

自己タイミング型パイプラインシステムのマクロシミュレーションモデル

三 宮 秀 次[†] 大 森 洋 一[†] 岩 田 誠[†]

隣接ステージ間でのみ同期することで配線や電力消費を局所化する自己タイミング型パイプライン (STP) は、負荷に応じてパケットの転送時間を調整し、一時的な過負荷を受容できる。このとき、動的な負荷変動に応じて、性能が変化する。本稿では、実際のステージ長にばらつきのある STP をマクロにモデル化し、仮想的なパケット間の距離に着目することで、性能見積りを簡略化するマクロシミュレーションモデルの有効性を確認するため、シミュレータを実装し、性能見積りを実プロセッサと比較した。その結果、実プロセッサの実効性能を概観でき、加えてシミュレーション時間を短縮化できることから、高速かつ正確な性能見積りが可能と判った。

A Macro-Simulation Model for Self-Timed Pipeline Systems

SHUJI SANNOMIYA,[†] YOICHI OMORI,[†] and MAKOTO IWATA[†]

This paper presents Macro-Simulation model of STP (Self-Timed Pipeline) systems. STP provides the robustness for temporal overload by adjusting the transfer time of packets. On the other hand, the variation of packet transfer time under the overload fluctuates the system performance, therefore in practical system design, the performance estimation model, which can treat the dynamic load fluctuation and execute fast simulation, is essential. In order to reduce computational cost, the proposed model introduces a virtual packet average speed, and applied it to every packet. A simulator based on the proposed model obtains an overview for an effective performance of a real processor, with fast simulation time.

1. ま え が き

隣接ステージ間でのみ同期することで、配線および電力消費を局所化する自己タイミング型パイプライン (STP) は、耐遅延性の高さなどの信号処理に適した特性を活用したメディアプロセッサ DDMP¹⁾ などで採用され、高い有効性が確認されている。

STP は、過負荷になるとパケットの転送が負荷に応じて遅らされ、ある程度の過負荷にもしなやかに適応することが知られている。この特性を活用して、処理中の負荷変動が大きい動画処理や IP ルーティング等のアプリケーションにおいて、設計の目標値を理論的な最大値よりも低い値へ最適化することで、よりハードウェア利用効率の良いシステムを実現できる。この場合、ステージ数等の回路設計パラメータとハードウェアコストとのトレードオフ²⁾ の決定に、負荷変動に応じたシステム性能の見積りに基づく設計の試行錯誤が必要であり、その期間短縮が課題となる。

STP は全体として非同期に動作するので、論理設計やタイミング設計に同期式设计とは異なる手法が必要である⁷⁾。具体的には、非同期式タイミングモデルの下で論理設計を行い、論理をパイプラインステージに分割する。その後、配置配線され、チップ実現に至る。ここで、論理の分割では、最適なステージの割り当ての試行錯誤の抑制のため、

ステージの遅延を鑑みたシステム性能の見積りが要となる。

これまでに、非同期システムの性能見積りを離散事象シミュレーションと捉える、確率的なモデル³⁾ やマルコフ鎖⁴⁾ 等が研究されているが、これらは設計の最適化において重要な、具体的なアプリケーションにおける時間的な負荷変動パターンに対して、厳密な評価ができない。このため、パケット転送時間を考慮したシミュレーションが用いられているが⁵⁾⁶⁾、上記の設計手順がかなり進行しないと、こうした回路レベルの遅延を考慮するのは難しい。

本稿では、設計の初期段階において、性能見積りを簡略化するために、STP をパケットの移動する平均的な速度に注目して捉えることで、遅延をステージ毎に評価せずとも、性能見積りに反映できるマクロシミュレーションモデルを提案し、シミュレータを実装して評価した。

2. STP システムの性能見積り

2.1 リング型 STP

STP の基本的な構成は、図 1 に示すとおり、データラッチ、処理回路、および C 素子と呼ばれる転送制御回路により各パイプラインステージが構成され、パケットは、転送の要求を意味する send 信号および転送の許可を意味する ack 信号を用いた、C 素子間のハンドシェイクによりステージ間を転送される。

アルゴリズム中の依存関係の連鎖を直接回路上で実現するためには、図 1 の構成を環状に接続したリング型で STP システムを構成する。例として、図 2 に DDMP のプロッ

[†] 高知工科大学

Kochi University of Technology

ク図を示す。

DDMP は、データ駆動型処理を採用しており、パケットの待ち合わせ、発火、演算、および次命令フェッチの機能を STP に配置し、プロセッサエレメント (PE) を構成する。さらに、PE 間を、2 つのパケットの流れを調停し合流させるマージステージ (M)、および 1 つのパケットの流れを選択し分流するブランチステージ (B) で構成する多段パケットルータで結合し、リング型 STP による CMP (Chip Multi-Processor) を形成する。以降、本稿では、リング型 STP を対象とし、リング型 STP を単に STP と呼ぶ。

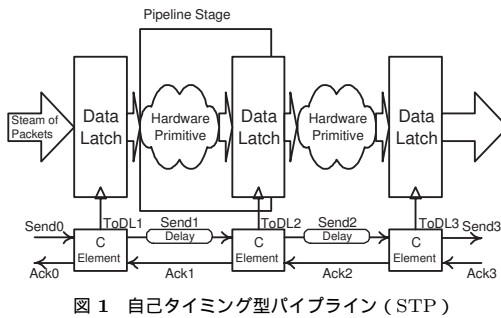


図 1 自己タイミング型パイプライン (STP)

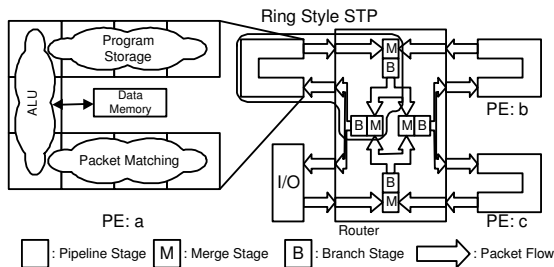


図 2 DDMP:リング型 STP システムの例

2.2 STP の性能特性

STP の性能は、パケットが周回する時間であるターンアラウンドタイム [秒]、および単位時間あたりの出力パケット数であるスループット [パケット/秒] で表される。これらはパケット毎のハンドシェイクに要する時間より求められる。以降、パケット流量は定常的に安定しているものと仮定し、性能特性を明らかにする。

ここで、send 信号の伝達に要する時間 (遅延) を T_f [秒]、ack 信号の遅延を T_r [秒] とする。STP では、send 信号の到達に先立ち、ack 信号が到着していれば、パケット転送時間において、 T_r は T_f に隠蔽される。つまり、ステージ数に対するパケット数の比 (R_f) が十分に低く、図 3 に示すように、平均的なパケット間隔が時間的に ($T_f + T_r$) 以上になる場合 (図 3(a))、後続ステージからの send 信号の到達に先立ち、先行ステージから ack 信号が到着するため、各パケットは常に $\frac{1}{T_f}$ [ステージ/秒] の速度で前進する。さらに、後続ステージには直ちに ack 信号が送信され

るため、時間的な間隔は維持される (図 3(b))。この状態では、パケット速度は最大となることから、以降、この状態をパケット最速状態と呼ぶ。

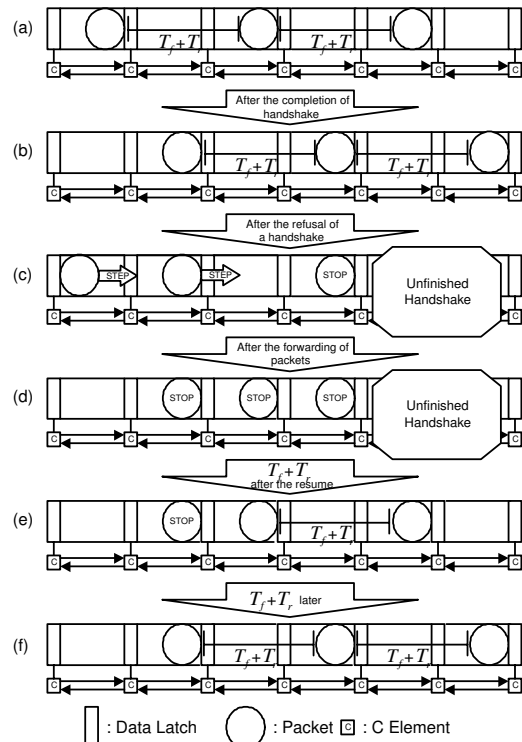


図 3 STP のパケット転送

また、STP では、例えばマージステージでの調停で、ハンドシェイクが延期され、パケットが停止する場合でも、後続パケットは可能な限り前進できる (図 3(c))。ハンドシェイクの延期や、 R_f の増加により、2 つのパケットが時間的に ($T_f + T_r$) 以下に詰められた場合、後続パケットは、send 信号の到達後も、ack 信号の到着を待つ (図 3(d))。これを衝突と呼ぶ。衝突は、パケットを減速させ、STP の性能を低下させる。衝突が発生する状態をパケット減速状態と呼ぶ。次に、エラスティック能力を説明する。

STP では、ハンドシェイクが完了すると同時にパケットは移動を再開し、1 ステージの空きがあれば処理が継続される。ハンドシェイクが完了すると、先頭のパケットが移動を始めてから ($T_f + T_r$) 後に、2 目目のパケットが転送されるため (図 3(e))、パケットは ($T_f + T_r$) の間隔で隔てられ、パケット最速状態へ復帰する (図 3(f))。

2.3 実プロセッサに関する検討

衝突の影響は、実プロセッサで 3 段階の不連続な性能低下として現れることが知られている⁸⁾。図 1 に示すハンドシェイクを逐一模擬する簡易な性能見積りモデル (以降、ナイーブモデルと呼ぶ) を導出し、現行の DDMP チップと比較したシミュレーション結果を図 4 に示す。グラフの横軸は R_f であり、縦軸は、 R_f を変化させた場合のパケッ

ト速度の最大値を 1 として正規化したものである。

DDMP チップで 3 段階のグラフであるのにナイーブシミュレーションで 2 段階になった理由は、両者の最も大きな違いである回路実装における $(T_f + T_r)$ の不均質化が主因である。設計ツールや製造環境に委ねられる回路実現工程において、設計レベルで定めた遅延を保つのは困難であり、実用的な性能見積りモデルでは、ステージ毎に T_f または T_r の異なる STP を扱う必要がある。

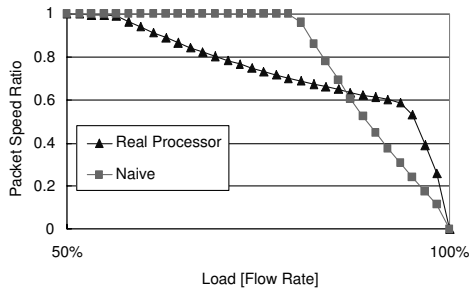


図 4 実プロセッサとの性能見積りの比較

実際、各ステージの $(T_f + T_r)$ を変化させた STP に対して、ナイーブモデルを用いてシミュレーションを行った結果、 $(T_f + T_r)$ の偏差によらず、図 5 に示すような 3 段階の不連続なフェーズからなるグラフが得られた。

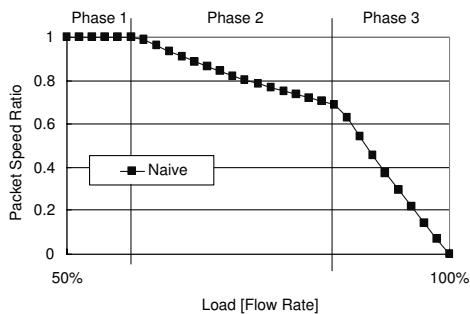


図 5 不均質なステージを持つ STP

図 5 では、性能低下の特性に基づき、パケット最速状態である Phase 1、パケット減速状態である Phase 2 および Phase 3 に分類している。

実システムのスループットは、1 パケットあたりの性能にパケット数を掛けた、図 6 に示すような曲線を示す。したがって、STP では、 R_f を Phase 2 に収めることがハードウェア設計を最適化する目標となる。この時、同期システムと異なり、Phase 3 の分だけ余裕を持たせることができる。本稿では、性能評価の指標に、ターンアラウンドタイムではなくスループットを用いた。

3. マクロシミュレーションモデル

本章では、STP の不連続な性能低下の特性を R_f とパケットの速度の関係で説明するマクロシミュレーションモ

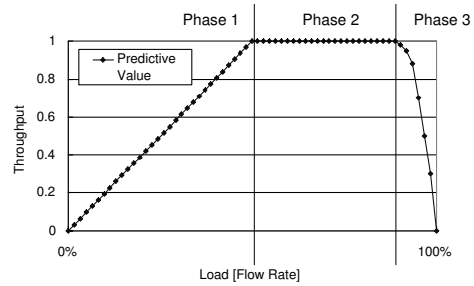


図 6 STP システムのスループット

デルを説明する。STP は、パケット減速状態ではパケットの衝突がパケットを減速させ、性能を低下させる。一方、エラスティック能力により衝突の影響はパイプライン全体へ波及する。これらの事実に基づき、マクロシミュレーションモデルでは、パケットの平均速度 (V_p [ステージ/秒]) を導入し、すべてのパケットに一律に適用することで、STP の状態管理を簡略化する。

ここで、パケット総数、ステージ数、全ステージの T_f の合計、全ステージの T_r の合計、および最大の $(T_f + T_r)$ を持つ最長ステージ長を、それぞれ、 P_{total} 、 pl 、 $\sum T_f$ 、 $\sum T_r$ 、および $\max(T_f + T_r)$ とおく。

次節より、各フェーズにおけるパケットの時間的な間隔と移動速度の関係から、不連続点の境界条件および V_p を求める。

3.1 Phase 1

R_f が十分に低い場合、STP のエラスティック能力により、最長ステージを通過したパケットの間隔は、 $\max(T_f + T_r)$ 以上に保たれ、パケットの転送時間において、 T_r は隠蔽され、パケットは $\sum T_f$ で周回する。そのため、 V_p を最大に保つには、パケットの間隔が $\max(T_f + T_r)$ 以上であるよう、 P_{total} は、

$$\sum T_f \leq \max(T_f + T_r) \times P_{total} \quad (1)$$

を満足する必要がある。式 (1) を満足する P_{total} の最大値を P_{1-2} とすると、

$$P_{1-2} = \lfloor \frac{\sum T_f}{\max(T_f + T_r)} \rfloor \quad (2)$$

である。図 7 に Phase 1 のパケットの模式図を示す。

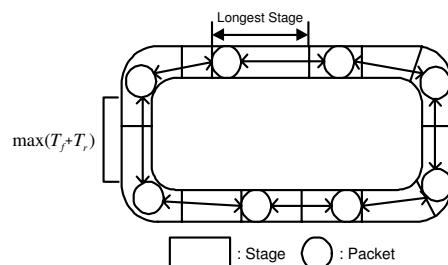


図 7 パケットの模式図 (Phase 1)

Phase 1 では、 V_p は定数 $\frac{pl}{\sum T_f}$ となる。

3.2 Phase 2

STP のエラスティック能力により、最長ステージを通過したパケットは、 $\max(T_f + T_r)$ 以上に隔てられるため、 P_{total} の増加により、式(2)が破綻したとき、最長ステージの通過に際し、最初のパケット衝突が発生し、影響は後続パケットに伝播する。この状態では、パケットは最長ステージの直前で、先行パケットの通過時間を、最大 $\max(T_f + T_r)$ の間待つことになり、パケットが周回する時間は、式(2)を超過するパケット1つにつき、最大で $\max(T_f + T_r)$ 増加する。図8に衝突の影響が伝播する模様を示す。

Phase 2 で衝突が伝播する範囲は、各ステージのエラスティック能力で決まる。最長ステージを S_1 とし、後続ステージを順に S_2, S_3, \dots, S_{pl} とおくと、ステージ S_i におけるエラスティック能力は、 $\max(T_f + T_r) - (T_{fi} + T_{ri})$ となる。ここで、 $(T_{fi} + T_{ri})$ はステージ i における $(T_f + T_r)$ である。このことから、衝突に影響されるステージ数 n は、

$$P_{over} \times \max(T_f + T_r) \leq \sum_{i=1}^n \max(T_f + T_r) - (T_{fi} + T_{ri}) \quad (3)$$

を満たす最小数となる。ここで、 $P_{total} - P_{1-2}$ を P_{over} とする。

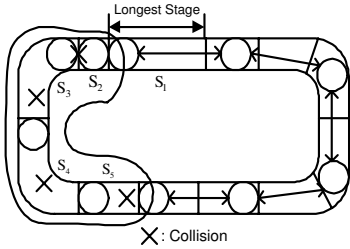


図8 パケットの模式図 (Phase 2)

Phase 2 と Phase 3 の境界条件は、式(3)より、

$$P_{over} \leq \frac{\sum_{i=1}^n \{\max(T_f + T_r) - (T_{fi} + T_{ri})\}}{\max(T_f + T_r)} \quad (4)$$

である。定義より、左辺は、

$$P_{over} = P_{total} - \lfloor \frac{\sum T_f}{\max(T_f + T_r)} \rfloor \quad (5)$$

となる。右辺は衝突が1周する $n = pl$ のとき最大で、

$$\begin{aligned} & \frac{\sum_{i=1}^{pl} \{\max(T_f + T_r) - (T_{fi} + T_{ri})\}}{\max(T_f + T_r)} \\ &= \frac{\sum_{i=1}^{pl} \max(T_f + T_r)}{\max(T_f + T_r)} - \frac{\sum_{i=1}^{pl} T_{fi} + \sum_{i=1}^{pl} T_{ri}}{\max(T_f + T_r)} \\ &= pl - \frac{\sum T_f + \sum T_r}{\max(T_f + T_r)} \end{aligned} \quad (6)$$

となる。式(4)に式(5)と式(6)を代入すると、

$$\begin{aligned} P_{total} - \lfloor \frac{\sum T_f}{\max(T_f + T_r)} \rfloor &\leq pl - \frac{\sum T_f + \sum T_r}{\max(T_f + T_r)} \\ \lfloor \frac{\sum T_r}{\max(T_f + T_r)} \rfloor &\leq pl - P_{total} \end{aligned} \quad (7)$$

となる。式(7)を満足する P_{total} の最大値を P_{2-3} とすると、

$$P_{2-3} = pl - \lfloor \frac{\sum T_r}{\max(T_f + T_r)} \rfloor \quad (8)$$

である。

Phase 2 における V_p は、パケット増加に対する速度の変化が定数 $\max(T_f + T_r)$ であるから、 P_{total} に関する一次式となる。

3.3 Phase 3

さらなるパケット数の増加により、式(8)が破綻した場合、図9に示すように、もはやエラスティック能力に余裕がないので衝突の伝播は1周し、最長ステージでの新たな衝突にまで波及する。即ち、一旦発生した衝突はSTPを周回し続け、すべてのパケットがステージを転送されるたびに衝突する。

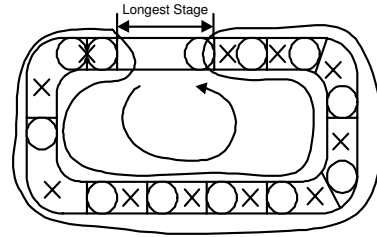


図9 パケットの模式図 (Phase 3)

つまり、Phase 3 では、各パケットが前進するかどうかは、ある時刻 t における P_{total} ($P_{total}(t)$) に比例して発生する衝突に依存する。この場合、衝突の発生確率 $p(t)$ は、

$$p(t) = \frac{P_{total}(t)}{pl} \quad (9)$$

となる。パケットは衝突した時に停止し、衝突していない時に1ステージ進むので、 t における V_p の変化率 ($\frac{dV_p}{dt}$) は、

$$\begin{aligned} \frac{dV_p}{dt} &= 0 \times p(t) + 1 \times (1 - p(t)) \\ &= 1 - p(t) \end{aligned} \quad (10)$$

となる。 $\frac{dP_{total}(t)}{dt} = \text{一定}$ 、即ち R_f に時間的な変化は無い範囲では、

$$\begin{aligned} \int \frac{dV_p}{dt} dt &= \int 1 - p(t) dt \\ &= \frac{1}{C} \int 1 - \frac{P_{total}(t)}{pl} \frac{dP_{total}(t)}{dt} dt \\ &= -\frac{1}{2pl} (P_{total}(t))^2 + \frac{1}{C} P_{total}(t) + Const. \end{aligned} \quad (11)$$

が得られる。ただし、定数 $C = \frac{dP_{total}(t)}{dt}$ である。

式(11)は、Phase 3 における V_p は P_{total} に関する2次式で求められることを示している。

以上より、STP の性能を、 V_p を用いて統一的に扱うマクロシミュレーションモデルを導出できた。

4. 評価

4.1 評価手法と環境

前章で明らかにしたマクロシミュレーションモデルのパラメータの決定に必要な情報により、実システムの実効性能を高速に概観できることを示す。DDMP を対象に、まず既存のシミュレータとの比較を行い、次に実プロセッサとの比較を行った。

DDMP チップの性能値の計測には、評価ボードを用いた。評価ボードは PCI バス経由でホスト PC と接続され、バッチ処理により実行される。評価ボードでは、DDMP チップは FPGA チップに接続されており、FPGA チップ内で、出力パケットに性能値の算出のため出力時刻のタイムスタンプを付与する。

現行の DDMP は、10 個の PE を持ち、PE はステージ数と命令セットにより 3 つに分類され、ここでは、それぞれ $PE : x$ 、 $PE : y$ 、および $PE : z$ とする。各 PE は、内部メモリアクセス命令等の共通の命令以外に、 $PE : x$ は外部メモリアクセス、 $PE : y$ はタグ処理および算術演算、さらに $PE : z$ は論理演算のための命令セットを持つ。本稿では、実アプリケーションの実行に不可欠な命令セットを単体で備える $PE : y$ を対象とする。

マクロシミュレーションモデルに基づいたシミュレータは、Java を用いて新たに実装した。本シミュレータは、プログラム、入力パケット、および pl 等の PE 情報を入力にとり、タイムスタンプを付与された、処理結果のパケットを出力する。シミュレーションでは、シミュレーションサイクル毎に算出される R_f に対する V_p に基づき、パケットの位置を更新し、プログラムに基づいて演算を適用する。

4.2 マクロシミュレーションモデルのパラメータ調整

マクロシミュレーションモデルのパラメータは、本来ならば、設計の試行で定まる遅延を基に調整する。ところが、既に DDMP は商用プロセッサとして 1 チップ実装されており、 T_f 、 T_r および R_f に関して正確な値を得ることは困難である。本稿では、DDMP チップから、パラメータ調整に必要な情報を採取した。

パラメータ調整には、 pl 、 P_{1-2} 、 P_{2-3} 、 $\sum_{T_f}^{pl}$ 、および P_{total} に関する 1 次と 2 次の近似式が必要である。評価に先立って、簡単なプログラムを評価ボードで実行し、必要なパラメータを推定した。

pl は、リング状のパイプラインにおいて、全ステージがパケットに占められるとデッドロックが発生することを利用して、パケットのコピー操作によりパイプラインを充足させて推定した。また、 P_{1-2} と P_{2-3} は、定常的な R_f を発生するプログラムによりスループットをグラフにプロットし、推定した。 $PE : y$ の場合、 pl は 42 段、スループットをプロットしたグラフから変曲点のパケット数を読み取り、 $P_{1-2} = 31$ 、 $P_{2-3} = 39$ を得る。

次に、 $\sum_{T_f}^{pl}$ である。 $\sum T_f$ は、式 (1) より、 $\sum T_f = P_{1-2} \times \max(T_f + T_r)$ である。式 (4) において、衝突が 1 周する $n = pl$ となる、Phase 2 と 3 の境界では、 P_{over} は最大値をとり、両辺を pl で割ると、

$$\frac{P_{over}}{pl} = \frac{\max(T_f + T_r) - (\overline{T_f + T_r})}{\max(T_f + T_r)} \quad (12)$$

を得る。ここで、 $(\overline{T_f + T_r})$ は平均の $(T_f + T_r)$ である。 $pl = 42$ なので、式 (12) は、 $\frac{P_{over}}{42} = \frac{\max(T_f + T_r) - (\overline{T_f + T_r})}{\max(T_f + T_r)}$

となり、 $\max(T_f + T_r) = 42$ より、 $\sum T_f = 31 \times 42$ を得る。故に、 $\frac{pl}{\sum T_f} = \frac{42}{31 \times 42} = \frac{1}{31}$ と推定される。

Phase 2 および 3 における、それぞれの近似式は、定常的な R_f を変化させた場合のスループットの数値解析により求めた。

4.3 既存シミュレータとの比較

DDMP 評価ボードに付属する既存のシステム設計支援ツールである DADT (DDMP Application Designer's Toolkit) に含まれるシミュレータ (以降、単に既存シミュレータ) との比較を行った。

2.3 節と同じく、 R_f に対するパケットの速度の変化をシミュレーションした結果と、実プロセッサによる実行結果を図 10 に示す。実プロセッサの実効性能では、Phase 1 内であってもパケット速度は低下し、数%の誤差がある。これは、現行のチップ製造技術がもたらす不均質な STP における、パケットの偏りが主因であると考えられる。そのような偏りの性能見積りへの反映は、半導体素子のレベルでの模擬を必要とし、設計の初期段階では困難である。そのため、マクロな性能見積りでは、この誤差の分だけ、設計のマージンをとる必要がある。このことから、結果のグラフでは、パケットの速度を $P_{total} = P_{1-2}$ における V_p を 1 として正規化している。この結果、 $PE : y$ に関しては、マクロシミュレーションモデルは最大で、約 3% の設計マージンが必要と判った。

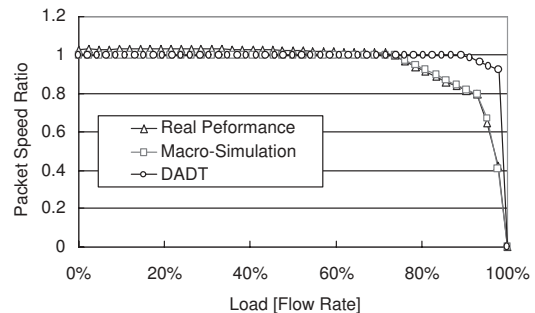


図 10 実プロセッサとの比較

一方、既存シミュレータとの比較では、より正確に R_f や衝突の影響が見積もれることが判った。結果から、既存シミュレータは、全ステージ長が均一のナイーブモデルを用いていると考えられる。ナイーブモデルに基づくシミュレーションでは、シミュレーションサイクル毎に、すべての C 素子およびパケットの状態を管理する必要があり、計算量の指標となるメモリアクセス数は、シミュレーションサイクル毎に $P_{total} + pl$ となるため、計算量は $O(P_{total} + pl)$ となる。一方、マクロシミュレーションモデルに基づくシミュレーションでは、メモリアクセス数は pl に依存しないため、計算量は $O(P_{total})$ まで削減できる。離散ウェーブレット変換を対象として、シミュレーション時間を測定

した結果、約 2 倍の高速化を達成した。

4.4 実アプリケーションへの適用

続いて、マクロシミュレーションモデルと実プロセッサの比較を行い、パラメータ調整したマクロシミュレーションモデルが、動的に負荷が変動する実システムの実効性能の見積りに適用できることを示す。

DDMP では、STP に対するパケットの入出力以外に、パケットの待ち合わせや廃棄が動的な R_f を減少させるが、それらのステージ数や配置は、設計の後半でないとは特定できないので、本シミュレータでは、設計の手戻りを回避するため、すべての機能ロジックを STP の出口で評価することで、 R_f の減少を最大まで遅らせており、 R_f を大きめにみつもることで、STP の最終的な性能が見積もり値を下回らないようにした。本構成では、最大で 1 パケット分の悲観的な見積りを招く。

本稿では、パケットの待ち合わせと廃棄のあるラプラシアン 4 近傍フィルタ (4 近傍フィルタ) と、どちらも無い、画像を縦横 2 倍に伸張するフィルタ (伸張フィルタ) を用いた。4 近傍フィルタを、 R_f の変動が極端に性能が変化する Phase 3 に達するよう、定常的な R_f を発生するプログラムと並列実行し、定常的な負荷が安定する時間を排除するため、出力パケット数を出力の開始から終了までの時間で割り、最大値を 1 とするスループット比で性能を測定した結果を、図 11 に示す。結果では、実際に結果を 1 パケット分読み替えた Macro-Simulation' は、最大で約 2% の誤差となり、 $PE : y$ の設計マージンに収まっている。

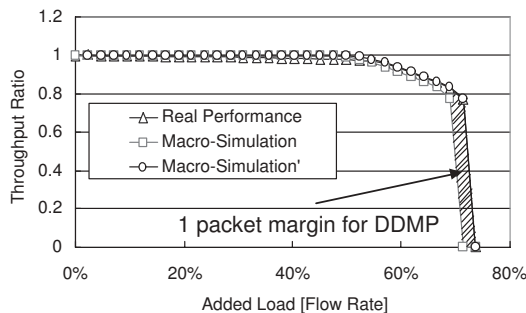


図 11 実システムの性能見積り (4 近傍フィルタ)

同様に測定した伸張フィルタの結果を図 12 に示す。誤差は、最大で約 2% となり、設計マージンに収まっている。また、1 パケット分だけ早くデッドロックを検出しており、1 パケット分だけ安全側に見積もれている。これまでのことから、マクロシミュレーションモデルは、高速かつ正確な実システムの性能見積りが可能であることを確認した。

5. む す び

本稿では、パケット流量とスループットの関係から、STP をパケットの移動速度でモデル化し、マクロシミュレーションモデルを提案した。マクロシミュレーションモデルでは、

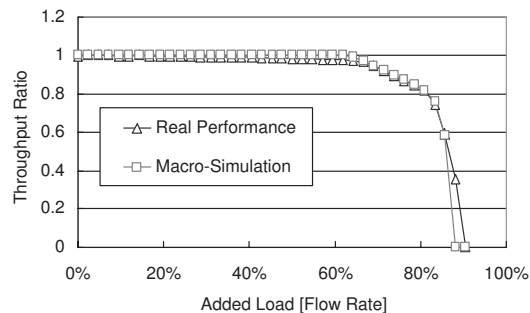


図 12 実システムの性能見積り (伸張フィルタ)

仮想的な平均速度を導入し、実プロセッサにおける性能の変化を统一的に扱うことで、パケットの管理を単純化した。提案モデルに基づくシミュレーションでは、既存のシミュレータと比較してシミュレーション時間を約 $\frac{1}{2}$ まで短縮でき、また、実プロセッサと比較して、負荷に応じた STP の実効性能を、現行のチップ製造技術を想定した設計に必須となる、最大 1 パケット分のマージン内で予測できることから、設計の初期段階で必須となる高速かつ正確な性能見積りができることを確認した。

参 考 文 献

- 1) H. Terada, S. Miyata, and M. Iwata, "DDMP's: Self-timed Super-pipelined Data-driven Multimedia Processors," Proceedings of the IEEE, 87(2), pp. 282-296, 1999.
- 2) T. E. Williams, "Latency and Throughput Trade-offs in Self-Timed Independent Pipelines and Rings," Technical Report No. CSL-TR-90-431, Stanford University, 1990.
- 3) S. Chakraborty and R. Angrish, "Probabilistic Timing Analysis of Asynchronous Systems with Moments of Delays," in Proc. of Int. Symp. on Advanced Research in Asynchronous Circuits and Systems, pp.99-108, 2002.
- 4) A. Xie and P. A. Beerel, "Accelerating Markovian Analysis of Asynchronous Systems using String-based State Compression," in Proc. of Int. Symp. on Advanced Research in Asynchronous Circuits and Systems, pp. 247-260, 1998.
- 5) J. C. Ebergen, S. Fairbanks, and I. E. Sutherland, "Predicting Performance of Micropipelines Using Charlie Diagrams," in Proc. of 4th Int. Symp. on Advanced Research in Asynchronous Circuits and Systems, pp. 238-246, 1998.
- 6) L. W. Nagel, "SPICE 2: A Computer Program to Simulate Semiconductor Circuits," Memorandum No. ERL-M520, Electronic Research Laboratory, College of Engineering, University of California, Berkeley, 1975.
- 7) C. J. Myers, 米田友洋 (訳), "非同期式回路の設計," 共立出版社, 東京, 2003.
- 8) 岡本俊弥, "データ駆動型メディアプロセッサ," 情報処理学会誌, 39 巻, 3 号, pp.208 - 214, March 1998.