

メニーコアプロセッサにおける仮想チャネルの効率的利用法

池田 貴一 †

吉瀬 謙二 ‡

東京工業大学 情報工学科 †

東京工業大学 大学院情報理工学研究科 ‡

1 はじめに

プロセッサの性能向上と消費電力の削減を目的に、広い分野で複数のコアを搭載するマルチコアプロセッサが主流となりつつある。今後は、半導体の集積度向上により、多くのコアを集積するメニーコアプロセッサへ向かうと考えられる。メニーコアプロセッサにおいて、コア間の通信は、一般的に Network on Chip (NoC) と呼ばれる通信路を用いておこなわれる [2]。本稿では、NoC におけるルータの仮想チャネルを効率的に利用することで、性能向上を目指す。通信の状況に応じ、仮想チャネル数とバッファサイズを動的に変更することで性能向上を目指す手法を提案する。

2 ルータアーキテクチャのモデル

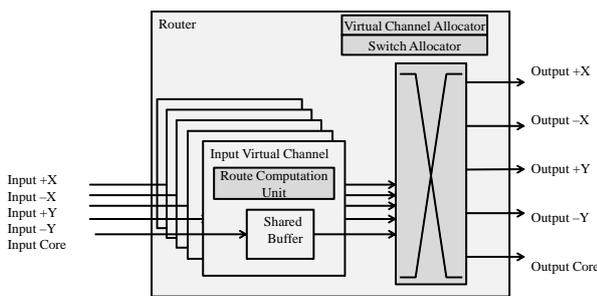


図 1: ルータアーキテクチャのモデル。

図 1 に想定するルータアーキテクチャを示す。想定するルータアーキテクチャは入力バッファルータを前提としている。従来の入力バッファルータと異なる点は、仮想チャネルが本来は分散されたバッファを各仮想チャネルごとに持つのに対して、ポートごとに共有されたバッファを仮想チャネルごとに割り当てて利用するというものである。

2次元メッシュであることを前提としており、XY次元順ルーティングで、フロー制御はクレジットベース、LT+RC,VA,SA,STの4段パイプラインである。また本稿において、仮想チャネルには1つのパケットが一度割り当てられたら全てのフリットが出ていくまで他のパケットが入らないものとする。

3 仮想チャネルの利用

仮想チャネルは、ポートにおけるチャネルの競合を低減し、物理チャネルのバンド幅を有効に利用する目的で利用される。実行性能に影響する仮想チャネルのパラメー

タとして仮想チャネルの本数とバッファ量が挙げられる。通信量が多いか少ないかによって、仮想チャネルの効率の良い利用方法が異なってくる [1]。

通信量が少ない場合、仮想チャネルの競合の頻度は少ない。したがって仮想チャネルの本数は少ない数でも十分である。代わりに、仮想チャネルに割り当てるバッファ量を大きくすることで、パケットがスムーズに流れることができる。

通信量が多い場合、仮想チャネルに割り当てられるパケットが、1パケットのみなので、複数のパケットによる仮想チャネルの競合が増加する。そのためバッファサイズを大きくしてもあまり改善が見られない。代わりに、仮想チャネルの数を多くすることで、複数のパケットによる仮想チャネルの競合を低減することで、通信を改善することができる。

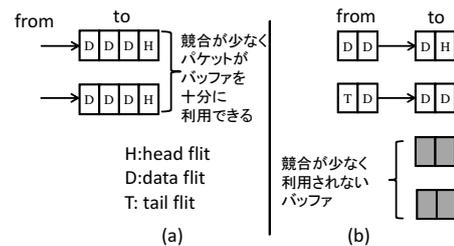


図 2: 通信量が少ない場合の仮想チャネル利用例。

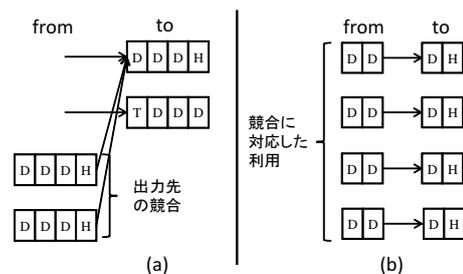


図 3: 通信量が多い場合の仮想チャネル利用例。

図 2 に通信量が少ない場合の仮想チャネルの利用例を示す。隣接するルータからのフリットのフローを示しており、1つのブロックは1つのバッファ領域である。(a)は仮想チャネルの本数よりもバッファ量を大きくすることに注目している。(b)はバッファ量よりも仮想チャネルの本数を多くすることに注目している。

競合するパケットが少ない(b)の場合、利用されない仮想チャネルが存在する一方で、バッファ量不足によりパケットの流れが不十分である。(a)の場合、バッファを大きくすることでフリットのフローを高めることができるのでレイテンシ低減することができると思われる。

一方、通信量が多い場合において、仮想チャネルの本

The efficient way of using Virtual Channels on Many-Core Processor

Takakazu IKEDA†, and Kenji KISE‡

†Department of Computer Science, Tokyo Institute of Technology

‡Graduate School of Information Science and Engineering, Tokyo Institute of Technology

数が多いと仮想チャネルの競合に対応できる。

図3は通信量が多い場合の仮想チャネルの利用を表したものである。(a)の場合、仮想チャネルが空になるまで次のパケットが入って来ないので、仮想チャネルの競合が多発し性能が低下する。したがって通信量が多い場合は、(b)のように多くの仮想チャネルを用意したほうが競合が減少し性能低下を抑えることができる。これにより、スループットの向上が見込めると考えられる。

以上のことから、この2つの条件にあった仮想チャネルの利用を効率良く行うことで性能向上が見込める。

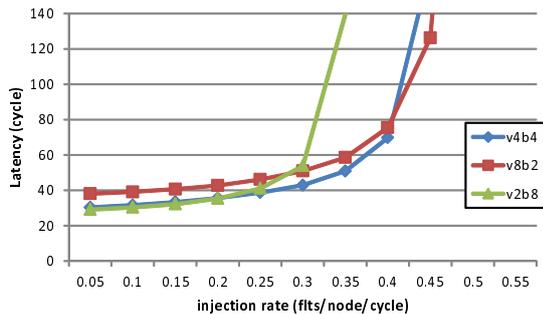


図4: コンプリメント通信における性能評価。

図4にコンプリメント通信における仮想チャネルの本数とバッファによる影響を示す。 v が仮想チャネルの数、 b が仮想チャネル当たりのバッファサイズである。図4より、チャネル数が大きいほどスループットが向上し、バッファが大きいほど、通信量が少ない場合のレイテンシが小さいことが分かる。このことから、通信量が少ない場合、仮想チャネルのバッファ量を大きくし、通信量が大きい場合、仮想チャネルの数を多くすることで性能向上が可能であることが分かる。

4 提案手法

仮想チャネルの効率的利用を目的として、仮想チャネルの本数と仮想チャネル当たりのバッファサイズを動的に変更する。

通信量の少ない場合は仮想チャネルの本数を半分に減らし、減らした仮想チャネル分のバッファを残りの仮想チャネルのバッファとして利用することでレイテンシを改善する。また、通信量の多い場合は、1つのチャネルのバッファを2つに分割し、利用できる仮想チャネルの本数を増加させることでスループットの向上を目指す。

隣接するルータからのVAリクエスト数とVAのコミット率(VAリクエスト当たりどれだけVAが成功したか)を指標として、仮想チャネルの本数とバッファ量を定める。双方を用いた理由としては、VAのリクエストが多くても、コミット率が高ければ、仮想チャネルが効率よく利用できていると考えられるからである。

また、コミット率が著しく悪くなると、仮想チャネルの競合が多くなっていると考えられるので、仮想チャネルの本数を増加させ、ボトルネックであるリンクを解消する。VAのリクエスト数が少ない場合、VAのコミット率は高くなると考えられるが、仮想チャネルの利用効

率が低下する場合は、仮想チャネルの数を減らし、バッファ量を増加させることで利用効率を向上させる。

5 実装と評価の方針

ここで、共有バッファの利用方法について述べる。仮想チャネルの数を n とし、ここでは n は2の倍数数であるという制約を設ける。通常の利用の場合、共有バッファにおいて仮想チャネルのバッファは共有バッファを均等に n 分割した領域が与えられる。この状態により、通信量の多い場合に対応する。通信量が少ない場合、仮想チャネルのバッファ量を増加させて、レイテンシを低減する。仮想チャネルの競合が発生する頻度が少ないので仮想チャネルの本数を減らし、利用されなくなるバッファを割り当てることでレイテンシの低減を実現する。

提案手法においては2つの仮想チャネルが1つのペアを組む。バッファ量を増加させたい場合は、ペアの片方の仮想チャネルの利用を停止し、停止した仮想チャネルのバッファをもう一方の仮想チャネルのバッファに割り当てる。

性能評価としては、パケットの注入レートが変化するようなモデルを用いる。

6 まとめと今後の課題

メニーコアプロセッサの性能向上を目的として、通信状況に応じて、仮想チャネルの本数とバッファ量を動的に変更し、通信性能の向上を実現する手法を提案した。

今後の課題として、ソフトウェアシミュレータによる、実装。通信状況を加味した、仮想チャネルの本数とバッファサイズ変更のトレードオフのスケジューリング、などが挙げられる。

参考文献

- [1] Chrysostomos A. Nicopoulos, 他. ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers. 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)
- [2] William Dally, William Dally. Principles and Practices of Interconnection Networks. pp. 325-348
- [3] Jose Flich, 他. Designing Network On-Chip Architectures in the Nanoscale Era. pp. 67-88.
- [4] Yi Xu, 他. Simple virtual channel allocation for high throughput and high frequency on-chip routers. High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on .
- [5] Thomas Moscibroda, Onur Mutlu. A case for bufferless routing in on-chip networks. ISCA 2009 The 36th International Symposium on Computer Architecture .