

冗長実行時の SmartCore システムの性能評価

池田 貴一^{†1} 佐藤 真平^{†2,†3} 吉瀬 謙二^{†2}

我々は、メニーコアプロセッサにおいて、多数のコアの利用と、高機能ルータによって性能向上を目指す、SmartCore システム (Smart many-core system with redundant cores and multifunction routers) を提案している。本稿では、SmartCore システムの利用において、冗長実行させるアプリケーションの配置を変えて性能評価する。冗長実行時のパケットの依存関係により、デッドロックが起きる可能性があるが、依存を取り除きデッドロックを回避することを示す。冗長実行させるノードの距離に着目して性能評価を行った結果、性能評価で冗長実行させるノードを本来実行させるノードの隣に配置した場合のサイクル数は、最大で 3.7 % の増加であったのに対し、ノードすべてを一行に並べるような、冗長実行させるノードを遠くした場合の配置で、最大で約 4 倍の性能低下を示した。これより、冗長実行させるノードを離すほど性能低下が大きいことを確認した。

The evaluation of Smart Core System under the redundant executions

TAKAKAZU IKEDA,^{†1} SHIMPEI SATO^{†2,†3} and KENJI KISE^{†2}

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、広い分野にわたり、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、半導体の集積度の向上により、

さらに多くのコアを集積するメニーコアプロセッサへと向かうと考えられる。

一方、LSI の一層の微細化により、ソフトエラー率およびばらつきが増加し、信頼性の低下が深刻化している^{1)–3)}。ソフトエラーは、メモリや論理回路などでランダムに発生する一時的な誤動作である。微細化が進むことで臨界電荷量が低減すると、中性子などの影響によりビットが反転する。マルチコアやメニーコアプロセッサにおいてはプロセスの微細化と共に集積するトランジスタ数が増加するため、ソフトエラー率の増加が懸念される。同時に、電源、温度の不均一などに起因するチップ内のランダムなばらつきが信頼性の低下を招いている⁴⁾。

我々は、1 チップに搭載可能なコアが数十から数千コアというオーダーまで増え続けることを想定し、そのような豊富なコアを活用し信頼性向上を目指す SmartCore システム (Smart many-core system with redundant cores and multifunction routers)^{5),6)} を提案している。SmartCore システムは、メニーコアの空間冗長性と高機能ルータを利用し、冗長実行を行うことで、信頼性を向上させるものである。アプリケーションを実行させるために本来必要なノードをマスターノード、冗長実行のためのノードをミラーノードとすると、高機能ルータにより、マスターノードが受信するパケットの複製、ミラーノードのパケットの送信先の変更、マスターノード、ミラーノードの両パケットの比較を行う。

SmartCore システムを用いる際に、マスターノードとミラーノードの配置を柔軟に決定したいが、パケットの依存関係により、マスターノードとミラーノードの配置によってはデッドロックが起きる可能性がある。マスターノードからミラーノード、マスターノードからマスターノード、ミラーノードからマスターノードの三種類のパケットの関係性を、仮想チャネルを分け、依存関係を切ることでデッドロックを回避する。

本稿では、アプリケーションを冗長実行時の配置の違いに着目して性能評価する。

本稿の構成を以下に示す。2 章で想定するメニーコアアーキテクチャを述べる。3 章で SmartCore システムの概要を述べる。4 章で SmartCore システムを実行するアプリケーションの配置について述べる。5 章で SmartCore システムによる評価を行う。6 章で関連研究について述べる。最後に、7 章でまとめる。

2. 想定するメニーコアアーキテクチャ

この章では、今回性能評価する SmartCore システムにて、想定するアーキテクチャモデルを述べる。図 1 に、M-Core アーキテクチャのモデルを示す。チップはノード、メモリコントローラの 2 種類で構成される。ノードは X 座標と Y 座標で一意に指定される識別子 (ID) を持つ。本稿では、(X,Y) という ID を持つノードのことを **Node(X, Y)**、と表記す

^{†1} 東京工業大学 情報工学科

Department of Computer Science, Tokyo Institute of Technology

^{†2} 東京工業大学 大学院情報理工学研究所

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

^{†3} 日本学術振興会 特別研究員 (DC2)

Research Fellow of the Japan Society of the Promotion of Science (DC2)

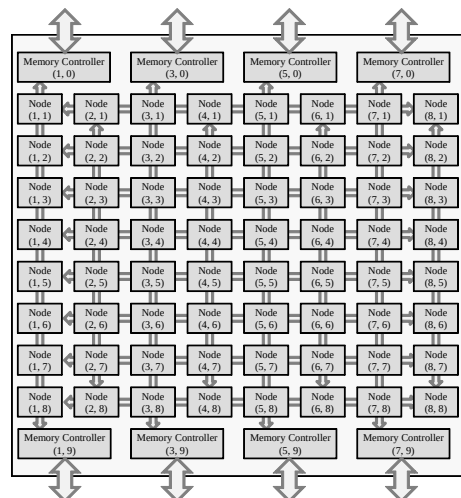


図 1 M-Core アーキテクチャ

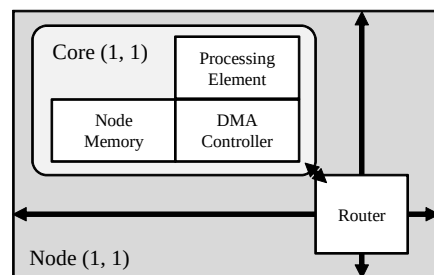


図 2 ノードの構成

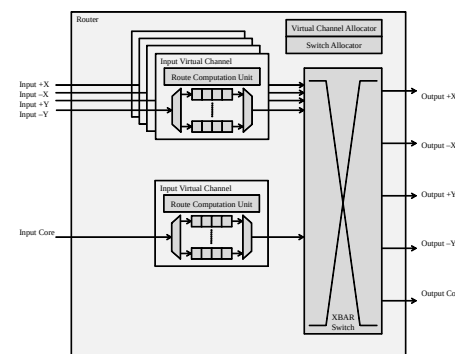


図 3 ルータアーキテクチャ

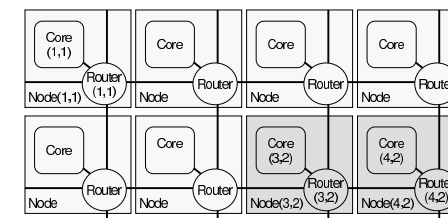


図 4 Node (3,2) と Node (4,2) による冗長実行例

る。図 1 の例では、Node(1, 1) から Node(8, 8) がノードとなる。メモリコントローラはオンチップのメモリコントローラで、オフチップのメインメモリに接続されている。それぞれのノードは 2 次元メッシュのネットワークで接続している。ノード間のデータの受け渡しは DMA 転送を用いる。

ノードの内部構成を図 2 に示す。ノードは **Processing Element**、**ノードメモリ**、**DMA コントローラ**、**ルータ**、で構成される。本稿では、Node(X, Y) のコアやルータを **Core(X, Y)**、**Router(X, Y)**、と表記する。コアは MIPS32 命令セットの RISC プロセッサ、ノードメモリはノードが持つ小規模のメモリである。DMA コントローラはノードメモリと他ノードとの DMA 転送を制御するユニットで、ルータに接続されている。コアはロード/ストア命令により自ノードのノードメモリにアクセスする。また、メモリマップド I/O を利用し、DMA コントローラに対して DMA 転送を発行する。

ルータの構成を図 3 に示す。ルータは隣接する 4 つのルータ及び DMA コントローラと接続している。スイッチング方式はワームホールスイッチング、ルーティングは、XY 次元順ルーティング、フロー制御はクレジットベースを採用する。仮想チャネルを用いており、ルータにおける、ルート計算は Look-Ahead 型の 4 段パイプラインである。ルート計算 (NRC) と仮想チャネルの割り当て (VA) を 1 サイクルで、スイッチアロケーション (SA)、スイッ

チトラバーサル (ST)、リンクトラバーサル (LT) をそれぞれ 1 サイクルで行う。

3. SmartCore システム

3.1 概要

SmartCore システムは、オペレーティング・システムなどのシステムソフトウェアと協調動作し、高機能ルータによる複数コアの多重実行によって、メニーコアプロセッサにおける信頼性の向上、チップ内ネットワークのバンド幅、レイテンシの改善を目指す仕組みである^{5),6)}。本稿では、SmartCore システムの機能の 1 つである信頼性向上の手法について述べる。

図 4 を用いて、信頼性向上の仕組みを述べる。SmartCore システムでは、高機能ルータを用いたパケットレベルでの信頼性向上を目指す。図 4 の例では、Node(3, 2) と Node(4, 2) を用いて 2 重実行をおこない、DMR(Dual Modular Redundant) を実現する。高機能ルータにおいて Node(3, 2) と Node(4, 2) が送信するパケットを比較することでエラーを検出できる。エラーが検出された場合は、実行を停止し再実行するなどの適切な回復処置が必要となる。

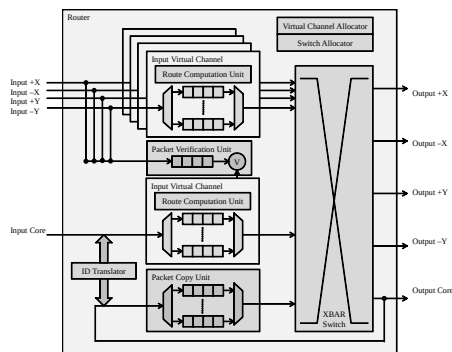


図5 SmartCore システムに利用するルータ

3.2 信頼性向上のための高機能ルータ

アプリケーションを実行するために本来必要とされるノードを**マスターノード**、多重実行のために追加されたノードを**ミラーノード**とする。SmartCore システムにおいて、以下の3つの機能を高機能ルータで実現する。

- パケットの複製
- パケットの送信先変更
- パケットの比較

アプリケーションは、マスターノードに対してのみ通信を発生させる。したがって、マスターノードとミラーノードの多重実行を維持するために、マスターノードのルータにおいて受信するパケットを複製し、ミラーノードに送信する。ミラーノードから送信されるパケットをマスターノードのルータにおいて比較し、エラー検出をおこなう。そのために、ミラーノードのルータは送信するパケットの送信先をマスターノードへと変更する。SmartCore システムのルータアーキテクチャを図5に示す。図3のルータアーキテクチャをもとに、**Packet Verification Unit**、**ID Translator**、**Packet Copy Unit**を追加する。**Packet Verification Unit**は、ミラーノードからマスターノードに届いたパケットを比較するユニットである。**Packet Copy Unit**によって、マスターノードは他のノードから届いたパケットを複製する。**ID Translator**によって、ミラーのパケットの送信先をマスターに変更し、マスターに届いたパケットを、コピーしてミラーに送る場合の送信先をミラーへと変更する。

また、仮想チャンネルは3重冗長化を前提に8本とする。マスター、ミラー間のコピーに

よる通信、ミラーから比較するためにマスターへ送るための通信向けに仮想チャンネルを通常のノード間通信とは別に確保している。

3.3 デッドロック回避へのアプローチ

ミラーノードからマスターノードへ比較のために送られるパケット通信をマージ通信、マスターノード間の通信をノード間通信、マスターノードにパケットが受信されパケットが複製されてミラーノードへ送られる通信をコピー通信と呼ぶことにする。各通信は、XY次元順ルーティングであるので、各通信ごとの経路のみではデッドロックは起こらない。

しかし、図6において、A、Bをマスターノード、A'、B'をそれぞれのミラーノードとすると、A'からAへのマージ通信、AからBへのノード間通信、BからB'のコピー通信という一連の通信経路すべてにおいて、パケットの依存関係があるので、A'からB'への一連の通信において、向きの変更が複数回行われ、XY次元順でないような振る舞いを起こす。この場合、一連の通信において、仮想チャンネルを分けることによってパケットの依存関係を取り除かなければ、デッドロックする可能性のある経路が存在する。

例えば、図8のような配置の場合、AからBへの通信と、BからAへの通信が同時に行われた場合、AはBから届いたパケットを複製し、A'へ送信する。同様にBはB'へパケットを送信する。ここで、AからA'へ通信する場合において、AからBへの通信経路の確保がなされていた場合、AからA'への通信は経路が解放されるまで待たなければならない。

しかし、AからA'への通信は、BからAへの通信に依存しているため、複製したパケットがA'へ届くまで、BからAの経路を確保したままになる。このとき、BはAから届いたパケットを複製してB'へ送る必要があるが、BからAへの通信経路が確保されたままなので、待ち状態になる。このように、互いに経路を確保し合い、デッドロックを起こす。

そのため、デッドロックを回避するために、マスターノードに届いたパケットを複製してミラーノードに送信するときに、パケットの依存関係を取り除かなければならない。そこで、文献7)で提案されているStatic Algorithmという手法を用いる。図7のように、各通信に使用する仮想チャンネルを全て別のチャンネルを用いることで、通信の依存関係を取り除き、デッドロックを回避する。図8の通信経路を例に挙げると、図9のようにA、A'間、B、B'間の通信に別のチャンネルを割り当てることで、デッドロックを回避する。SmartCore システムにおいては、各ルータの仮想チャンネルのうち、マージ通信、コピー通信向けの仮想チャンネルを1つずつ持つことで、Static Algorithmを実現する。

以下に仮想チャンネルを分け、パケットの依存関係を取り除けば、デッドロックしないことを示す。

ノード間通信からコピー通信へ移行する場合：

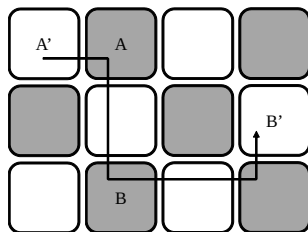


図 6 SmartCore システムによる
依存関係を持つ一連の通信経路

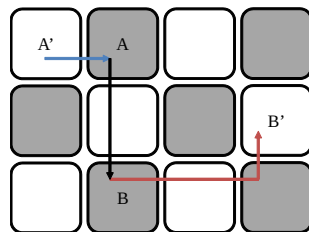


図 7 各通信の種類ごとに
仮想チャンネルを変えた場合の経路

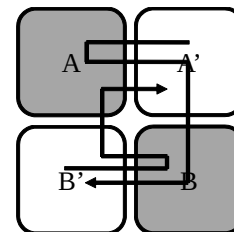


図 8 一連の通信の依存関係により
デッドロックが起こる通信例

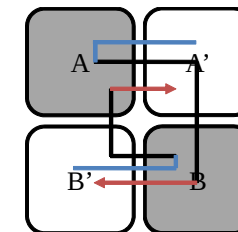


図 9 各通信の依存関係を取り除き
デッドロックが解消される例

マスターノードが受信するパケットをミラーノード方向に転送する際、ミラーノード方向のチャンネル割り当ては、通常のパケットのチャンネルを割り当てと同様の処理を行う。マスターに届いたパケットが、ミラー方向に転送されるための条件が、通常のパケットと同様のため、ノード間通信からコピー通信への移行は通信とみなすことができる。

マージ通信からノード間通信へ移行する場合：

マスターノードのルータにおいて、マスターノードとミラーノードの送信するパケットの待ち合わせが発生する。したがって、マスターノードから送信されるパケットが先に届く場合と、ミラーノードから送信されるパケットが先に届く場合の2つが考えられる。

まず、マスターから送信されるパケットが先に届く場合、マスターから送信されたパケットはネットワークに入る手前で止まる。そして、ミラーノードからのパケットが届き次第パケットの転送が再開される。この時、ミラーノードから送信されたパケットは通常のパケットと同じチャンネルの割り当てと同様の処理でノード間通信先へ転送されるため、1つのパケット通信とみなすことができる。

次に、ミラーノードから送信されるパケットが先にルータに届く場合、マスターノードからパケットが送信されるまでマージ通信間の資源を確保したまま止まる。そして、マスターからのパケットが届き次第パケットの転送が再開される。このとき、マスターノードからパケットが送信されない場合、経路が確保されたままになり、通信不能に陥る可能性がある。よって、ミラーノードがパケットを送信する場合、必ずマスターノードがパケットを送信することを保証する必要がある。

ミラーノードがあるパケットを送信するときは、ミラーノードにおいて送信以前のパケットの受信が適切に行われている。それらの受信はマスターノードを経由して転送されているため、ミラーに届いたパケットはマスターが必ず同じパケットを同じ順番で受信している。

る。SmartCore システム利用に想定されるプログラミングモデル^{*1}により有限時間内にマスターノードもミラーノードと同一のパケットを送信することが規定されている。したがって、ミラーノードがパケットを送信する場合、必ずマスターノードも同一パケットを送信するのでマージ通信の資源は必ず解放されることが保証できる。この場合もマスターから送信されるパケットの待ち合わせにより遅延が生じるが、ミラーノードから送信されたパケットはノード間通信の送信先へ転送されるため、1つのパケット通信とみなすことができる。

以上より、マージ通信、ノード間通信、コピー通信の一連の通信は1つのパケット通信とみなすことができる。

よって、SmartCore システムは、デッドロックしないことが示された。

4. SmartCore システム実行時の配置

SmartCore システムを利用する場合、バンド幅の関係性から、冗長実行させるアプリケーションをどのように配置するかによって、性能低下の度合いが変わってくる。本稿では、マスターノード間のバイセクションバンド幅の違い、マスターミラーの配置の距離、によって性能低下の度合いを調べる。

本稿のアプリケーションの実行ノードは16ノードで計算を行う。図10(a)から(e)のように5つの配置を採用する。図10における灰色のノードがマスターノードであり、白色のノードがミラーノードである。(a)の配置は、マスターノードすべてを1列に並べ、対応するミラーノードをマスターノードのとなりに配置する。(b)の配置は、マスターノードを

*1 SmartCore システムではマスターが受信するパケットをマスターのルータにおいて複製し、ミラーノードへ転送する。したがって、マスターノードと、ミラーノードのパケット受信の順番は同じであるが、タイミングが異なる。そのため、SmartCore システムを利用するためには、ミラーノードがパケットを送信する際に、これまでに受信した系列がマスターノードが受信した系列に含まれれば、タイミングは異なっても同じパケットを送信するモデルでプログラムを記述する必要がある。

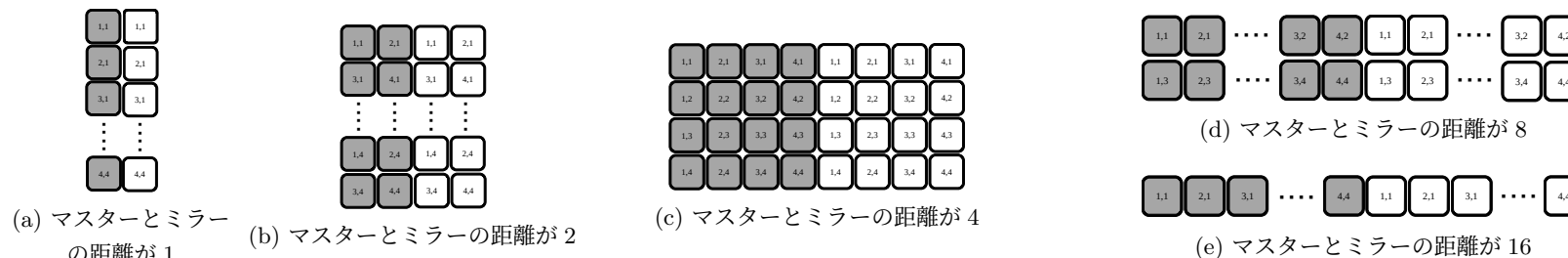


図 10 冗長実行時のアプリケーション配置例

2 列に並べ、マスター、ミラー間を距離が 2 になるようにミラーノードを配置する。(c) の配置は、マスターノードを 4 列に並べ、ミラーとの距離が 4 になるような配置である。(d) の配置は、マスターノードを 2 列に並べるが、ミラーとの距離が 8 になるような配置である。(e) の配置は、マスターノードも、ミラーノードも同じ 1 列に並べる。この場合、マスターノードと、ミラーノードの距離は 16 となる。

マスターノードのみの関係に着目すると (a) と (e)、(b) と (d) において、それぞれ、バイセクションバンド幅は同じである。また、(c) はマスターノードの実行においてバイセクションバンド幅は 5 つの内最大であるので、マスターノードのみの実行は通信による遅延が最も少ないと考えられる。

しかし、SmartCore システム実行時には、マスターノードは、ミラーノードからのパケットを比較するための待ち時間が存在する。そのため、マスターノードとミラーノードの距離は、コピー通信、マージ通信による遅延に影響を与える。

マスター、ミラー間の通信において距離に着目した場合、ミラーノードからのパケットを受けるのに最も優れているのは、距離が最も近い (a) の配置であり、最低の場合は (e) の配置である。さらに距離の関係だけでなく、複数のマスターノードとミラーノード間の通信経路が重なることによる性能変化についても述べる。(a) の場合は、マスターノードとミラーノードの間に他の通信が存在しないので、相互の通信の遅延は少なくなる。しかし、(b)、(c)、(d)、(e) の配置は、他のマスター、ミラー間の通信経路が重なる。特に (e) のような場合は、最大で 16 の経路が重なるので、通信衝突が大きくなり、性能低下が予想される。

以上から配置ごとの特性をまとめると、

- (a) の場合、マスターノード間通信におけるバイセクションバンド幅は最低であるが、

他のマスターノードとミラーノードの通信経路と重ならないので他の配置と比べて、パケットを待ち合わせる時間の大幅な削減が可能となる。

- (b) の場合、マスター間のバイセクションバンド幅は大きくなったが、ミラーノードとの距離が大きくなった。
- (c) の場合は、マスターノード間の通信の効率が最も良いが、マスターノード、ミラーノード間の距離が 4 であり、マージ通信、コピー通信の経路が、最大で 4 つも重なってしまう。そのため、マスターノードのルータにおいて、ミラーノードのパケットを待ち合わせる時間が長くなると考えられる。
- (d) の場合、(b) に比べて、マスター、ミラー間の経路の重なりが多く、通信衝突が増し、性能が低下すると考えられる。マージ通信、コピー通信の経路の重なりは最大は 8 である。
- (e) の場合は、マスターノード間のバンド幅が最小かつミラーノードとの距離も最大である。よって通信衝突が増加し、遅延が大きくなると考えられる。

5. 性能評価

5.1 評価環境

メニューコアプロセッサシミュレータの SimMc Version 1.0.0⁸⁾ に信頼性向上のための SmartCore システムを実装し、冗長実行時のアプリケーション配置を図 10 (a) から (e) の 5 つの場合について性能評価する。

ベンチマークとして、ランダム通信、all_to_all 通信、Nqueen、行列積 (Canon) を使用する。使用するノード数は 32 ノードとし、本来実行させるアプリケーションのためのノード数は、16 ノードとする。SimMc のパラメータを以下に列挙する。

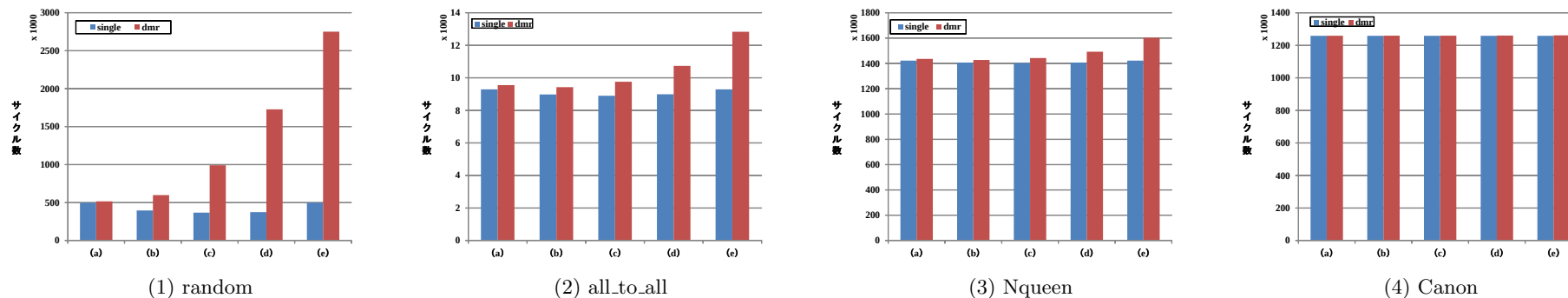


図 11 SmartCore システムによる各配置ごとの性能評価

- コアは MIPS ISA のシングルサイクルのプロセッサとする。
- オンチップネットワークのトポロジは 2 次元メッシュとする。
- DMA コントローラは DMA が発行されると次のサイクルからパケットの送信を開始する。1 サイクルに 1 フリット送信する。
- 1 パケットは 10 フリットを最長とし、転送量が多い場合は複数のパケットに分割して転送する。先頭の 2 フリットにヘッダとアドレス、続くフリットにデータが格納される。1 フリットに格納されるデータは 4 バイトとする。
- ルータは 1 サイクルに 1 フリット伝送する。ルーティングは XY 次元順ルーティング、フロー制御はクレジット方式を採用する。

また、本稿では、通常の実行と SmartCore システムを用いた性能低下を調べるために、SmartCore システムを用いずに性能を評価したもの (single), SmartCore システムを利用して性能評価したもの (dmr) の 2 つの条件で評価する。

5.2 多重実行による性能評価

図 11 に、上記に述べた条件の性能を評価した結果を示す。縦軸に実行に要したサイクル数、横軸は、図 10 に対応した (a) から (e) の配置である。配置によるネットワーク性能の考察のために、ランダム通信、all_to_all 通信を用いた。ランダム通信において、冗長実行を行わない場合では、(c) が実行サイクル数は最小である。しかし、多重実行した際には、(a) の性能低下が最も少なく、サイクル数の増加は 3.7 % である。また、最大で (e) のサイクル数の増加が 451 % である。all_to_all 通信の場合も同様に、マスターとミラーの位置が離れていくにつれて、通常の実行と比較した性能低下度合いが大きくなった。コピー通信、

マージ通信における経路上で通信衝突が増加し、マスターノードにてミラーノードから送られてくるパケットを待ち合わせるのに要する時間が増えることが性能低下の原因であると考えられる。

Nqueen と Canon の性能を比較すると、マスターノードとミラーノードの距離が遠くなるにつれて Nqueen の方が顕著な性能低下が見られた。Nqueen の場合、1 つのノードにパケットの送受信が集中する。対して Canon は全てのコアが同程度の通信量である。このことから、通信が一部に集中するアプリケーションは待ち合わせによる性能低下が大きいと考えられる。

6. 関連研究

マルチコアプロセッサの空間的冗長性を利用し、高性能化とディペンダビリティ向上を達成する方式に Slipstream プロセッサ⁹⁾がある。通常の命令列を 1 つのコアで動作させ、同時に、同様の挙動を示す短い命令列を異なるコアで動作させることで、高性能化とディペンダビリティの向上を達成する。この方式は、コアの同期のために低レイテンシの特別なコア間通信の仕組みを必要とするため、我々が想定している大規模なメニーコアアーキテクチャにおける利用は困難である。

マルチコアプロセッサの空間的冗長性を利用し、エラー検出および回復をおこなう手法に Loose Lock-stepped システム^{10),11)}がある。この手法では、コア、キャッシュおよび主記憶をいくつかのグループに分割し、冗長実行させる。I/O のレベルでエラー検出する。主記憶を冗長実行させる点、比較をプロセッサのチップ外で行っている点が SmartCore システ

ムとは異なる。

複数の CPU に同一の動作をさせることで故障を検出する技術として、PowerPC 750GX に採用されている Lockstep¹²⁾ がある。チェック用のプロセッサを追加し、プロセッサの入出力のレベルでエラー検出する。2つのプロセッサは、システムの起動時からクロックレベルで完全に同期する。SmartCore システムとは、冗長実行させるコアを柔軟に変更することが出来る点で異なっている。

7. まとめと今後の課題

我々は、メニーコアプロセッサによる空間冗長性を利用することで信頼性の向上を目指す SmartCore システムを提案している。SmartCore システムは、マスターノードとミラーノードにより冗長実行を行い、パケットの比較、エラー検出をおこなう。という3つの機能により信頼性の向上を達成する。マスターからコピーされたパケットをミラーノードへ送る通信をコピー通信、マスターノード間の通信をノード間通信、ミラーノードがマスターノードへパケット比較のために送る通信をマージ通信とする。マスター、ミラーの配置パターンによっては、通信の経路次第では、デッドロックの可能性があるが、コピー通信、ノード間通信、マージ通信の仮想チャネルを区別することで、各通信への切り替わりの際の依存関係を取り除き、デッドロックを回避する。

本稿では、SmartCore システムを冗長実行時の配置において、マスターノードとミラーノードの間の通信の距離が異なる場合について考察した。

性能評価により、ノード間通信とマスター、ミラー間の通信の関係から、マスターとミラーの距離が遠くなるほど性能低下が大きいことが分かった。

今後の課題として次のことが挙げられる。

- エラーのモデルを設定し実際にエラー検出を行い信頼性を計測する
- エラーの効率的な回復手法を検討する。
- マージ通信、コピー通信の追加による性能低下を緩和する通信手法を検討する。

謝辞 本研究の一部は、科学技術振興機構・戦略的創造研究推進事業 (CREST) 「アーキテクチャと形式的検証の協調による超ディペンダブル VLSI」の支援による。ルータの基礎から実装に至るまで丁寧に指導していただきました電気通信大学の吉永努教授、東京大学の松谷宏紀博士に感謝いたします。

参考文献

- 1) 佐藤寿倫, 舟木敏正: マルチコアプロセッサのための電力・性能間トレードオフを考慮したディペンダビリティ選択法, 情報処理学会論文誌, Vol.49, No.6, pp.2005–2015 (2008).
- 2) 佐藤寿倫, 舟木敏正: 性能・消費電力・信頼性の間のトレードオフを考慮出来るマルチ・クラスタ型コア・プロセッサ, 電子情報通信学会技術研究報告. CPSY, コンピュータシステム, Vol.107, No.276, pp.39–44 (2007).
- 3) 佐藤真平, 植原 昂, 三好健文, 吉瀬謙二: SmartCore システムによるメニーコアプロセッサの信頼性向上手法, 情報処理学会研究報告. EMB, 組込みシステム, Vol.2010, No.13, pp.1–6 (2010-01-21).
- 4) 入江英嗣, 杉本 健, 五島正裕, 坂井修一: 動的タイミング・エラー検出のための「書き込み保証バッファ」の評価, 情報処理学会研究報告, 2007-ARC-173, Vol.2007, No.55, pp.73–78 (2007).
- 5) 吉瀬 謙二, 植原 昂, 佐藤真平: メニーコアプロセッサのディペンダビリティ向上と高性能化を目指す SmartCore システム, 情報処理学会研究報告 2008-ARC-180, Vol.2008, No.101, pp.49–52 (2008).
- 6) 佐藤真平, 植原 昂, 吉瀬謙二: メニーコアプロセッサのオンチップネットワーク性能を向上させる SmartCore システム, 先進的計算基盤システムシンポジウム SACSIS2009 論文集, pp.27–35 (2009).
- 7) Dally, W.J. and Aoki, H.: Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels, *IEEE Trans. Parallel Distrib. Syst.*, Vol.4, pp. 466–475 (1993).
- 8) Uehara, K., Sato, S., Miyoshi, T. and Kise, K.: A Study of an Infrastructure for Research and Development of Many-Core Processors, *Workshop on Ultra Performance and Dependable Acceleration Systems held in conjunction with PDCAT'09*, pp.414–419 (2009).
- 9) Sundaramoorthy, K., Purser, Z. and Rotenberg, E.: Slipstream processors: Improving both Performance and Fault Tolerance, *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, Vol.35, No.11, pp.257–268 (2000).
- 10) Aggarwal, N., Ranganathan, P., Jouppi, N.P. and Smith, J.E.: Configurable Isolation: Building High Availability Systems with Commodity Multi-core Processors, *ISCA '07: Proceedings of the 34th annual International Symposium on Computer Architecture*, pp.470–481 (2007).
- 11) Aggarwal, N., Jouppi, N.P., Smith, J.E., Ranganathan, P. and Saluja, K.K.: Implementing High Availability Memory with a Duplication Cache, *Proceedings of the 41st Annual International Symposium on Microarchitecture (MICRO-41)* (2008).

12) PowerPC 750GX Lockstep Facility: IBM Application Note (2008).