

## 異機種計算機間 MPI 通信ライブラリ Stampi による VPN を介した SCore クラスタシステム間通信性能評価

辻田 祐一† 山岸 信寛† 木村 和幸†  
大谷 孝之† 鶴岡 信彦†† 藤田 直行†††

PC クラスタにおける並列処理を支援する目的で開発された SCore は、利用者からは PC クラスタ内の各ホストを意識させないシステムとして作られており、SCore が稼働しているクラスタは一つの計算機のように扱える。SCore における MPI 通信ライブラリとして MPICH-SCore があるが、クラスタ間通信を含む異機種計算機間では MPI 通信が出来ない。本研究では、この SCore に、異機種計算機間 MPI 通信ライブラリ Stampi を移植し、クラスタ間の通信を可能にした。さらに Stampi によるクラスタ間の通信性能の評価を行なった。

### Evaluation of Communication Performance among SCore Cluster Systems via VPN using MPI Communication Library, Stampi

YUICHI TSUJITA,† NOBUHIRO YAMAGISHI,† KAZUYUKI KIMURA,†  
TAKAYUKI OTANI,† NOBUHIKO TSURUOKA†† and NAOYUKI FUJITA †††

SCore system has been developed to provide flat computing infrastructure for parallel computing without awareness of each compute host in a PC cluster system. An SCore cluster system can be handled as a single computer. MPICH-SCore, which is an MPI library in SCore, is not available across multiple SCore cluster systems as well as multiple platforms. We have implemented an MPI communication library, Stampi in SCore cluster system to enable MPI communication between SCore cluster systems and measured communication performance of the library using SCore cluster systems.

#### 1. はじめに

近年、PC クラスタ・システムが、手軽に大規模な計算資源を得る手段として注目されている。ただし、クラスタ内の各ホストはそれぞれ独立したシステムとして動作しているため、利用者は各ホストを意識したプログラム開発をしなくては行けない。そこで、利用者

に各ホストを意識させないシステムとして、新情報処理開発機構で SCore<sup>1)</sup> が開発された。SCore が稼働しているクラスタ・システムは、一台の計算機として扱えるフラットな計算環境を提供している。さらに並列計算に必要な様々なツールセットが用意されており、その中で、SCore 上で MPI<sup>2),3)</sup> 通信が行なえる MPICH-SCore<sup>1)</sup> がある。MPICH-SCore は MPICH<sup>4)</sup> をベースに SCore 上に移植されたものであり、Myrinet やイーサネットなどの通信媒体を用いた高速通信が可能である。しかし、MPICH-SCore には動的に外部の計算機へプロセスを生成し、MPI 通信を行なう機能や SCore クラスタ間で MPI 通信を行なう機能はない。

一方、日本原子力研究所 計算科学技術推進センターでは、これまでに、異なる並列計算機間で利用可能な MPI 通信ライブラリ Stampi<sup>5)</sup> を開発してきた。Stampi は様々な並列計算機やワークステーションなどで利用可能である。今回、ITBL プロジェクト<sup>6)</sup> の一環として行なった通信試験において、SCore クラスタ間で MPI 通信を行なうために、MPICH-SCore と Stampi の計算機間通信機能を利用して、SCore ク

† 日本原子力研究所 計算科学技術推進センター  
Center for Promotion of Computational Science and  
Engineering, Japan Atomic Energy Research Institute  
(JAERI)

〒110-0015 東京都台東区東上野 6-9-3

†† 理化学研究所 情報基盤研究部 情報環境室  
Division of Computer and Information, The Institute  
of Physical and Chemical Research (RIKEN)

〒351-0198 埼玉県和光市広沢 2-1

††† 航空宇宙技術研究所 CFD 技術開発センター  
CFD Technology Center, National Aerospace Laboratory (NAL)

〒182-8522 東京都調布市深大寺東町 7-44-1

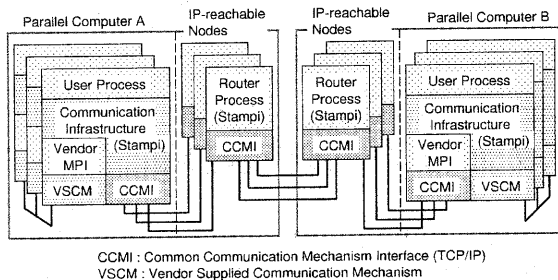


図 1 Stampi の機能図。

ラスタ間の動的プロセス生成や MPI 通信を可能にするためのライブラリの設計並びに開発を行なった。本稿では、まず始めに Stampi の機能について述べ、次に Stampi の SCore 対応における実装方法、理化学研究所と航空宇宙技術研究所（以下、それぞれ理研、航研と記す。）に配置された SCore クラスタ間を VPN ルータを用い、スーパー SINET で接続した環境での SCore 対応版 Stampi の性能評価テストについて述べ、最後に今後の開発予定について述べる。

## 2. 異機種計算機間 MPI 通信ライブラリ Stampi

並列計算機内部では、計算機ベンダが提供する高速な MPI 通信ライブラリが利用できるが、計算機間の MPI 通信は実現されていない。異なる計算機間でも MPI 通信を可能とするために、異機種計算機間 MPI 通信ライブラリ Stampi が開発された。Stampi の機能図を図-1 に示す。Stampi は MPI-2 に準拠しており、以下に述べる機能を実現している。

- (1) MPI-2 に準拠した動的プロセス生成機能
- (2) 計算機内部と計算機間の通信機構の動的切替え機能
- (3) 通信中継プロセスを用いた柔軟な通信基盤
- (4) 異機種計算機間で利用可能な MPI-I/O 機能<sup>7)</sup>
- (5) 計算機の仕様に依存しない柔軟な実装

以上の特徴により、各計算機の通信ならびに I/O システムの仕様の違いを意識することなく MPI による並列計算が自由に行なえる。以下、Stampi の持つ様々な機能のうち、特に今回のテストに関係する (1) から (4) の機能の詳細について説明する。

### 2.1 MPI-2 に準拠した動的プロセス生成機能

MPI-2 において新たに定められた機能に、MPI\_Comm\_spawn や MPI\_Comm\_spawn\_multiple による動的プロセス生成機能がある。これらは MPICH ではサポートされておらず、今回使用した SCore にあ

る MPICH-SCore でもサポートされていない。Stampi では、上記関数による動的プロセス生成機能を実装しており、利用者は、プロセス生成に関するパラメタ（ホスト名、ユーザー名、作業ディレクトリなど）を info オブジェクトに設定することにより、計算機間の動的プロセス生成が実現できる。

### 2.2 計算機内部と計算機間の通信機構の動的切替え機能

Stampi では、計算機内部の通信では計算機ベンダが提供する高速な通信ライブラリ（Vendor Supplied Communication Mechanism: VSCM）を、計算機間は共通通信プロトコル層（Common Communication Mechanism Interface: CCMi）のライブラリを利用しており、通信形態により、この二つの通信モードが動的に切替えられる。Stampi では、CCMi 層を介した通信には TCP/IP が用いられている。図-1 に示すように、ユーザー・プロセスからは、まず Stampi の通信基盤（Communication infrastructure）のライブラリを呼び出す。ここでこのライブラリを利用するのは、計算機内部の通信と計算機間の通信で、ユーザー・プロセス側から共通のインターフェースを持たせるため、計算機内部の通信の場合、この通信基盤から高速な計算機ベンダ提供の通信ライブラリが呼び出され、高速な通信が行なわれる。一方、計算機間の通信の場合、通信相手のプロセスと CCMi 層を通して通信経路が形成され、データ通信が行なわれる。また必要に応じ、次に説明する通信中継プロセスが起動され、これを介して外部と通信を行なう。

### 2.3 通信中継プロセスを用いた柔軟な通信基盤

計算機の通信において、特定のノードのみ外部と通信可能なものと、全てのノードが外部と通信可能なものがある。Stampi では、前者の場合、外部と通信が出来るノードに通信中継プロセスを起動し、これを介して外部と通信を行なう。一方、後者の場合、通信中継プロセスは起動せずに直接外部と通信を行なう。Stampi が用いている通信基盤は、この二つの通信パターンを計算機の通信機構に合わせて動的に切替える事が出来る。

さらに計算機間のデータ通信とリモート I/O の性能向上のために、図-2 に示す複数の通信経路が利用できる機能を通信中継プロセスに実装している。複数通信中継プロセスの起動方法は、計算機の通信機構に柔軟に対応している。例えば複数の通信中継プロセスが異なる外部通信ノードに起動され、各々が別々の通信経路を扱う場合（図-2 左側の起動方法）や、一つの通信中継プロセスが複数の通信経路を扱う場合（図-2

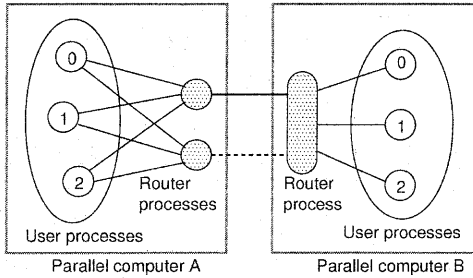


図 2 複数通信経路を用いた通信中継プロセス機能の概念図。

右側の起動方法) などがある。

#### 2.4 計算機の仕様に依存しない柔軟な実装

計算機間でデータ通信を行なう際、バイト・オーダーや浮動小数点の形式に注意を払う必要がある。Stampi では、MPI-2 における external32 形式に基づいた計算機間のバイト・オーダーや浮動小数点の形式の変換機能を通信基盤に実装したことにより、計算機間の自由なデータ通信を実現している。よって、利用者は各計算機の仕様を意識することなく、計算機内部並びに計算機間で自由に MPI 通信並びに MPI-I/O 機能が利用できる。

### 3. Stampi の SCore への対応

SCore における MPI 通信ライブラリである MPICH-SCore では、クラスタ内で管理されているグループに属する計算機単位でプログラムが実行されるが、異なるグループ間や異なるクラスタ間で MPI 通信を行なう機能はない。そこで、Stampi が持つ通信基盤を利用し、異なるグループあるいはクラスタに対し、TCP/IP を用いたプロセス起動機能と通信経路形成機能を実装することで、クラスタ間あるいはグループ間の MPI 通信を可能にした。SCore に対応した Stampi のアーキテクチャを図-3 に示す。この図に示すように、クラスタ内部のユーザー・プロセス間は MPICH-SCore による高速通信が行なわれる。一方、クラスタ間の通信では、通信相手の計算機にユーザー・プロセスを起動し、TCP/IP により接続された通信経路を用いて MPI 通信を行なう。また、この図に示すように各計算機が直接外部と通信が出来ない場合は、サーバーホストに通信中継プロセスを起動し、これを経由して通信を行なう。

次に、プログラム起動方法について述べる。Stampi の起動コマンドを、SCore 特有の起動方式に対応させた。SCore 対応の Stampi の起動方式は以下に示すようになる。("\ " は継続を意味する。以下同様。)

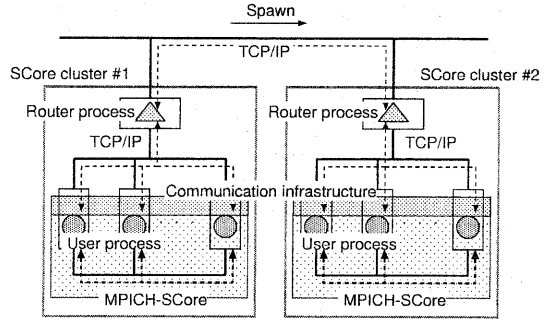


図 3 SCore に対応した Stampi のアーキテクチャ。

```
% jmpirun -node 2 -group pccall,myrinet2k \
program
```

この中で、"program" は起動するプログラムである。"-node"、"-group" は、それぞれ使用するノード数、使用するノードが属する SCore 上のグループ名を指定している。さらに計算機間の通信に利用する通信経路が複数ある場合、通信経路の指定もできる。この例では、パラメタとして Myrinet を使用するパラメタ、myrinet2k を指定している。指定がない場合は SCore でのデフォルトの通信経路が使用される。

Stampi での異機種計算機間の MPI 通信では、MPI-1 並びに MPI-2 形式の両方をサポートしている。この機能を SCore にも移植した。それぞれの場合でのプログラムの起動方式を以下に説明する。

例として、計算機 A (host-a) と計算機 B (host-b) があり、計算機 A から起動コマンドを実行するというケースを考えてみる。MPI-1 形式で、計算機間で通信を行なう場合、以下のような起動になる。

```
% jmpirun -node 1 -group half0 program \
: -node 1 -group half1 -host host-b \
program
```

この場合、計算機 A と計算機 B にそれぞれ 1 プロセスずつ起動され、計算機 A、B 全体で一つの MPI\_COMM\_WORLD を形成する。計算機間の通信は、TCP/IP を用いた通信となる。一方、MPI-2 形式では、以下の起動コマンドで、まず計算機 A 上にプログラムが起動される。

```
% jmpirun -node 1 -group half0 program
計算機 B へは、計算機 A 側のプログラムの中に記述された MPI_Comm_spawn 等の動的プロセス生成関数でプロセスが起動され、計算機間で MPI 通信が可能になる。ここで、MPI_Comm_spawn 等を用いた動的プロセス生成において、ホスト名、ユーザーアカウント名、作業ディレクトリや、使用する PE 数などが info オ
```

表 1 info オブジェクトに設定される SCore 対応のキー一覧。

キー名	機能
node	使用する計算ホストの数
group	使用する計算ホストが属するグループ名、計算ホスト間で使用するネットワーク

表 2 使用した SCore クラスタシステムの仕様。

サーバーホスト 1 台 + 計算ホスト 8 台	
サーバーホスト (NEC Express5800/56Wf)	
CPU	Intel Xeon 2.2 GHz
メモリ	3.6 GByte
ネットワークインターフェース	100 Mbps Intel 82559 (オンボード)
	1 Gbps Intel Pro/1000 XT
計算ホスト (NEC Express5800/54We)	
CPU	Intel Pentium4 2.2 GHz
メモリ	2 GByte
ネットワークインターフェース	100 Mbps Intel 82559 (オンボード)
	1 Gbps Intel Pro/1000 XT
イーサネットスイッチ (クラスタ内接続用)	
1 Gbps	Cisco Catalyst 3550
100 Mbps	NEC ES100X/124

プロジェクトに設定され、それらがプロセス生成時に反映されるが、今回新たに SCore 用に表-1 に示すものを新たに実装した。計算機間の通信は、MPI-1 形式の時と同様に TCP/IP を用いる。

#### 4. 性能評価

SCore クラスタを理研と航研にそれぞれ一台ずつ配置し、クラスタ間をスーパー SINET<sup>®</sup> を経由して接続した。使用したクラスタの仕様を表-2 に示す。サーバーホスト、計算ホスト間は 100 Mbps のイーサネット・スイッチ (NEC ES100X/124) を、計算ホスト間はギガビットのイーサネット・スイッチ (Cisco Catalyst 3550) を用いて接続した。

この性能評価試験において、データサイズはプログラム全体で扱うデータサイズを意味する。複数のユーザー・プロセスを起動した場合には、そのデータが均等に各ユーザー・プロセスに分配された後に処理が行なわれた。以下、性能測定の手順並びに測定結果について述べる。

##### 4.1 クラスタ内部での通信性能

始めに、MPICH-SCore と SCore に対応した Stampi で、クラスタ内部の通信性能を評価した。評価方法は、MPI-1 形式で MPI\_Send と MPI\_Recv による ping-pong 型通信をするプログラムを動かし、プロセス間通信のレイテンシーと通信性能を、MPICH-SCore

表 3 MPICH-SCore と SCore に対応した Stampi による、SCore クラスタ内の MPI 通信性能。表内の数字 (レイテンシー以外) の単位は MB/s。

	Latency ( $\mu$ s)	Data size(Byte)		
		1 M	16 M	256 M
MPICH-SCore	48	46.1	46.4	46.2
Stampi	48	46.2	46.3	46.2

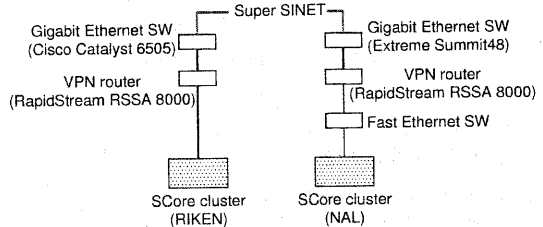


図 4 理化学研究所、航空宇宙技術研究所間で行なった SCore クラスタ間の Stampi による通信性能評価のテスト環境。

で直接起動する方法と Stampi を用いて MPICH-SCore を使用する場合で測定した。測定方法は、ユーザー・プロセス内部の MPI\_Send から MPI\_Recv までの時間を MPI\_Wtime で計測し、その半分を通信時間とした。この計測を複数回行ない、通信時間の最小値を用いて転送性能を算出した。この時の測定結果を表-3 に示す。MPICH-SCore を直接使った場合、帯域 (1 Gbps) の 37% ( $\sim 46/125 \times 100$ ) 程度の転送速度が得られた。一方、Stampi の共通通信基盤を介して MPICH-SCore を用いても、ほぼ同様の結果が得られており、通信性能が落ちることはなかった。これにより、Stampi を SCore に移植することにより、クラスタ内の通信性能を落とすようなことはないことが分かった。

##### 4.2 クラスタ間の通信性能

理研ならびに航研に配置された SCore クラスタ間をスーパー SINET を経由し VPN ルータで接続した。VPN ルータは、クラスタ間にセキュリティ上安全な通信経路を形成するために使用した。使用した VPN ルータは、理研側、航研側ともに RapidStream 社の RSSA 8000 であった。今回のテストのセットアップを図-4 に示す。理研側は、スーパー SINET からギガビットのルータ (Cisco Catalyst 6505) と VPN ルータを通して、1 Gbps (full duplex) のネットワークで SCore クラスタが接続された。一方、航研側は、スーパー SINET からギガビットのルータ (Extreme Summit48) と VPN ルータを介し、都合上、100 Mbps (full duplex) のネットワークで SCore クラスタが接続された。よってクラスタ間のバンド幅は 100 Mbps

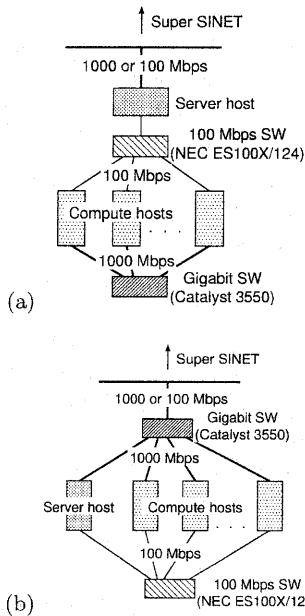


図5 (a)異なるセグメントと (b)同一セグメントでの Score クラスタ内部のネットワーク接続

であった。

通信性能評価におけるクラスタ内のネットワーク接続形態については、サーバー・ホストと計算ホストが異なるセグメントにある場合(図-5(a)参照)と、サーバーホストと計算ホストが同一セグメントにある場合(図-5(b)参照)の2つのケースをテストした。

まず、Stampi による MPI-2 形式での MPI\_Send、MPI\_Recv による ping-pong 通信でのクラスタ間の性能を計測した。時間の計測は、理研側のクラスタ上のユーザー・プロセスで MPI\_Send から MPI\_Recv までの時間を MPI\_Wtime により計測し、その時間の半分を通信時間とした。ユーザー・プロセスが複数ある場合、各計測毎にプロセス間で最も時間がかかったものを通信時間とした。さらにこの計測を複数回行ない、通信時間の最小値により転送性能を算出した。また、ユーザー・プロセスで送受信するデータサイズが0の時に上記の方法で測定した通信時間の半分をレイテンシーとした。理研、航研両方のクラスタともに、起動したプロセス数は同じで、1個、2個、4個、8個の4ケースをテストした。外部との通信においては、計算ホストがサーバーホストと異なるセグメント上にある場合は、計算ホストは直接外部と通信出来ない為、サーバーホスト上に起動した通信中継プロセスを経由して、外部と通信を行なった。一方、同じセグメント

表4 MPI-2型起動による理研、航研に配置された SCore クラスタ間の通信性能。表中の np は各クラスタに起動したプロセス数を意味する。表内の数字(レイテンシー以外)の単位は MB/s。

	np	Latency (ms)	Data size (Byte)			
			1 M	8 M	64 M	256 M
異なるセグメント	1	1.83	4.82	9.76	7.40	6.67
	2	1.75	5.13	11.9	8.00	6.72
	4	1.78	7.96	10.8	7.61	6.80
	8	1.78	11.8	14.0	7.82	6.92
同一セグメント	1	1.61	7.87	10.4	7.33	6.66
	2	1.60	10.6	12.4	8.91	7.51
	4	1.71	7.49	11.2	10.2	8.12
	8	1.69	10.3	11.6	12.5	9.33

にある場合は、各計算ホストは直接外部の計算機と通信を行なった。

このテストで得られた測定結果を表-4に示す。通信性能は、異なるセグメントの場合よりも、同一セグメントで行なった方が高い性能が出る事が分かった。異なるセグメントの時は、サーバーホストに通信中継プロセスが起動され、これを介してクラスタ間に一本の通信経路が形成されていた。一方、同一セグメントの場合、各計算ホストのユーザー・プロセスは通信相手のユーザー・プロセスと直接接続するので、異なるセグメントの時よりも高い通信性能が得られたと考えられる。

異なるセグメントの時は、データサイズが小さい所では、起動するプロセスの数を増やすことによる通信性能向上が見られたが、データサイズの大きいところでは、あまり性能が向上する事はなかった。これは、クラスタ間通信で1本の通信経路で出せるピーク性能に近い値を出していた為、その効果が出にくかったものと考えられる。一方、同一セグメントの場合、データサイズが大きい所でも性能が向上することが分かった。また両方のケースともに、データサイズが256 MByteのように大きくなると、通信性能がかえって低くなる傾向があった。この理由として考えられるのは、TCP/IPによるソケット通信の送受信バッファが、データサイズが小さい所では、一時的にデータを保持する余裕があったのが、データサイズが大きくなるにつれて、送受信バッファの余裕がなくなり、結果として通信処理能力が、ユーザー・プロセス側でデータを送り込むスピードに追いつかなくなり、通信性能が低くなってしまったと考えられる。

次に、StampiによるMPI-1形式でのMPI\_Send、MPI\_Recvによるping-pong通信でのクラスタ間の性能を計測した。理研側、航研側にそれぞれ1プロセ

表5 MPI-1型起動による理研、航研に配置されたSCoreクラスタ間通信性能。表内の数字(レイテンシー以外)の単位はMB/s。

	Latency (ms)	Data size (Byte)			
		1 M	8 M	64 M	256 M
異なるセグメント	1.80	4.48	8.78	7.18	6.72
同一セグメント	1.60	7.52	10.1	7.26	6.66

すずつ起動し、全体で一つのMPI\_COMM\_WORLDを形成し、この中でプロセス間通信を行なった。測定方法はMPI-2形式の計測の時と同様であった。前述の2つのネットワーク構成で測定した結果を表-5に示す。この結果と、MPI-2形式でのプロセス数1の場合と比較すると、レイテンシー、通信性能ともに、ほぼ同じ値である。この結果、クラスタ間通信において、MPI-1起動形式でもMPI-2起動形式とほぼ同じ性能でプログラム実行が可能であることが分かった。

## 5. ま と め

我々は、異機種計算機間MPI通信ライブラリStampiをSCoreに対応させ、クラスタ内部では、MPICH-SCoreを使い、クラスタ間では、TCP/IPによる通信を行なうようにした。

理化学研究所と航空宇宙技術研究所に配置されたSCoreクラスタをスーパーSINETを介してVPNルータで接続し、Stampiを用いてクラスタ間の通信性能を計測した。この評価試験で、各クラスタ内のサーバー・ホストと計算ホストのネットワーク接続を、異なるセグメントと同一セグメントの場合の2通りをテストした。Stampiによるクラスタ間通信において、前者の場合はサーバー・ホストに通信中継プロセスが起動され、各計算ホストはこれを介して外部と通信を行なった。一方後者の場合、各計算ホストは直接外部と通信を行なった。

この環境で、Stampiを用いたMPI-1およびMPI-2形式での通信試験を行なった。まず、MPI-2形式での試験では、異なるセグメントのケースに比べて、同一セグメントのケースの方が、通信中継プロセスを介することがなかったことや、クラスタ間に複数の通信経路が形成されたことなどにより、より高い性能が得られた。

また、このテストにおいて、ユーザー・プロセスの数を増やす事により、異なるセグメントの時は、データサイズが小さい時は性能向上が見られた。一方、同一セグメントの時は、全体的に性能向上が見られた。前者の場合、サーバー・ホストに通信中継プロセスが

起動され、さらにクラスタ間の通信経路が1本であったため、性能向上の効果が少なかったものと思われる。

次に、MPI-1形式によるテストでは、理研側、航研側にそれぞれ1プロセスずつ起動し、両方のクラスタ全体で1つのMPI\_COMM\_WORLDを形成し、クラスタ間の通信性能を評価したところ、MPI-2形式とほぼ同等の性能が得られた。

今後、クラスタ間をスーパーSINET経由で全体をギガビットのバンド幅で接続し、通信性能向上のための機能開発を行ないたい。

**謝辞** この研究は、ITBLプロジェクトの一環として行なわれました。この研究を進めるにあたり、理化学研究所 情報基盤研究部 情報環境室の姫野龍太郎氏、重谷 隆之氏、黒川 原佳氏には、今回の開発並びに性能評価を行なう環境を与えて頂き、さらに有用なコメントを頂きました。航空宇宙技術研究所 CFD 技術開発センターの中村 孝氏、山本 一臣氏、高木 亮治氏には、試験環境の構築をして頂くだけでなく、有用なコメントも頂きました。日本原子力研究所 計算科学技術推進センターの平山 俊雄氏、福田 正大氏、樋口 健二氏には試験の準備段階から様々なサポートをして頂きました。ここに感謝の意を表します。

## 参 考 文 献

- 1) PC クラスタコンソーシアム,  
<http://www.pccluster.org/>.
- 2) Message Passing Interface Forum: "MPI: A Message-Passing Interface Standard", June 1995.
- 3) Message Passing Interface Forum: "MPI-2: Extensions to the Message-Passing Interface Standard", July 1997.
- 4) W. Gropp and E. Lusk: "User's Guide for mpich, a Portable Implementation of MPI", (1998),  
<http://www-unix.mcs.anl.gov/mpi/mpich/>.
- 5) T. Imamura, Y. Tsujita, H. Koide and H. Takemiya: "An Architecture of Stampi: MPI Library on a Cluster of Parallel Computers", LNCS 1908, Recent Advances in Parallel Virtual Machine and Message Passing Interface, Springer, pp. 200-207.
- 6) ITBL プロジェクト, <http://www.itbl.jp/>
- 7) Y. Tsujita, T. Imamura, N. Yamagishi and H. Takemiya: "Stampi-I/O: Flexible Distributed Parallel-I/O Library for Heterogeneous Computing Environment", LNCS 2474, Recent Advances in Parallel Virtual Machine and Message Passing Interface, Springer, pp. 288-295.
- 8) [http://www.sinet.ad.jp/s\\_sinet/index.html](http://www.sinet.ad.jp/s_sinet/index.html)