10-IS05

計算機シミュレータ BSIM, NSIM によるスーパーコンピュータの性能予測及び性能解析

井上　弘士（九州大学大学院システム情報科学研究院）

概要
本研究では，これまでに開発したスパコン性能予測環境 BSIM を対象とし，次世代スーパーコンピュータのターゲットアプリケーションの１つであるフラグメント分子軌道（FMO）法を用いた実行時間予測精度の改善を試みた．また，インターコネクト・シミュレータ NSIM の活用事例として，本研究グループで提案している直接網インターコネクトにおける通信衝突ペナルティ算出方法を対象とした詳細な性能解析を行った．

1. 研究の目的と意義

本研究では，九州大学，財団法人九州先端科学技術研究所，富士通株式会社が共同で開発したスパコン向け性能予測環境 BSIM[3]ならびにインターコネクト・シミュレータ NSIM[2]を対象とし，実用性向上に向けた改良や応用展開を行う．

BSIM（Base SIMulator）は, Multi-Processing Environment(MPE)を拡張して構築された仮想超並列実行環境である．高い抽象度で記述された MPI 相当のプログラムを入力とし，BSIM 内部に搭載した仮想タイマを適切に更新することで仮想的な大規模並列計算機システムでのプログラム実行を模擬する．一方，NSIM は，大規模な相互結合網を対象とした通信遅延時間の見積りが可能なインターコネクト・シミュレータである．離散事象シミュレーションによる並列実行をサポートしており，比較的高速な性能評価が可能である．このような計算機シミュレータの主な役割としては，以下の 2 点が挙げられる．

- 現時点で利用不可能な計算機における実効性能の予測
- 実機での性能計測では行えない，計算機内部の細かい挙動の解析

BSIM ならびに NSIM の実用性を示すためには，幅広い計算機を対象とした性能予測ならびに性能解析の精度と速度に関する検証が必要である．これまでに, 主に PC クラスタにおいてこれらの検証を行い，BSIM と NSIM が高い実用性を有していることを示してきた[2] [3]．本研究では，これらツールの更なる改善や適用範囲の拡大を目的とし，以下に示す 2 つのサブテーマに関する研究を実施している．

1.1 BSIM における実効性能予測精度の改善

アプリケーションプログラム開発には長い時間を要するのが一般的である．そのため，開発中あるいは将来開発が予定されている先端的な大規模並列計算機の利用を前提としてアプリケーションプログラムを開発するには，実行環境となる計算機（ターゲットマシン）が存在しない状態での最適化が余儀なくされる．その場合，開発したアプリケーションプログラムがターゲットマシン上でどの程度の性能を示すかという性能推定を，アプリケーション開発時点で利用可能な計算機（実機）を用いて短時間で精度良く行うことが必要不可欠となる．

この問題を解決するため，我々は，実機を用いないスパコン向け実効性能予測環境として BSIM を開発した．これまでに，いくつかのアプリケーションについて BSIM を用いた性能推定を行ってきた[3]．BSIM を用いた並列アプリケーションプログラムの性能推定では，実際の計算を行う代わりに，計算時間をモデル化し，それを数式として埋め込んだ**スケルトンコード**と呼ばれるソースコードを作成する．そして，このスケルトンコードを実機上で実行することで性能推定を行う．これまで BSIM で性能推定を行った評価対象のアプリケーションプログラムの多くでは，BSIM を

用いることで,高精度かつ高速な性能推定が可能であることが確認された．しかしながら，次世代スーパーコンピュータでのターゲットアプリケーションの 1 つであるフラグメント分子軌道 (FMO) 法[1]については推定精度が非常に悪という結果であった．これは，FMO 計算で行う多数の小規模電子状態計算部分の推定実行時間を精度よく与えることができなかったことが主な原因である．並列 FMO プログラムは，10 万並列を超える超並列実行を行うことを目指している．そのような超並列実行時に効率よく動作するコードを効率的な開発には，BSIM などの性能推定ツールを活用することが望ましい．そこで，BSIM を用いた超並列実行時の性能推定を高い精度で行うことができるようにするために,小規模電子状態の計算時間を精度よく推定する手法を確立することを目的とした.

このような性能推定技術を確立することで BSIM による実効性能予測精度が向上し，引いては，実機が存在しない時点からのアプリケーションプログラムの開発を支援することが可能になると考えられる.また,スケルトンコードでは計算の細かな部分を推定計算時間の埋め込みで抽象化するため,大規模並列プログラムの見通しが良くなる.そのため，大規模並列アプリケーションプログラムをトップダウン的に開発することに大きく役立つと考えられる.したがって，大規模並列プログラム開発用ツールとしての有効性を検証する意味でも,本研究で得られた知見は大きな意義を持つ.

1.2 NSIM を用いた通信衝突ペナルティ算出法の評価解析

大規模並列計算機では，そのコストや設置面積のため，クロスバ網のかわりに fat tree や Mesh/Torus 網を相互結合網として使用している．これらのトポロジでは複数の通信が 1 つのリンクを共有するため通信衝突が発生する．通信衝突は通信性能を十分に悪化させうる．そのため，通信衝突によって生じるペナルティを精度良く見積る方法の確立が重要となる．また，通信衝突を考慮したタスク配置最適化などを実施する場合には，この通信衝突ペナルティを短時間で見積らなければならない.

この問題を解決するため，本研究グループでは直接網における通信衝突のペナルティ算出法を提案している．一般に，メッセージはパケットに分解されて送付されるが，パケットが同時に同一リンクに到着した場合は何らかの優先度に基づき処理される．そのため，このようなパケット優先度を考慮した通信衝突ペナルティの見積りがより重要となる．本研究では，NSIM の活用事例として，提案している通信衝突ペナルティ計算法の精度評価と詳細な解析を行う．このように，スーパーコンピュータ向けインターコネクション・ネットワークの研究開発において NSIM を実際に利用することにより，フィードバックに基づく改善を図ると共に，活用事例を多く示すことにより広い普及を目指す.

参考文献

[1] K. Kitaura et al., “Fragment molecular orbital method: an approximate computational method for large molecules”, Chem.Phys.Lett. 313(1999)701-706

[2] H. Miwa, R. Susukita, H, Shibamura, T. Hirao, J. Maki, Y. Inadomi, K. Inoue, Y. Ajima, I. Miyoshi, T. Simizu and H. Ando, ″NSIM: Communication Simulator for Next Generation Extreme-scale Interconnection Networks,″ IPSJ SIG Technical Report, vol2010-HPC-125, NO.5, in 2010.

[3] R. Susukita et al.,” Performance Prediction of Large-scale Parallel System and Application using Macro-level Simulation”, the International Conference for High Performance Computing, Networking, Storage and Analysis (SC08), Nov. 2008.

2. 当拠点公募型共同研究として実施した意義

(1) 共同研究を実施した大学名
九州大学

(2) 共同研究分野
大規模情報システム関連研究分野

(3) 当公募型共同研究ならではという事項など
スーパーコンピュータの性能予測における精度向上に向けた取組みを実施できた．精度向上には実機の利用が必要不可欠であり，本共同研究を通したからこそ実施できたも

のである．また，未だ課題は残されているが，スーパーコンピュータを実際に利用している現場において，BSIM や NSIM といったスパコン開発用ツールの実用性検証の取組みを実施できた．

3. 研究成果の詳細

3.1 FMO 計算における BSIM 性能予測精度の改善

FMO 法は，たんぱく質や DNA，あるいは，糖鎖といった大規模生体分子に対する量子力学に基づいた第一原理電子状態計算を高速に行うために開発された計算手法である．巨大な分子を 20～40 原子程度の小規模なフラグメントに分割して，フラグメント（モノマー）とフラグメントペア（ダイマー）に対する小規模な電子状態計算を行うことで，巨大分子全体の電子状態（エネルギー，電荷分布など）を近似する．したがって，FMO 計算の主要部分は，多数の小規模電子状態計算である．この小規模電子状態計算の模擬コードを図 3.1.1 に示す．この模擬コードでは，電子反発積分，4 中心クーロン積分，3 中心クーロン積分，2 中心クーロン積分，1 電子積分（これらをまとめて「**分子積分**」と呼ぶ）を順番に計算した後，これらの結果を用いた繰り返し計算（SCF 計算）を calc_scf 関数内部で行う構造になっている．小規模電子状態計算の呼び出し１回あたり，4 中心クーロン積分，3 中心クーロン積分，2 中心クーロン積分は，それぞれ，10～30 回（図中の `nifc4c` の値），10～30 回（図中の `nifc3c` の値），モノマー数（図中の `nfrag` の値）回実行される．また，電子反発積分，1 電子積分の計算は，それぞれ，1 回ずつ行われる．小規模電子状態計算の実行時間を推定するためには，模擬コードに現れる各部分の計算時間を推定する方法を考える必要がある．1 電子積分計算は，0.1～1 秒程度と計算時間が短いうえに，2 中心クーロン積分計算と計算量が似通っているため，2 中心クーロン積分計算時間の推定方法と同じで構わない．そこで，本研究においては，電子反発積分，4 中心クーロン積分，3 中心クーロン積分，2 中心クーロン積分，SCF 計算の計算時間を推定する方法を考えることにした．

まず，電子反発積分，4 中心クーロン積分，3 中心クーロン積分，および，2 中心クーロン積分（まとめて「分子積分」と呼ぶ）の計算時間推定について考える．2 中心クーロン積分を除く分子積分の計算時間（t）は，計算する積分の個数（$n_{\mathrm{calcd\,integ}}$）に比例する一次式で近似できると考える．

```
calc_eri_first( eri_buf );
for ( i=0; i<nifc4c; i++ ) {
  calc_4c_coulomb( i, dU );
  U += dU;
}
for ( i=0; i<nifc3c; i++ ) {
  calc_3c_coulomb( i, dU );
  U += dU;
}
for ( i=0; i<nfrag; i++ ) {
  calc_2c_coulomb( i, dU );
  U += dU;
}
calc_oneint( S, H );
calc_scf( U, S, P, eri_buf );
```

図 3.1.1. FMO 計算における小規模電子状態計算の模擬コード

$$t = \alpha n_{\mathrm{calcd\,integ}} + \beta \qquad \text{（式 1）}$$

比例定数αは，積分１つあたりの平均計算時間に相当する値である．2 中心クーロン積分については，計算に利用する基底関数ペア数と計算する積分数の２つの量に比例する式で近似することにした．

$$t_{2C} = \alpha n_{\mathrm{calcd\,integ}} + \beta n_{\mathrm{PSpair}} + \gamma \qquad \text{（式 2）}$$

係数$\alpha, \beta\,(,\gamma)$は，ターゲットマシンに依存するが，今回の精度検証でのターゲットマシンである PRIMEQUEST の一部ノードを用いて小規模な積分計算を行い，積分数と計算に要した時間を測定して，その実行時間と計算した積分数との関係から，最小二乗法を用いて$t = \alpha n_{\mathrm{calcd\,integ}} + \beta$（式 1，$t_{2C} = \alpha n_{\mathrm{calcd\,integ}} + \beta n_{\mathrm{PSpair}} + \gamma$（式 2 の係数$\alpha, \beta\,(,\gamma)$を求めた．その結果の一部を表 3.1.1 に示す．求められた近似式の決定係数（R^2）は，すべて，0.99 を超えており，この近似式の精度が非常に

表 3.1.1. 全て s 型関数で構成された積分の計算時間推定式の係数

分子積分の種類	α	β	
電子反発積分 4 中心クーロン積分	1.34×10^{-7}	2.51×10^{-2}	
3 中心クーロン積分	1.24×10^{-7}	-2.05×10^{-3}	
	α	β	γ
2 中心クーロン積分	8.48×10^{-8}	1.93×10^{-7}	9.50×10^{-5}

よいことが分かった．

一方，計算する積分の個数については，簡単に推定することが困難である．それは，分子積分計算では，通常，計算量削減の観点からカットオフを行うためである（計算量が 1/10 以下になる）．カットオフ計算で行き残る，すなわち，計算する積分の個数は，計算に用いる基底関数の種類や，分子の形状，フラグメントのサイズ，あるいは，フラグメントの組成などの様々な要因により変化する可能性があるため，特別な計算をしないで入力データから得られる基底関数の数などのデータから，単純に推定することが困難である．このことは，以前に行った FMO 計算の計算時間推定で入力データの単純な解析結果だけを使った小規模電子状態計算部分の実行時間推定がうまくいかなかったことでも明らかである．そこで本研究では，各フラグメントのカットオフテーブルを計算し，その結果を用いて積分数を推定した．カットオフテーブルの計算結果には，先に述べた積分数と相関する可能性のある基底関数の種類やフラグメントの形状などの情報が含まれている．そのため，カットオフテーブルの計算結果を利用することで，計算時間推定の精度が向上すると考えた．また，利用するカットオフテーブルは，一般的に用いられている複雑な形状を持った Gauss 関数の電子反発積分の対角項を利用する手法よりも計算コストが非常に小さい s 型 Gauss 関数の重なり積分を利用する手法を採ることにした．これは，カットオフテーブル計算に時間はフラグメントあたり 1 秒未満と短いが，計算するフラグメント数が多いため計算時間をできるだけ短くするためである．また，通常用いられる複雑な Gauss 関数ではなく s 型 Gauss 関数の重なり積分にもフラグメントの形状などの各種情報が含まれていると考えられることも s 型関数の重なり積分をカットオフテーブル計算に用いた理由の 1 つである．電子反発積分，4 中心クーロン積分，3 中心クーロン積分，2 電子積分に対して，計算する積分数をカットオフ計算で得られた結果を基に推定する式を作成した．カットオフテーブル計算を行うことで得られるデータは，全原始基底関数ペア数，重なり積分が十分に大きな原始関数ペア数である．また，全原始基底関数数も入力データから算出できる．これらの値を用いて，計算する積分数を推定する式を求めた．得られたデータを用いて，計算する積分数を推定する式を探索したが，分子積分の種類によって最も実測データに合う推定式の形が違うという結果になった．ここでは，電子反発積分についての結果のみを示すことにする．図 3.1.2 は，カットオフ計算で得られた生き残り原始関数ペア数より求めた原始関数 4 重対のうち，実際に積分計算を行った原始関数 4 重対の割合（生き残り率）が原始関数数とどのように依存しているかを示したグラフである．このグラフから，フラグメントに含まれる原始関数数が増加するに従って，生き残り率が減少していくことが分かる．この関係を 1 次式や 2 次式，反比例の式などの様々な式で近似することを試みたが，いまのところ，累乗関数への近似が，もっともよく実測値を再現している（決定係数 $R^2\fallingdotseq0.93$）．そこで，計算する電子反発積分数（N_{ERI}）を，以下の式で近似することにした．

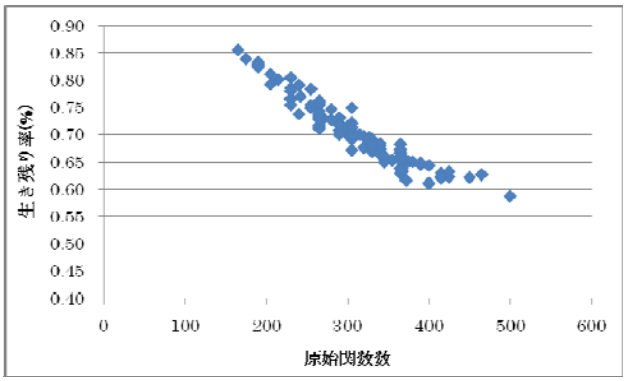

図 3.1.2. 2 電子積分の生き残り率と原始関数数との相関

$$N_{\mathrm{ERI}} = 5.7003 \times \left(n_{\mathrm{PS}}\right)^{-0.36617} \times N_{\text{total PS quartet}}$$

ここで，n_{PS} は原始基底関数数，$N_{\text{total PS quartet}}$ はカットオフ計算結果から得られる原始基底関数 4 重対の数である．

その他の 4 中心クーロン積分,3 中心クーロン積分,2 中心クーロン積についても,原始基底関数数と生き残り率の相関を調べて，カットオフテーブルの計算結果を用いて計算する積分数を推定する式を求めた．SCF 計算時間は，計算する電子反発積分数の推定値，および，フラグメントの基底関数の数に相関していると考えて，実行時間の推定式を求めた．近似式の詳細については割愛するが，4 中心クーロン積分数，3 中心クーロン積分数，2 中心クーロン積分数，および，繰り返し回数を 2 回に固定した場合の SCF 計算時間の近似式の決定係数（R^2）は，それぞれ，0.94，0.84，0.999，および，0.998 であった．

3.2 NSIM を用いた通信衝突のペナルティ計算手法の評価解析

通信衝突は，あるリンクにおいて，同時刻かつ同一方向に通信が実行された場合に発生する．数万から数十万の計算ノードを有するスーパーコンピュータにおいては，この通信衝突により生じる通信性能の低下（以降，通信衝突ペナルティと呼ぶ）を考慮した設計が必要となる．また，システム運用時においても，通信衝突ペナルティを短時間で求めることができれば，これをタスク配置最適化に利用することも可能となる．このような要求を満足するため，我々の研究グループでは，同時に実行され得る通信に基づき短時間で通信衝突ペナルティを計算する手法を提案している．本手法では，通信衝突が発生した際のパケット優先度を考慮して，発生するペナルティを算出する．図 3.2.1 は，通信衝突ペナルティの見積りにおいて，パケット優先度の考慮の必要性を示す．この図では，BlueGene/L と同様，自ノードからのパケットに低い優先度を与えている．ここで，通信衝突を発生させずに 1 つのメッセージを 1 ホップ先に送付するのにかかる時間を 1 time unit と定

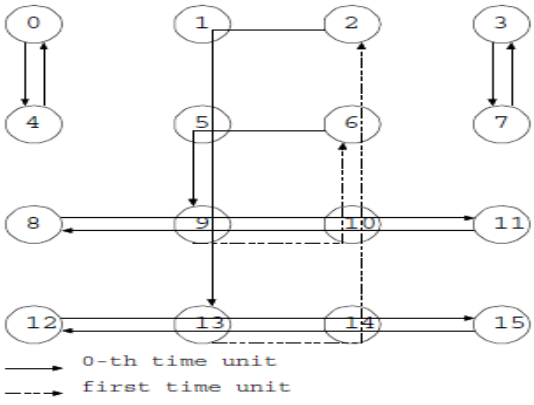

図 3.2.1. パケット優先度を考慮しないと通信衝突によるペナルティが正確な予測できない例

義する．図 3.2.1 における破線は 1 time unit で通信が開始，実線は 0 time unit で通信が開始されることを示す．破線の通信は 0 time unit の際にパケット優先度によりブロックされた通信である．もし，パケット優先度を考えていなければ本システムにおける最大の通信要求数は 2 であるため，通信にかかる時間は 2 time unit と予測される． 一方，パケット優先度を考慮した場合は，ノード 6, 10 間で 1 time unit で開始された通信が衝突するのでこのシステムの通信時間は 3 time unit となる．

この通信衝突ペナルティ算出法では，同時に実行される通信ならびに各リンクで同一方向に要求される通信の個数を調べ，メッセージサイズと通信帯域幅から通信時間を求める．以下，通信遅延を算出するアルゴリズムを示す．ここで，メッセージサイズ msize は全ての通信において等しいとする．また，通信帯域幅 B は全てのリンクにおいて等しいと仮定し，送信するメッセージ集合を M，リンク集合 L と表記する．これに加え，ある time unit における各リンクの通信要求数のカウンタ集合を Lc，メッセージ 1 つを通信した時の遅延を 1 とした時の各リンクの通信時間を格納する集合を Ld とする．関数 TL(m)は，あるメッセージ m が通過するリンク番号の集合を返す.この時,ルーティングが Dimension order routing (DOR) であるとして処理する．図 3.2.2 に通信衝突によるペナルティの算出法を示す．まず，初期化を行う．送信すべきメッセージの集合 M を定義する．さらに集合 Lc, Ld の初期化を行う．ここで，最初に

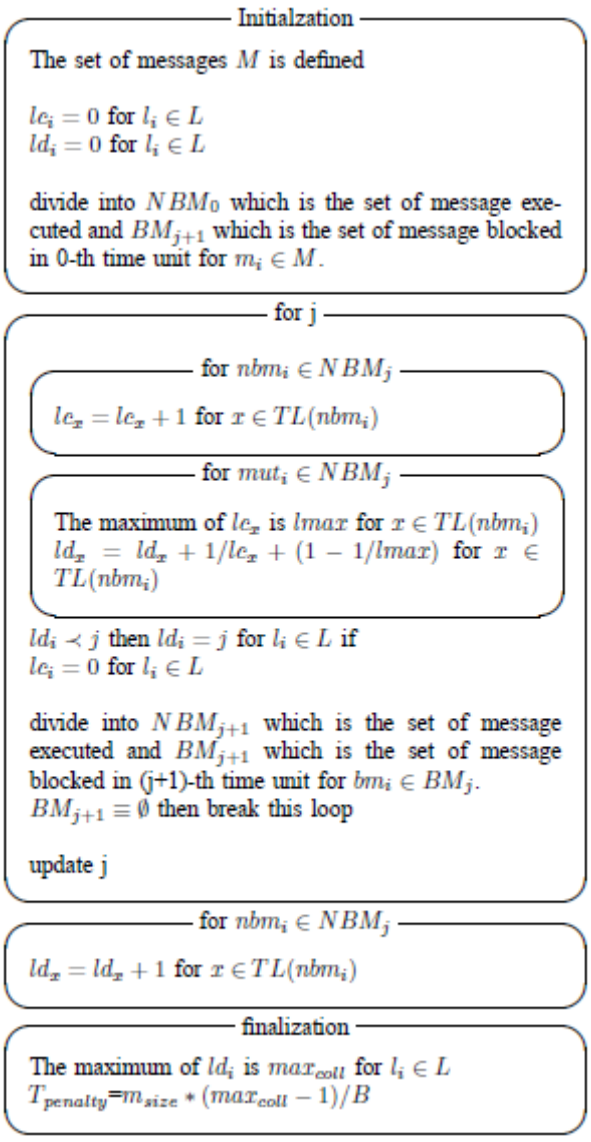

図 3.2.2.　通信衝突によるペナルティの算出方法

ブロックされるメッセージの選別を行う．これにより 0-th time unit において実行されるメッセージを得る．この時，パケット優先度を考慮してブロックされない通信を選ぶ．次に各 unit time において実行される通信の遅延を調べる．j-th time unit おいて実行される通信 NBMj による遅延を計算する.NBMj による各リンクの要求通信数をカウントする．通信 nbmi に関して各リンクの通信要求数の最大値 lmax を求める．複数のメッセージでリンクを平等に使用するため，nbmi は 1 time unit で 1-1/lmax のメッセージがブロックされることになる．このため，この値を通信の遅延として加算する．次の time unit の処理に移る時には，j-th time unit において ld が j に満たなかったリンクは遅延を j と更新する．同時に j-th time unit における通信要求数 Lc を初期化する．以上の作業をブロックされる通信の集合 BMj+1 が空集合となるまで実行する．最後に残った BMj 通信の集合による各リンクに対する通信要求数を加算する．通信遅延がもっと大きかったリンクを調べ，その値から通信衝突によるペナルティ Tpenalty を求める．

表 3.2.1. NSIMの設定

相互結合網仕様	
トポロジ	3D-Mesh
リンクスループット	4GB/s
スイッチバンド幅	4GB/s
ルーティング計算時間	4ns
仮想チャネルアロケーション時間	4ns
スイッチアロケーション時間	4ns
スイッチ遅延時間	78ns
ケーブル遅延時間	10ns
MTUサイズ	2048B
ノード設定	
HCA転送レート	4GB/s
DMA転送レート	4GB/s
MPIオーバヘッド	200s

表3.2.2. 実験結果

	予測時間	比率(%)
NSIM 実行結果	624usec	100.0%
提案算出法	524usec	80.1%

直接網におけるパケット優先度を考慮して通信衝突によるペナルティを予測することの有用性を示すため，NSIM を用いた実験を行った．NSIM に入力するインターコネクトの構成情報を表 3.2.1 に示す．NSIM を用いることにより，実機で発生する外乱や揺らぎの影響が無い状況での性能評価が可能であり，さらに計測結果がログ収集による影響を受けないため，正確な解析を行うことができる．なお，各通信のメッセージサイズは全て 1MB とした．表 3.2.2 に実験結果を示す．シミュレーション結果では通信衝突のペナルティは 624 usec となった．これに対し，

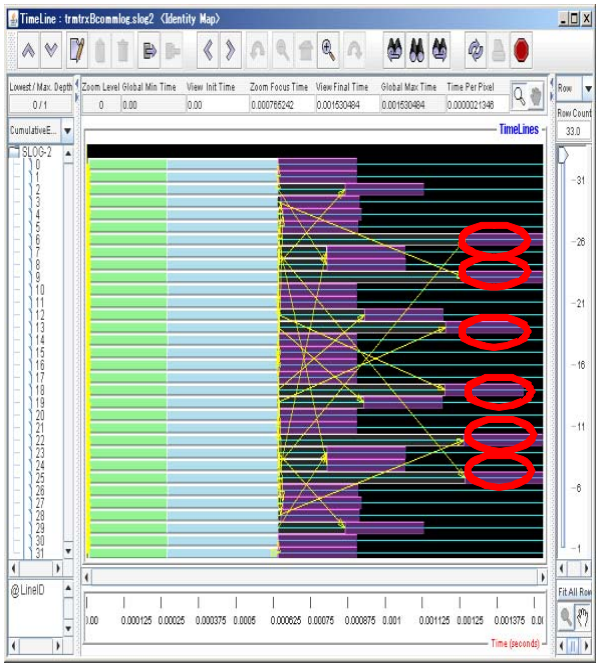

図 3.2.3. NSIM シミュレーション結果（JumpShot 表示）

提案している通信衝突パネルティ算出法では 524 usec となり，誤差は 20%程度であることが分かる．

NSIM は通信ログ出力機能を有しており，JumpShot を用いてインターコネクトの状態を表示することができる．この様子を図 3.2.3 に示す．これにより，直接網におけるパケット優先度を考慮した通信衝突によるペナルティの予測ではノード 6，9，13，18，22，25 は 2 time unit になるとされる．図 3.2.3 の JumpShop による通信の様子から予測と同一のノードのペナルティが 2 time unit となっていることが分かる．これにより，通信衝突によるペナルティを想定通りに予測できていることを確認できた．

4. これまでの進捗状況と今後の展望

4.1 BSIM における実効性能予測精度の改善

本研究での目標は，性能予測のターゲットとなるシステムよりも数桁性能が小さな計算機を用いて，高速かつ正確（ターゲットマシン実測値からの誤差が±10%未満の精度）で FMO 計算プログラムの性能推定を実現することである．これを達成するため，これまでに FMO 計算に現れる小規模電子状態計算の実行時間モデルを構築した．現在，小規模クラスタ計算機（約 50GFLOPS）を用いて，九州大学情報基盤研究開発センターの大型計算機 PRIMEQUEST（約 1.6TFLOPS）をターゲットとした場合の大規模 FMO 実行時間推定実験を実施している．今後はこの結果に基づき，必要に応じて更なる改善を行うと共に，学術論文としてまとめる予定である．

4.2 NSIM を用いた通信衝突ペナルティ算出法の評価解析

NSIM を用いた性能解析について，九州大学において提案した直接網インターコネクトにおける通信衝突ペナルティ算出方法の評価解析を行った． その結果，提案手法では 20%以内の精度で通信衝突によるペナルティを予測できるという結果を得た．また，NSIM が出力する通信ログを解析し，通信衝突によるペナルティを想定通りに予測できていることを確認した．今後の課題としては，NSIM を用いたより詳細な解析を行い，精度の更なる改善を試みる予定である．

5. 研究成果リスト

(1) 学術論文

無し

(2) 国際会議プロシーディングス

Y. Morie, T. Nanri, R, Susukita and K. Inoue, ″A Method for Predicting a Penalty of Contentions by Considering Priorities of Routing among Packets on Direct Interconnection Network″, in Proceedings of the International Joing Conference on Computational Sciences and Optimization 2011(to appear).

(3) 国際会議発表

無し

(4) 国内会議発表

無し

(5) その他（特許，プレス発表，著書等）

無し