PSI-NSIM: 大規模並列システムの性能解析に向けた並列相互結合網シミュレータ

柴村, 英智
九州システム情報技術研究所

薄田, 竜太郎
福岡県産業・科学技術振興財団

本田, 宏明
九州大学情報基盤研究開発センター

稲富, 雄一
九州大学情報基盤研究開発センター

他

出版情報：電子情報通信学会技術研究報告, CPSY2007-32. 107 (276), pp.45-50, 2007-10-25. IEICE
バージョン：
権利関係：

PSI-NSIM: 大規模並列システムの性能解析に向けた並列相互結合網シミュレータ

柴村　英智[†]　薄田竜太郎[††]　本田　宏明[†††]　稲富　雄一[†††]

于　雲青[†††]　井上　弘士[††††]　青柳　睦[†††]

† (財) 九州システム情報技術研究所　〒 814–0001 福岡市早良区百道浜 2–1–22
†† (財) 福岡県産業・科学技術振興財団　〒 810–0001 福岡市中央区天神 1–1–1
††† 九州大学情報基盤研究開発センター　〒 812–8581 福岡市東区箱崎 6-10-1
†††† 九州大学大学院システム情報科学研究院　〒 819–0395 福岡市西区元岡 744
E-mail: †shibamura@isit.or.jp

あらまし　大規模並列システムの設計開発ならびに性能解析に向けた相互結合網シミュレータ PSI-NSIM について述べる．PSI-NSIM は，評価対象とする相互結合網の仕様を記述した仕様ファイルとアプリケーション実行から生成した通信プロファイルを基にシミュレーションを行う．所望する相互結合網の評価に必要な各種情報を出力するのみならず，システム全体の性能を高速かつ精度良く予測するとともに，アプリケーションの性能解析や可視化のための各種情報も出力する．本稿では、シミュレータの実装、および既存のクラスタシステムの性能評価について報告する．
キーワード　スーパーコンピュータ，相互結合網，シミュレータ，PSI-NSIM

PSI-NSIM: A Parallel Interconnection Network Simulator for Performance Analysis of Large-scale Parallel Systems

Hidetomo SHIBAMURA[†], Ryutaro SUSUKITA[††], Hiroaki HONDA[†††], Yuichi INADOMI[†††],

Yunqing YU[†††], Koji INOUE[††††], and Mutsumi AOYAGI[†††]

† Institute of Systems & Information Technologies/KYUSHU
†† Fukuoka Industry, Science & Technology Foundation
††† Research Institute for Information Technology, Kyushu University
†††† Graduate School of Information Science and Electrical Engineering, Kyushu University
E-mail: †shibamura@isit.or.jp

Abstract　This paper presents an interconnection network simulator, PSI-NSIM, toward designing and performance analysis of large-scale parallel system. PSI-NSIM simulates desired interconnection network base on a configuration file which specifies specification of target network and a communication profile generated from an execution of application. Furthermore, this simulator provides not only various information for performance evaluation but estimates entire performance of system with fast and good accuracy. Then information for performance analysis and visualizing of application execution is also provided. In this paper, implementation of PSI-NSIM and results of performance evaluation of existing cluster system are reported.
Key words　Supercomputer, Interconnection network, Simulator, PSI-NSIM

1. はじめに

大規模並列理システムにおいて，多数の計算ノードを高速に相互接続するインターコネクト技術はシステム全体の性能を左右するため非常に重要であり，これまでに様々な仕様のシステムインターコネクト（相互結合網）が提案されてきた．また，実際の設計開発では，所望するインターコネクトについてシミュレーションなどによる事前の性能評価が行われて

いる．このような評価段階においては，インターコネクトの諸特性のみならず，アプリケーションとの親和性も鑑み，実践的な並列アプリケーションや計算ノードの振る舞いも併せて熟慮することが重要である．インターコネクトの性能評価を目的とした研究には，INSIGHT [1], [2] や INSPIRE [3] がある．また，アプリケーションの振る舞いも加味したシステムの性能予測技術として，EXCIT+INSPIRE [4]，MPIETE [5]，MPI-SIM [6]，BigSim [7] などが開発されている．これらの共通点は，システムの振る舞いを細部にわたりモデリングする方針が採られていることである．

一方，近年の大規模システムは，飛躍的な計算ノードの性能向上とともに，実行するアプリケーションも大規模かつ複雑化している．したがって，新しい並列システムの設計においては，開発時に利用可能な実用技術の範ちゅうで，広大な設計空間を探索し，最適なシステム構成を決定しなければならない．そのためにも，シミュレーションによって種々のアプリケーションを模擬し，高い精度でシステム性能を見積もる必要がある．しかし，詳細な評価のためにインターコネクトや計算ノードを忠実にモデル化しすぎた場合には，シミュレーションに要する時間は実機速度と比較して数桁以上遅くなる場合もある．この問題は，数万ノードの計算機資源を利用するペタスケール級のシステム評価ではいっそう深刻となり，実用時間内に性能評価を完了することが困難になることが予想される．したがって，シミュレーションの精度とシミュレーションに費やす時間の均衡を保つ必要がある．そこで，我々は，システムの性能評価を目的としたシミュレーションでは，インターコネクトの具体的なモデル化に加えて，計算ノードやアプリケーションの振舞いを高度に抽象化する技術が重要と考えた．

本研究では，次世代スーパーコンピュータの開発現場で利用できる精度を持ち，実用時間内に大規模システムの性能評価を行う，PSI-SIM [8] と呼ぶシステム性能評価環境を開発している．本環境は，ペタフロップス級スーパーコンピュータの振る舞いをシミュレーション可能とし，大規模インターコネクトやシステム全体の性能を高速かつ精度良く見積もり，大規模アプリケーションの効果的な性能解析支援や可視化の実現を目指している．本稿では，PSI-SIM において大規模インターコネクトのシミュレーションを行う PSI-NSIM について述べる．

以下，第 2 章ではシステム性能評価環境 PSI-SIM で用いる大規模システムの性能予測手法について述べ，第 3 章で PSI-SIM とその構成について概説する．また，第 4 章で PSI-NSIM のしくみについて述べ，第 5 章で PSI-NSIM を用いた評価実験を行う．そして，第 6 章でまとめと今後の課題について述べる．

2. 大規模システムの性能予測

本章では，大規模システムの性能評価環境に求められる要件をまとめ，数万ノードを有するペタスケール級システムの性能を実用時間内で予測する手法について述べる．

2.1 大規模システム向け性能評価環境への要件

スーパーコンピュータに代表される大規模計算機システムは，設計開発コストが極めて大きい．そこで，システムの仕様検討から設計・開発段階に至るまで，幅広い評価が重要となる．このような評価には，既存コンピュータを利用したシミュレーションによって設計空間を探索する方法が多く用いられる．このような大規模システム向け性能評価環境への要件として，以下の項目があげられる．

高速性：シミュレーションによる評価は，実機実行と比較して多くの時間を費やす．この問題は，シミュレーション対象の大規模化とともに，さらに深刻になる．そこで，実用時間内にペタフロップス級システムのシミュレーションを完了できること．

正確性：大規模システムでは，計算ノードの構成だけでなくインターコネクトの構成もシステム全体の性能へ影響を与える．したがって，大規模インターコネクトの振る舞いも考慮した精度の高い性能予測ができること．

外挿性：シミュレーションに利用可能な計算機システムは，性能評価対象システムよりも小規模である場合が多い．そこで，既存の小規模システム（例えば，テラフロップス級マシン）を用いて，大規模システム（例えば，ペタフロップス級マシン）の性能見積もり（外挿）ができること．

以上の要件を満たすには，従来のシミュレーション手法とは異なる新しい性能予測手法が必要である．次節では，評価にかかるコストを削減しつつ設計空間を効果的に探索できる，プログラムコード抽象化に基づいた性能予測手法について述べる．

2.2 プログラムコード抽象化に基づく性能予測手法

一般的に，並列システムのシミュレーションは，システムの構成要素の観点から，計算ノード部とインターコネクト部のシミュレーションに大別できる．また，従来はそれぞれについて詳細なモデリングを行うことが多いため，統合的なシミュレーションには非常に多くの時間を費やす．そこで，実際に並列システムで評価対象プログラムを別途実行し，その際の通信履歴や計算処理時間を通信プロファイルとして記録し，再度そのプロファイルを計算ノードの振る舞いとして，インターコネクトシミュレーションへの入力する手法 [2] がある．この研究では，通信ログの取得に PC クラスタシステムを用いている．これは，一つの計算ノード上に仮想的な複数の計算ノードを構築し，小規模（16 台）のクラスタシステムで評価対象プログラムを実行することで中規模システム（256 台）に相当する通信プロファイルを生成している．また，通信プロファイルの生成機構を持つ MPE(Multi-Processing Environment) [9] においても，プログラムを実行することで通信プロファイル生成している．しかし，アプリケーションが大規模かつ複雑化すると通信プロファイルの生成にも時間がかかるため，ペタフロップス級の大規模アプリケーションを既存の小規模システムで実行した場合には極めて多くの時間を費やしてしまう．そこで，通信プロファイル自体も高速に生成する必要がある．

このような問題を解決するため，高速な通信プロファイルの生成を目的としたプログラムコードの抽象化手法を PSI-SIM では採用している．まず，前述のようにプログラムコードを並列システムで実行することで得られる，通信履歴や計算処理時間を含んだ通信プロファイルに着目する．ここで，最終的に必要とする評価項目は，性能評価対象システム上でのプログラ

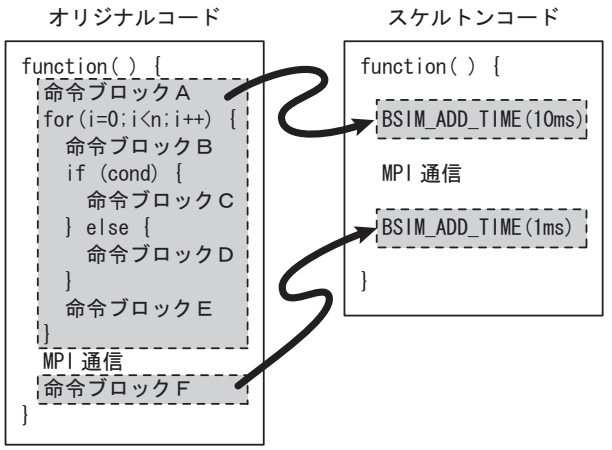

図 1　プログラムコードの抽象化（スケルトン化）

ム実行時間，およびインターコネクトの諸性能であるため，**実行結果（計算結果）**は不要である．換言すれば，通信に影響を及ぼさないプログラムコード部分に関してはその**実行時間**を，また，通信部分についてはその**通信発生タイミング**を通信プロファイル中へ反映することが重要となる．

そこで，我々は，計算ノード間の通信に影響を与えないプログラムコード部分を抽出し，これを「実行時間」という極めて抽象度の高い表現に置換えている．このような抽象化により作成されたプログラムコードを本研究では**スケルトンコード**と呼んでいる．スケルトンコードの例を図 1 に示す．これは，MPI 通信の前後に存在する 2 つの命令コード部分の抽象化（スケルトン化）の例を表している．これらのコード部分は，スケルトンコードでは BSIM_ADD_TIME 関数の呼び出しに置換され，本関数の引数には対応するコード部分の見積り実行時間が与えられる．そして，スケルトンコードを実行すると，BSIM_ADD_TIME 関数は引数の時間分だけ通信プロファイルに刻印される時刻を進める．したがって，コード部分の抽象化精度（見積もり実行時間の精度）が高い場合は，オリジナルコードを実行して生成された通信プロファイルと，オリジナルコードを実行して生成された通信プロファイル中にプログラム終了時刻は同一となる．

従来の性能予測手法は，プログラムコードの細部まで忠実に模擬する方針だが，本手法は相反して，予測精度に応じて可能なかぎりコードを粗く抽象化するという方針である．

2.3　スケルトンコードへの変換

以下に，オリジナルコードをスケルトン化する手順を示す．

（1）　抽象化対象となる連続コード部分を選択する．

（2）　性能予測対象計算ノードにおける当該コード部分の実行時間を見積もる．

（3）　当該コード部分を BSIM_ADD_TIME 関数で置換し，上記 (2) で求めた見積り実行時間を引数として与える．

PSI-SIM が提供する通信プロファイル生成ツールは，BSIM_ADD_TIME 関数から渡された引数の値（つまり，見積り実行時間）に基づき，通信プロファイルを生成するための内部時計を更新する．このようにして，図 1 の場合，通信プロファイル生成ツールはオリジナルコード内の 2 つの命令コード部分を実行することなく通信プロファイルを生成することができる．

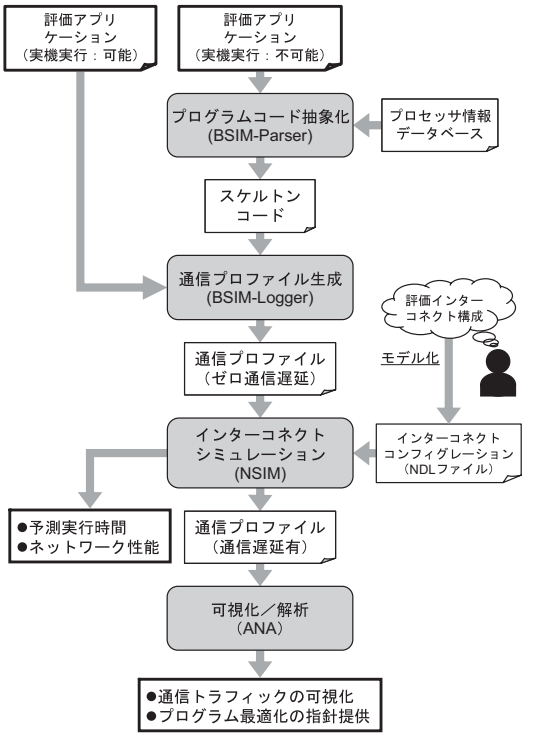

図 2　システム性能評価環境 PSI-SIM のワークフロー

3.　システム性能評価環境 PSI-SIM

現在，我々は，ペタフロップス級の大規模並列計算機システムの性能予測を目的とし，プログラムコード抽象化手法に基づく大規模システムの性能評価環境 PSI-SIM を開発している．PSI-SIM は 4 つのソフトウェアツールから構成されており，図 2 にそれらのワークフローを示す．まず，評価アプリケーションが既存システムで実行できる規模かを判断し，実行できない場合には，プログラムコードをスケルトン化する BSIM-Parser によってスケルトンコードを生成する．その際に，BSIM_ADD_TIME 関数に渡される見積もり実行時間は，別途用意しているプロセッサ情報データベースから得る．次に，通信プロファイルを生成する BSIM-Logger は，評価アプリケーション，もしくはスケルトン化されたコードを，MPI 処理環境上で性能評価対象システムの規模に応じた実行を行う．ここで，BSIM-Logger は，通信レイテンシ「ゼロ」の理想ネットワークを想定した通信プロファイルを生成する．また，この通信プロファイルには各プロセス間通信の発生時刻や送信/受信プロセスに関する情報などが含まれる．NSIM は，並列離散事象シミュレーションに基づくインターコネクトシミュレータである．NDL と呼ぶ仕様ファイルにインターコネクトをモデル記述することで，所望するネットワークのシミュレーションを行うことができる．BSIM-Logger が生成した通信プロファイルを入力とし，評価アプリケーションの各通信における実遅延時間をシミュレーションによって算出する．NSIM はゼロ通信遅延時間の通信プロファイルから通信遅延時間付きの通信プロファイルを生成し，その後に可視化／解析ツールである ANA によって各種の評価を行う．

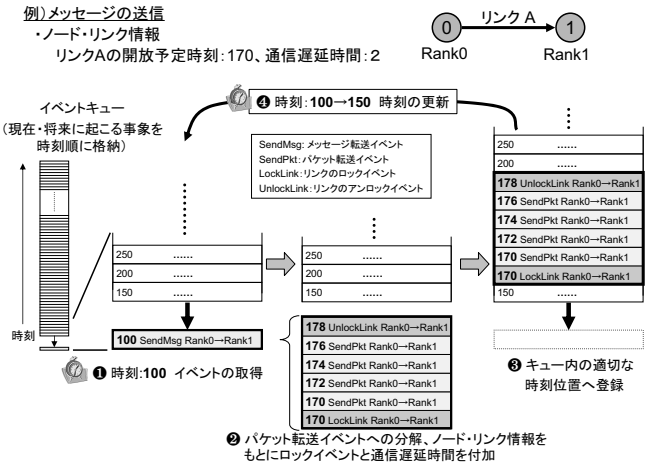

図 3　NSIM の基本シミュレーションフロー

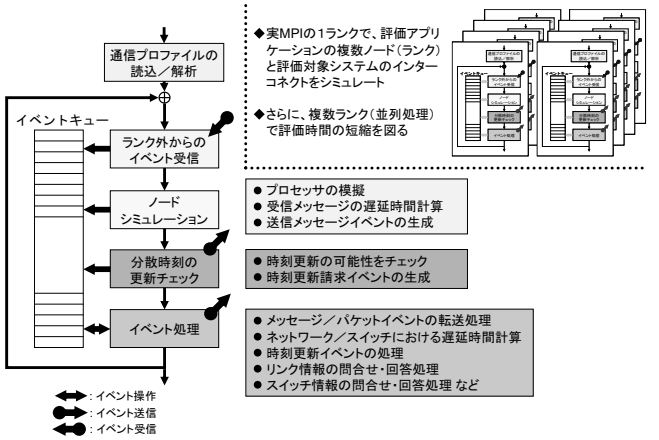

図 4　PSI-NSIM の主要処理フロー

現在，PSI-SIM 環境が持つ性能評価能力について，これらのツールの個別評価を行っている．BSIM-Parser に関する評価については，文献 [10] を参照されたい．

4. 相互結合網シミュレータ PSI-NSIM

本章では，PSI-NSIM の内部処理について説明する．

4.1 イベントキューを用いた離散事象シミュレーション

PSI-NSIM のシミュレーション機構は，イベントキューを用いた離散事象シミュレーションに基づいている．図 3 に，シミュレーションの流れを示す．

シミュレーション時に発生する様々な事象（イベント）には，その事象が処理される時刻を持っており，それらの事象はすべて単一のイベントキューに時刻順に格納されている．また，シミュレータはユニークな現在時刻を持っており，この時刻が進むことでシミュレーションが経過する．

図 3 は，MPI 環境においてランク 0 を持つノードからランク 1 を持つノードへのメッセージ通信の例を示している．ここで，2 つのノード間はリンク A で接続されており，そのリンクは利用中で時刻 170 において解放される予定である．また，リンク間の通信遅延時間は 2 とし，イベントキューの先頭には時刻 100 における事象が格納されているものとする．1) 現在時刻が 100 になると，キューの先頭事象の処理時刻と合致するため，当該事象を取得する．2) 取得した事象はメッセージの送信事象であるためパケット転送事象に分解する．リンク A が解放される時刻 170 の情報を基に，リンク A を時刻 170 にロックする事象，パケットを遅延時間 2 で転送する事象，そして，リンク A を時刻 178 に解放する事象を生成する．さらに，3) 生成された事象をキュー内の適切な時刻位置に挿入する．最後に，4) キューの先頭事象の時刻 150 まで現在時刻を進める．

このような一連の処理を繰り返し，インターコネクトの振る舞いを操作する様々な事象を生成・処理することで，シミュレーションが行われる．

4.2 PSI-NSIM の処理フロー

図 4 に，PSI-NSIM の主要処理フローを示す．PSI-NSIM は，前述のイベントキューに対して，様々な通信処理や計算処理に関する事象を当該時刻に挿入するノードシミュレーション部を持つ．また，先のシミュレーション機構を並列化し，並列離散事象シミュレーション (PDES: Parallel Discrete Event Simulation) に基づくシミュレータとして実装している．

まず，BSIM-Logger によって生成された通信プロファイルを読込んだ後，メッセージ通信の依存関係などを解析し，ノードシミュレーションの情報として保持する．次に，ノードシミュレーション部では，現在時刻に応じて通信プロファイルから解析・生成されたメッセージ転送事象をイベントキューに登録する．そして，イベント処理部において前節のシミュレーションフローを処理する．

PSI-NSIM は MPI 環境において並列化しており，このような処理フローが各ランクにおいて実行される．したがって，ランク間での事象の転送や情報の交換が必要となるため，必要に応じて他ランクとの送受信が行われる．また，全体のシミュレーション処理に矛盾が発生しないように全ランクの現在時刻を同期させるための分散時刻管理を行っている．

5. PSI-NSIM の評価実験

本章では，PSI-NSIM の実行速度，ならびに性能予測能力について評価実験を行う．

5.1 実 験 内 容

本実験では，BSIM-Logger によって生成した通信プロファイルを入力とし，NSIM を実行することで得られる以下の 2 つの項目について調査する．

(1) NSIM のシミュレーション時間

(2) 評価アプリケーションの予測実行時間

1) を評価することで，実用時間内での性能予測が遂行可能か否かを判断する．なお，本実験では，理想ネットワーク環境下，すなわち通信遅延をゼロとした並列シミュレーションを実行する．これにより，評価アプリケーション毎の並列シミュレーションに要する最低限の時間がわかる．

一方，2) の評価では，既存のクラスタシステム上での実行時間と予測実行時間を比較し，BSIM-Logger と NSIM による予測精度を明らかにする．

表 1　実験環境 1 (1CPU: デスクトップ PC)

CPU	Intel Xeon 3.8GHz (Single core)
メモリ	2GB
OS	Linux version 2.6.20-1.2320.fc5
コンパイラ	GNU C Compiler ver.4.1.1
MPI	mpich2-1.0.5p4

表 2　実験環境 2 (2CPU〜: クラスタシステム)

CPU	Intel Xeon 3.0GHz (Single core)
メモリ	7GB
ノード数	16 (2CPUs/node)
ネットワーク	InfiniBand (1xLink DDR), GigabitEthernet
OS	RedHat Enterprise Linux AS release 3
	(Linux Kernel 2.4.21)
コンパイラ	Fujitsu Fortran&C Compiler ver.5.0
MPI	Fujitsu MPI over SCore,
	mpich2-1.0.5p4 over SCore

5.2 実験手順

前述の調査項目毎の実験手順を以下に示す.

(1) NSIM のシミュレーション時間の測定

BSIM-Logger を利用して，評価アプリケーションを理想ネットワーク環境下で実行したと想定し，その際の通信プロファイルを生成する．この通信プロファイル，ならびに評価したいシステムの仕様を記述した NDL ファイルを入力とし，クラスタシステム上で NSIM を実行する．この実行により，通信遅延時間をゼロとした NSIM のシミュレーション時間を得る．

(2) 評価アプリケーションの予測実行時間の測定

評価アプリケーションを既存のクラスタシステムで実行し，その実行時間を測定する．一方，BSIM-Logger を利用して，評価アプリケーションを理想ネットワーク環境下で実行した場合の通信プロファイルを生成する．この通信プロファイル，ならびに同クラスタシステムをモデル記述した NDL ファイルを入力とし，NSIM によって通信遅延時間やハードウェアの仕様を考慮したネットワークシミュレーションを行う．この実行により，評価アプリケーションの予測実行時間を得る．

5.3 実験環境

本実験で利用した環境の仕様を表 1 と表 2 に示す．なお，単一 CPU システムによる実用的な性能評価の可能性を調査するために，一般的なデスクトップ PC 環境での実験も行った．

NSIM に入力する評価インターコネクトのモデルは，表 2 で示す実験環境と同じ単一の InfiniBand スイッチによるクラスタ構成とした．MPI メッセージ通信におけるスタートアップ遅延時間については，同クラスタシステムで別途測定した結果である 11.6 マイクロ秒を用いた．ただし，この値にはスイッチ 1 段あたりの片道遅延を含んでいる．また，NSIM によるシミュレーションの基本単位時間は 1 ナノ秒とした．

5.4 評価アプリケーション

評価アプリケーションには HPL [11] を用いた．HPL の諸パラメータとして，問題サイズ (N) を 500，1,000，2,000，5,000，また，プロセス数 (P×Q) を 4×4，16×16，32×32 と変化させ，ブロック数 (NB) は 128 に固定した．これらのパラメータの組合せについて BSIM-Logger で生成した通信プロファイルを表 3 に示す．表中のアプリケーション例として，HPL.N2000.16x16.4R は，問題サイズ 2,000，プロセス数 16×16 について，MPI の 4 ランクを利用した BSIM-Logger の実行から生成したことを意味する．また，それぞれのプロセス数，BSIM-Logger によって算出された理想実行時間についても併記している．通信プロファイル中には，プロセス数と同数の MPI ランクの振舞いが存在する．したがって，NSIM はプロセス数に応じた数の MPI ランクのシミュレーションを行う．

表 3　評価アプリケーション

通信プロファイル	プロセス数	理想実行時間（秒）
HPL.N500.4x4.16R	16	0.120
HPL.N1000.4x4.16R	16	0.207
HPL.N2000.4x4.16R	16	0.786
HPL.N5000.4x4.16R	16	8.042
HPL.N2000.16x16.4R	256	0.651
HPL.N2000.32x32.16R	1,024	1.019

5.5 実験結果

(1) NSIM のシミュレーション時間

図 5 に，問題サイズを変化させ，プロセス数を 16 に固定した場合の NSIM のシミュレーション時間を示す．また，図 6 に，問題サイズを固定し，プロセス数を変化させた場合のシミュレーション時間を示す．なお，これらの測定結果には，ANA による性能解析で用いるメッセージの送受信イベントや計算イベントのファイル出力時間を含んでいる．本実験では，この出力イベントファイルの大きさは数 10M バイトから最大で約 180M バイトになった．

図 5 からは顕著な台数効果は確認できない．これは，問題サイズの増加，すなわち，通信プロファイル中の通信メッセージが増えるとともに NSIM が単位時間あたりに実行する 1 ランク中のイベント処理数が増えるが，CPU 数も多くなりシミュレーション時刻の同期オーバヘッドが増加するためであった．

同様に，図 6 では，ある程度の並列化効率が確認できるが，CPU 数が 32 付近ではシミュレーション時間が遅くなっている．これは，プロセス数が増加，すなわち，NSIM を実行するランク数が増加）することによって通信メッセージも増え，単位時間あたりのイベント処理効率が向上するが，CPU 数が多くなりすぎることによって，1 ランクあたりのイベント処理数が低下するためであった．

本実験では，一般的に HPL で用いる問題サイズより小さなサイズで評価を行ったため，NSIM の並列化効率が十分に得られなかった．しかし，1,024CPU（=1,024 プロセス）相当の並列システムと，その上で 48 万回の通信を行うアプリケーションのシミュレーションを，8CPU のクラスタシステムを利用して 5 分半程度で完了した．

(2) 評価アプリケーションの予測実行時間

図 7 に，PSI-NSIM による評価アプリケーションの実行時間の予測結果を示す．この結果から，問題サイズが大きくなるにつれて，実測値と NSIM による評価アプリケーションの予測実行時間との誤差が小さくなり，約 5.3%の誤差で性能を見積もっ

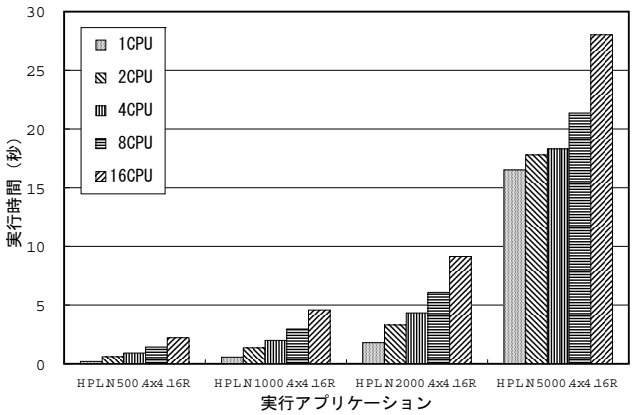

図 5　PSI-NSIM のシミュレーション時間 1

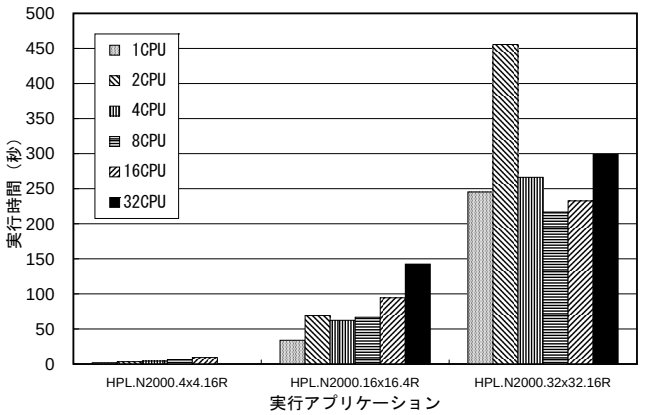

図 6　PSI-NSIM のシミュレーション時間 2

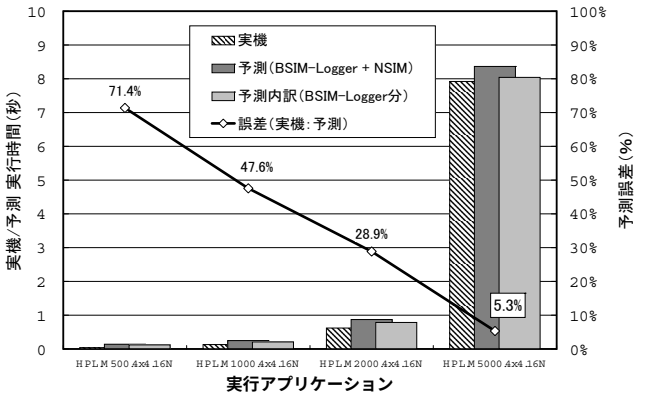

図 7　PSI-NSIM による実行時間の予測性能

ている．しかし，予測値が実測値を上回っており，また実測値と予測値の絶対値が徐々に増加している．これは，第 5.3 節で述べた，InfiniBand スイッチにおける片道遅延の累積によるものであると考える．したがって，NSIM におけるスイッチのモデリングについて今後検討する必要がある．

6. おわりに

本稿では，設計開発システムの性能を既存のシステムで実用時間内で予測する手法について述べた．本手法は，予測精度を保ちつつ，できるだけプログラムコードの抽象化を粗く行うという，従来手法とは相反する方針である．また，ペタフロップス級時代の次世代スーパーコンピュータの設計開発に向けた統合型システム性能評価環境 PSI-SIM について述べた．PSI-SIM 環境の一部である BSIM-Logger と NSIM を利用して，アプリケーションや既存のクラスタシステムの性能評価を行い，シミュレーション時間や見積もり誤差を基に本環境の有効性について議論した．その結果，PS-NSIM では，HPL アプリケーションを最大約 5.3 %の誤差で性能予測することができた．

現在，効果的な性能解析を支援するために各種情報の出力機構を拡張している．今後の課題として，PSI-SIM を構成する各ツールの連携を確立し，さらに大規模なアプリケーションを用いた性能評価実験を行う予定である．

謝　辞

本研究は，文部科学省「次世代 IT 基盤構築のための研究開発」，研究開発領域「将来のスーパー コンピューティングのための要素技術の研究開発」（平成 17 年度〜19 年度）における研究開発課題「ペタスケール・システムインターコネクト技術の開発」[12] によるものである．

文　献

[1] H. Shibamura, M. Kuga, and T. Sueyoshi: INSIGHT: An Interconnection Network Simulator for Massively Parallel Computers, *Proc. of IEEE Region 10's Ninth Annual International Conference (IEEE TENCON '94)*, Vol. 1of 2, pp. 77–81 (1994).

[2] 柴村英智，久我守弘，末吉敏則：超並列計算機のための相互結合網シミュレータ，情報処理学会論文誌，Vol. 35，No. 4，pp. 589–599 (1994).

[3] 原田智紀，曽根猛，朴泰祐，中村宏，中澤喜三郎：並列処理用ネットワークのための性能評価用シミュレータ生成系 INSPIRE，情報処理学会研究報告 ARC-95-113-9，pp. 65–72,(SWoPP'95), (1995).

[4] 久保田和人, 板倉憲一, 佐藤三久, 朴泰祐：大規模データ並列プログラムの性能予測手法と NPB2.3 の性能評価，情報処理学会論文誌, Vol. 40, No. 5, pp. 2293–2304 (1999).

[5] 堀井洋，岩渕寿寛，山名早人：MPI プログラム実行時間予測ツール MPIETE の評価，情報処理学会研究報告 2003-HPC-097, Vol. 2004, No. 20, pp. 55–60 (2004).

[6] S. Prakash, and Rajive L. Bagrodia: MPI-SIM: using parallel simulation to evaluate MPI programs, *Proc. of the 30th conference on Winter simulation*, pp. 467–474 (1998).

[7] G. Zheng, G. Kakulapati, and L. V. Kalé: BigSim: A Parallel Simulator for Performance Prediction of Extremely Large Parallel Machines, *18th International Parallel and Distributed Processing Symposium (IPDPS'04)*, p. 78b (2004).

[8] 柴村英智，薄田竜太郎，本田宏明，稲富雄一，于雲青，井上弘士，青柳睦：次世代スーパーコンピュータの設計開発に向けたシステム性能評価環境 PSI-SIM，情報処理学会研究報告 2007-HPC-111 (SWoPP'07), pp. 267–272，(2007).

[9] Performance Visualization for Parallel Programs: `http://www-unix.mcs.anl.gov/perfvis/download/`

[10] 松本幸，本田宏明，薄田竜太郎，柴村英智，井上弘士，青柳睦，村上和彰：大規模並列システムの性能評価を目的としたプログラムコード抽象化技法，情報処理学会研究報告 2007-HPC-111 (SWoPP'07) pp. 55–60, (2007).

[11] A. Petitet, R. C. Whaley, J. Dongarra, and A. Cleary: HPL – A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers, `http://www.netlib.org/benchmark/hpl/`

[12] Petascale System Interconnect Project: `http://www.psi-project.jp/`