社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－HPC－111 (46)
2007／8／3

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

柴村英智[†1] 薄田竜太郎[†2] 本田宏明[†3]
稲富雄一[†3] 于雲青[†3]
井上弘士[†4] 青柳睦[†3]

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について述べる．本環境は，実践的な並列アプリケーションから生成した通信プロファイルを基に，所望する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を支援する．本稿では，通信プロファイルを高速に生成するためのプログラムコード抽象化手法を提案する．また，PSI-SIMによる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について議論する．

PSI-SIM: A System Performance Evaluation Environment Toward Next Generation Supercomputer Development

HIDETOMO SHIBAMURA,[†1] RYUTARO SUSUKITA,[†2]
HIROAKI HONDA,[†3] YUICHI INADOMI,[†3] YUNQING YU,[†3]
KOJI INOUE[†4] and MUTSUMI AOYAGI[†3]

This paper presents a system performance evaluation environment, PSI-SIM, toward peta-scale next generation supercomputer development. This environment estimates performances of desired interconnect and system based on communication profile which generated from execution of practical parallel application, and supports easy application analysis and visualization. We propose a program code abstraction method for fast communication profile generation. Furthermore, PSI-SIM simulates applications and an existing cluster system, then the elapsed simulation times and the error rates of the estimation are discussed.

1. はじめに

並列処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る．このような評価が行われ始めた黎明期は，対象とするインターコネクト単体の評価が主体であった．その後，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考慮されるようになり，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問われるようになった．このような，インターコネクトの性能予測技術として，INSIGHT[1)~3)]やINSPIRE[4)]がある．また，システムの性能予測技術としては，EXCIT+INSPIRE[5)]，MPIETE[6)]，MPI-SIM[7)]，BigSim[8)]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抽象化する方針が採られていることである．

一方，近年の大規模スーパーコンピュータにおいては，飛躍的なシステム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になってきた．このような高性能コンピュータの設計においては，広大な設計空間を探索し，最適なシステム構成を決定する必要がある．そのために，コンピュータ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

†1 (財) 九州システム情報技術研究所
Institute of Systems & Information Technologies/ KYUSHU
†2 福岡県産業・科学技術振興財団
Fukuoka Industry, Science & Technology Foundation
†3 九州大学情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University
†4 九州大学大学院システム情報科学研究院
Graduate School of Information Science and Electrical Engineering, Kyushu University

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4桁以上遅くなる場合もある．この問題は，数万から数十万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掛ける時間の均衡を保つ必要がある．すなわち，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のみならず，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る．

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 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するもので，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予測し，大規模アプリケーションの効果的な性能解析や可視化の実現を目指している．

以下，第2章では我々が提案する性能予測手法について議論し，第3章でシステム性能評価環境 PSI-SIM とその構成ソフトウェア群について述べる．また，第4章で PSI-SIM のネットワークシミュレーション部を用いた基礎的な評価実験を行う，第5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，大規模な計算機システムの性能評価環境に求められる要件をまとめ，数万から数十万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を提案する．

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，幅広い性能評価が重要となる．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：コンピュータ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．したがって，ペタフロップス級システムのシミュレーションを実用時間内に完了できること．

拡張性：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) のシミュレーションができること．

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がシステム全体の性能へ与える影響も大きい．したがって，インターコネクトを含めた精度の高い性能予測ができること．

以上の要件を満たすためには，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望まれる．次章では，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を提案する．

2.2 プログラムコード抽象化に基づく性能予測手法

大規模並列計算機システムの性能を高速に予測するためには，ネットワークシミュレーションだけでなく，その入力となる通信プロファイルも高速に生成する必要がある．しかし，MPE(Multi-Processing Environment)[9] 等を用いた従来の通信プロファイル生成では，対象プログラムを実際に実行しなければならない．そのため，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す．

この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提案する．まず，MPE のようにプログラムコードを並列処理環境で実行し，その際の実行時刻や通信イベントを通信プロファイルとして生成するツールが存在すると仮定する．ここで，性能評価対象システム上でのプログラム実行を想定した通信プロファイルについて着目すると，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**を取得して通信プロファイルへ反映することが重要であり，**実行結果**そのものは不要であるといえる．そこで，計算ノード間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手法では**スケルトンコード**と呼ぶ．スケルトンコードの例を図1に示す．ここでは，MPI 通信の前後に存在する2つの命令コード部分に対する抽象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，この `BSIM_ADD_TIME` 関数は，プログラムコード抽象化後の通信プロファイル生成時に処理されるものとする．すなわち，通信プロファイル生成のためにスケルトンコードが実行されると，当初のコード部の実行時間に相当する値が通信プロファイルに出力されるというしくみである．

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

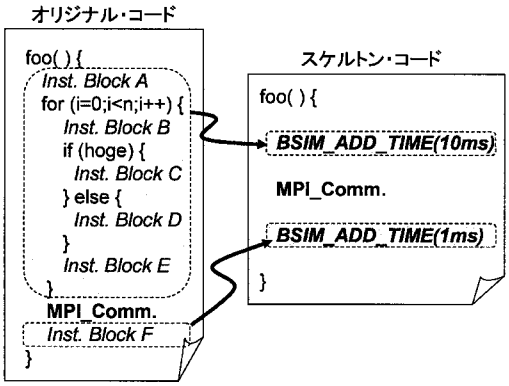

図 1　プログラムコードの抽象化

いう方針を採っている．

2.3　スケルトンコードへの変換手順

以下に，オリジナルコードをスケルトンコードに変換する手順を示す．

(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．

先に仮定した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精度の高い通信ログを生成することができる．

3.　システム性能評価環境 PSI-SIM

現在，我々は，プログラムコード抽象化手法に基づいた大規模システムの性能評価環境 PSI-SIM を開発している．本環境は，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主な目的としており，4 つのソフトウェアツールから構成される．

図 2 に PSI-SIM 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，前述のプログラムコードを抽象化する 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．

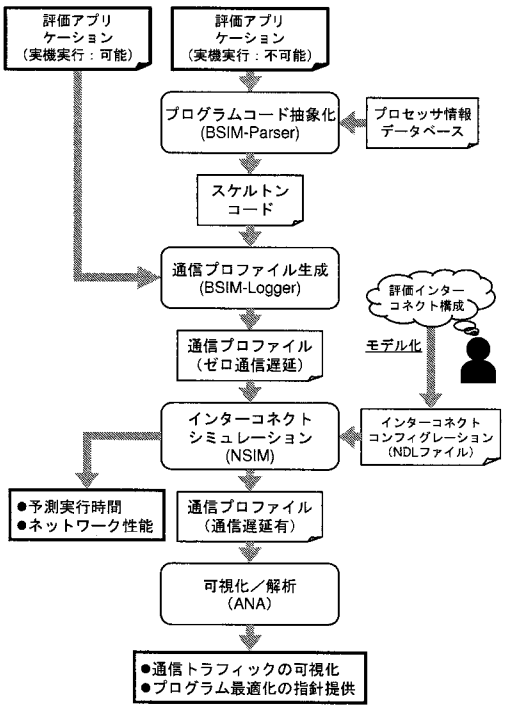

図 2　システム性能評価環境 PSI-SIM のワークフロー

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．以下，本稿では NSIM の性能予測能力について述べる．

4.　PSI-NSIM の評価実験

本章では，PSI-SIM 環境においてネットワークシミュレーションを担う NSIM について評価実験を行う．

4.1 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

前者を評価することで，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，後者の評価では，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，NSIM の予測精度を明らかにする．

4.2 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．

4.3 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．

表 1 実験環境 1 (1CPU: デスクトップ PC)

CPU	Intel Xeon 3.8GHz (Single core, Hyper-threading, EM64T)
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, Hyper-threading, EM64T)
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

4.4 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，ならびに HPL の実行に必要とした通信メッセージ数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．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．

4.5 実験結果

(1) NSIM のシミュレーション時間

表 4 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し，図 3 にグラフを示す．また，表 5 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し，そのグラフを図 4 に示す．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 3 から，問題サイズが大きくなるにつれて全体的に並列化効率が現れ出していることがわかる．しかし，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

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）	通信メッセージ数
HPL.N500.4x4.16R	16	0.120	4,767
HPL.N1000.4x4.16R	16	0.207	9,267
HPL.N2000.4x4.16R	16	0.786	18,267
HPL.N5000.4x4.16R	16	8.042	45,267
HPL.N2000.16x16.4R	256	0.651	175,169
HPL.N2000.32x32.16R	1,024	1.019	484,823

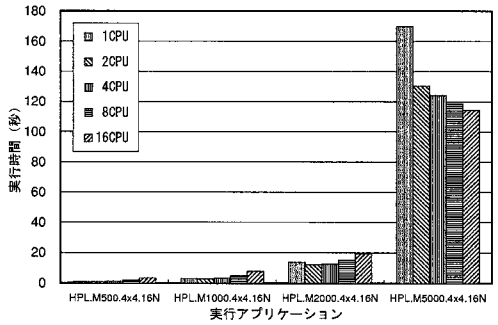

図 3　PSI-NSIM のシミュレーション時間 1

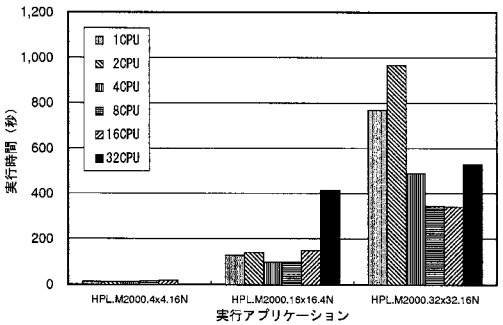

図 4　PSI-NSIM のシミュレーション時間 2

する 1 ランク中のイベント処理数も増えるが，CPU 数が多くなることによるシミュレーション時刻の同期オーバヘッド増加のためであった．

同様に，図 4 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

表 6 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し，グラフを図 5 に示す．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 7% の誤差で性能を見積もっている．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 4.3 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．

表 6　PSI-NSIM による実行時間の予測性能

通信プロファイル	実測値	予測値	誤差
HPL.N500.4x4.16R	0.04	0.15	74%
HPL.N1000.4x4.16R	0.13	0.27	52%
HPL.N2000.4x4.16R	0.62	0.91	32%
HPL.N5000.4x4.16R	7.92	8.54	7%

単位（秒）

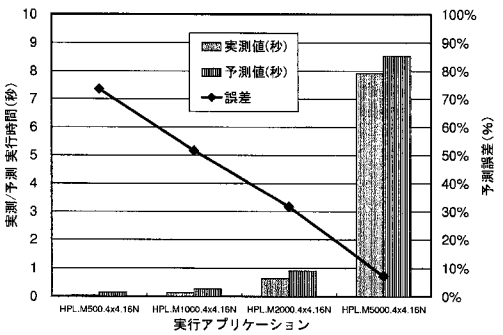

図 5　PSI-NSIM による実行時間の予測性能

5. ま と め

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を提案した．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

表 4 PSI-NSIM のシミュレーション時間 1（問題サイズ可変，プロセス数固定）

通信プロファイル	1CPU	2CPU	4CPU	8CPU	16CPU	32CPU
HPL.N500.4x4.16R	0.823	0.961	1.199	1.969	3.468	–
HPL.N1000.4x4.16R	3.008	2.912	3.344	4.854	7.686	–
HPL.N2000.4x4.16R	13.710	12.203	12.578	15.231	19.313	–
HPL.N5000.4x4.16R	169.777	130.174	124.150	118.652	114.45	–

※ 送受信・計算イベントのログ出力に要するファイル出力時間を含む　　単位（秒）

表 5 PSI-NSIM のシミュレーション時間 2（問題サイズ固定，プロセス数可変）

通信プロファイル	1CPU	2CPU	4CPU	8CPU	16CPU	32CPU
HPL.N2000.4x4.16R	13.710	12.203	12.578	15.231	19.313	–
HPL.N2000.16x16.4R	127.247	140.411	98.089	97.741	148.863	414.949
HPL.N2000.32x32.16R	767.895	966.888	489.500	344.374	339.987	529.594

※ 送受信・計算イベントのログ出力に要するファイル出力時間を含む　　単位（秒）

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 7 %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．

参 考 文 献

1) 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).
2) 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).
3) 柴村英智，久我守弘，末吉敏則：相互結合網シミュレータ INSIGHT の並列化について，電子情報通信学会技術研究報告 CPSY94-32 (SWoPP'94), pp.41-48 (1994).
4) 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).
5) 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).
6) 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).
7) 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).
8) 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).
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), (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/`