような場合、モデル間で主記憶性能に著しい差がないことがユーザにとっての使いやすさの条件となる。

このようなニーズに応えるには、メモリ競合による性能低下を最小限に抑えることと、メモリ構成の異なるシステムの間で性能低下の生じるメモリアクセスストライドが一定になっていることが条件である。しかしながら、要素並列処理やメモリ共有マルチプロセッサによる性能向上を採用した場合、メモリ競合の発生確率が高くなること、性能低下のおきるメモリアクセスストライドがメモリバンク数に依存することの、2つの問題点があった。

本論文において著者は、ベクトル演算器を増やした場合の一つのメモリパイプラインにおけるメモリ競合と、複数のベクトルプロセッサ間のメモリ競合を解析するための評価法を明らかにするとともに、それぞれのメモリ競合を独立して解決するメモリ競合調停方式を提案している。

第1章は序論で、本論文の背景を説明するとともに、ベクトル型計算機の概要と速度向上の手法について述べている。

第2章では、単一ベクトルロード/ストア命令内のメモリ競合について、トランジェント集中による性能低下を抑えながら、パーマネント集中の発生しないメモリアクセスストライドをメモリバンク数に依存させない方式：Scalable Parallel Memory Architecture

with a Skew Scheme (SPMA)を提案している。従来法では、メモリバンクへのアドレス割り付けにおいて、アドレスシフトの周期（スキュー係数）をメモリバンク数に等しくして効率を向上していた。これはスキュー係数をメモリバンク数よりも大きな値にすると、パーマネント集中が低下する代わりにトランジェント集中が増大するためである。

これに対し著者は、スキュー係数をメモリバンク数より大きくしてパーマネント集中を削減し、メモリバンク入り口のバッファを増やしてトランジェント集中を吸収することで、スキュー係数がメモリバンク数に依存しないことを明らかにしている。このスキュースキームを用いることで、メモリバンク数が異なる複数のシステム間でパーマネント集中の発生しないメモリアクセスストライドを等しくできることが実験的に示されている。

第3章では、複数のベクトルロード/ストア命令間のメモリ競合を解決する方式として、Instruction-Based Variable Priority Scheme (IBVPS)を提案している。一般にベクトルロード/ストア命令間のメモリ競合の調停には2つのアプローチ：リクエストレベルプライオリティ制御と固定プライオリティがあるが、リクエストレベルプライオリティ制御ではMutual Obstructionが、固定プライオリティではMemory Bank Monopolyが性能低下を招く可能性がある。

IBVPSでは、(a)各ベクトルロード/ストア命令の処理終了毎に全メモリアクセスパイプラインのプライオリティを変更し、(b)メモリバンクを占有する可能性のある非連続アクセスベクトルロード/ストア命令の場合には、命令実行中に1回以上全メモリアクセスパイプラインのプライオリティを変更している。IBVPSを用いると、全メモリアクセスパイプラインのプライオリティを一度に変更することによって、Mutual Obstructionを大幅に削減できる。また、特定のベクトルロード/ストア命令によってメモリバンクを占有される可能性も低くなり、Memory Bank Monopolyを低下させることができる。これによりTCMP型のベクトルスーパーコンピュータにおいて高い実効メモリ性能を達成できることが示されている。

第4章は結論であり、本研究の総括を行っている。

このように本論文で著者は、ベクトルプロセッサにおけるメモリアクセス競合を、一つのベクトルパイプラインにおけるメモリアクセス競合と、複数のプロセッサ間のメモリメモリアクセス競合とに分けて解析し、メモリ競合を解決する制御方式を提案している。提案方式を用いることで、スループットがベクトル演算器の数やベクトルプロセッサの数に依存しないベクトルプロセッサ型スーパーコンピュータが開発されるようになった。

これを要するに著者は、ベクトルプロセッサのアーキテクチャにおける、メモリ競合調停方式に関する新知見を得たものであり、計算機工学に貢献するところ大なるものがある。よって著者は、北海道大学博士（工学）の学位を授与される資格があるものと認める。