

## 名大センターにおけるグリッドの取り組みについて（その2）

津 田 知 子 永 井 亨  
長谷川 明 生 石 井 克 哉

## ．はじめに

本センターニュース第4号（2003年8月号）で名大センターにおけるグリッドの取り組み状況を紹介しました。ここでは名大センターの取り組みとして、グリッド研究用として導入されたFujitsu PRIMEPOWER HPC2500の紹介が主になっていました。今回は、本センターで取り組んでいるグリッドの実証実験の内容について紹介することとします。現在、グリッド研究用システムとしては、ユーザに公開されているHPC2500とVPP5000に加えて、本センターの大規模計算支援環境研究部門の石井研究室の8台のPCクラスタにもグリッド環境を構築し、実証実験を行っています。

## ．グリッドの実証実験について

今年度本センターで取り組んだグリッドの実証実験としては、以下の5つがあります。ここでは、それぞれについて簡単に紹介します。

- 1．Globusの認証局の構築
- 2．MPICH-G2の基礎的性能の測定
- 3．応用プログラムにおけるMPICH-G2の性能測定
- 4．UNICORE環境の構築と並列計算実験
- 5．Titanium Languageの性能測定

## 2.1 Globusの認証局の構築

Globus環境では、さまざまなサイトの多数のシステムにアクセスする場合があります。使用するシステムでいちいち認証を受けるのは、利用者にとって煩雑です。この煩雑さを避けるため、1度の認証でこれらのシステムを利用できるようにする機能がSingle Sign-onと呼ばれるものです。Globus環境では、昨年8月まではGlobus CAと呼ばれる認証局が、Globusの利用者に証明書発行を行うCA（Certification Authority）サービスを行っていましたが、今年1月でそのすべてのサービスが停止されました。本センターでは、Globusの認証局としてこのGlobus CAを利用していましたので、サービス終了に伴い認証局の構築が必要になってきました。そこで、Globus Projectから提供されているSimpleCAを用いて、研究室のPCクラスタに認証局を構築しました。HPC2500ではこの認証局を使うことにしたのですが、構築したマシンが研究用のマシンで、他の

目的にも使用するため、ネットワーク環境が度々変更されるなど、グリッドの認証局としては、相応しくないことがわかりました。そこで、HPC2500でCAを立ちあげることを考えたのですが、この時点のHPC2500のGlobusは、SimpleCAが構築できないバージョンでした。HPC2500のGlobusのバージョンアップを行い、SimpleCAによる認証局を立ち上げ、現在、認証サービスを行っています。

## 2.2 MPICH-G 2の基礎的性能について

Globus環境下でMPIのプログラムを動作させるのがMPICH-G 2というソフトウェアです。MPICH-G 2は、異なるアーキテクチャのマシン上で動作する並列プログラムのメッセージ交換を自動的に行ってくれるので、ユーザは、アーキテクチャの違いを意識することなく、MPIプログラムを実行することができます。この実験では、同一機種間の実験としては本センターのHPC2500と京大のグリッド研究システムHPC2500（1ノード構成96CPU）を、異機種間の実験としては本センターのVPP5000と東北大のスーパーコンピュータSX-7に対して測定を行いました。測定に用いたプログラムは、3000×2105の行列の乗算を求めるプログラム（C版とFortran版）です。測定結果を表1、表2に示します。

表1 MPICH-G 2の基礎的性能（C版）

（単位：秒）

	elapse-time	elapse-time比	備考
mpi	298.76	1.00	MPI2.0 転送（クロスバ4GB/sec）
mpich-g 2（ngrd 1内）	310.77	1.04	mpich-g 2（1.24） 転送（システムネットGbE）
mpich-g 2（ngrd1, ngrd 2）	316.55	1.06	mpich-g 2（1.24） 転送（システムネットGbE）
mpich-g 2（名大, 京大間）	346.48	1.16	mpich-g 2（1.24） 転送（スーパーSINET）
mpich-g 2（ngrd1, vpp）	514.65	1.72	mpich-g 2（1.24） 転送（システムネットGbE）
mpich-g 2 （名大, 東北大SX-7間）	1012.4	3.39	mpich-g 2（1.24） 転送（スーパーSINET）

表1と表2におけるelapse-time比の値は、いずれもHPC2500でMPIを動作させたときの処理経過時間（elapse-time）を1としています。このMPIでは、通信に高速光インターコネク（クロスバ：4GB/sec）を用いた転送を行っています。これに対し、MPICH-G 2では、TCP/IPでの転送が行われます。表1と表2から、同一機種間でのMPICH-G 2の基礎的性能は、HPC2500の高速光インターコネクを用いた場合に比べて、C版、Fortran版共に2倍以内に納まり、十分に使用

に耐え得るといえるでしょう。一方、異機種間の測定では、3倍を超える結果となりました。この原因としては、東北大のSX-7が運用マシンであることから生じるCPUの競合によるプログラムの実行時間のばらつきやSX-7のTCPのドライバの性能の問題などが考えられます。

表2 MPICH-G 2の基礎的性能 (Fortran版)

(単位: 秒)

	elapse-time	elapse-time比	備考
mpi	43.84	1.00	MPI2.0 転送 (クロスバ 4 GB/sec)
mpich-g 2 (ngrd 1 内)	57.77	1.32	mpich-g 2 (1.2.4) 転送 (システムネットGbE)
mpich-g 2 (ngrd1, ngrd 2)	58.80	1.34	mpich-g 2 (1.2.4) 転送 (システムネットGbE)
mpich-g 2 (名大, 京大間)	69.52	1.59	mpich-g 2 (1.2.4) 転送 (スーパーSINET)
mpich-g 2 (ngrd1, vpp)	58.54	1.34	mpich-g 2 (1.2.4) 転送 (システムネットGbE)
mpich-g 2 (名大, 東北大SX-7間)	165.39	3.77	mpich-g 2 (1.2.4) 転送 (スーパーSINET)

### 2.3 応用プログラムにおけるMPICH-G 2の性能

ここでは、本センターのユーザが作成した応用プログラムにおけるグリッドの性能測定を試みました。測定に使用したプログラムは、以下の2本です。

- 結合コンパクト差分法を計算するプログラム  
(名古屋大学大学院工学研究科計算理工学 二瓶友典氏作成)
- 地球磁気圏の3次元MHDシミュレーションプログラム  
(名古屋大学太陽地球環境研究所 荻野竜樹氏作成)

測定は、いずれも8並列で行いました。測定結果を表3に示します。この結果から、3次元MHDのシミュレーションプログラムでは、1.36倍と非常によい結果が出ています。グリッド環境での応用プログラムの実行に十分期待が持てる値といえましょう。

表3 応用プログラムにおけるMPICH-G 2の性能

(単位：秒)

測定環境	差分法プログラム		3次元MHD	
	elapsed-time	ratio	elapsed-time	ratio
HPC2500 MPI	250.02	1.00	429.06	1.00
HPC2500 2ノード使用	286.15	1.14	429.96	1.00
名大(4CPU), 京大(4CPU)使用	728.78	2.91	582.63	1.36

#### 2.4 UNICORE環境の構築と並列計算実験

UNICOREとは、UNiform Interface to COmputing REsourcesの略で、Globusと同じグリッドのミドルウェアです。Globusがアメリカで利用されているのに対し、主に、ヨーロッパで利用されています。NAREGI (National Research Grid Initiative) でもUNICOREをベースにグリッドアプリケーション環境が作成されています。Globusと大きく異なる点は、UNICOREではGUIが用意され、ユーザはそれを使って、走らせるジョブの構成を準備したり、ジョブのモニタリングや制御を行えることです。今回は、前述のPCクラスタに図1に示すUNICORE環境を構築しました。

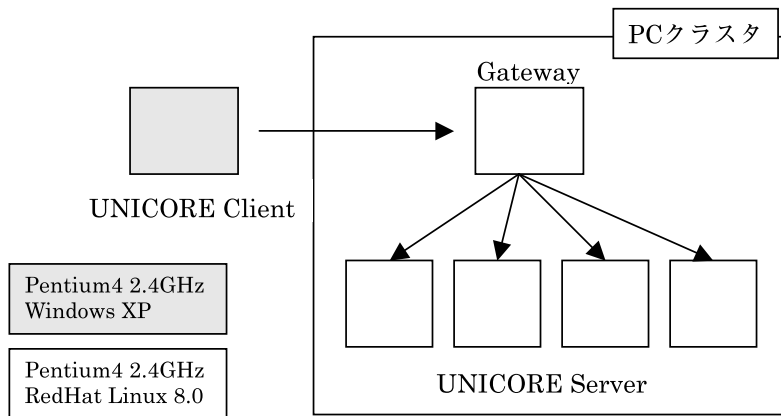


図1 UNICORE環境

そして、このUNICORE環境でJini技術を用いた並列計算を行いました。UNICORE環境を用いたことにより、1度の認証で複数台のPCに対して計算を実行するプロセスを立ち上げたり、計算の起動や停止の制御が可能になり、プロセッサ数の増加に伴う操作の煩雑さを減少することができました。現在は、Jini技術を利用しないで、UNICORE環境だけでの分散並列計算を行う実験に取り組んでいます。現状では、ファイル転送の負荷が大きく、並列性能を如何にして出すかが今後の課題となっています。

## 2.5 Titanium Languageの性能測定

Titanium Languageは、Javaをベースとした言語で、高い実行性能、プログラムの記述の容易さを目標とした並列計算用の言語処理システムです。このTitaniumでは、プログラム上必要な場合には、自動的にプロセス間通信を行うため、プログラマは、並列プログラムを特に意識することなく簡単に書くことができます。また、移植性にも富み、共有メモリでも分散メモリでも動作し、プロセス間の通信に対してもSSHやUDP等の指定をコンパイル時に指定することが可能となっています。今回は、共有メモリとしてグリッドシステムのHPC2500に、分散メモリとしてPCクラスタにTitaniumを移植し、並列性能を測定しました。ここで用いたプログラムは、結合コンパクト差分法のプログラムです。図2にHPC2500での並列性能を示します。この図からSMPでの並列性能の良さが確認できます。現在、PCクラスタでは、プロセッサを増加させた場合の並列性能に問題があります。これが、何に起因しているか今後の課題です。

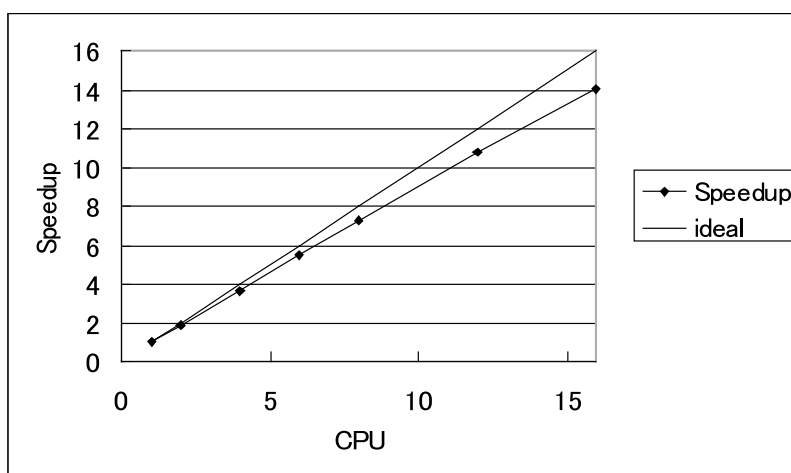


図2 HPC2500におけるTitaniumの性能

．おわりに

HPで公募しているグリッド研究推進実験プロジェクトには、4件の応募がありました。2-2の性能測定に利用させていただいた応用プログラムは、そのプロジェクトのプログラムであったことを附記しておきます。手持ちのプログラムをグリッド環境で流してみたいと思われる方は、是非、グリッドの実験プロジェクトに応募してください。

なお、UNICORE環境の構築とTitaniumの性能測定は、石井研究室の溝渕君、東君、宮本君によるものです。HPC2500のGlobus環境の構築に関しては、本センターのSE松井氏及び富士通のグリッドの開発部隊の方に多大なご協力を頂きました。ここに記して感謝します。

<参考文献>

- [1] <http://www.globus.org/>
- [2] <http://www3.niu.edu/mpi/>
- [3] <http://www.unicore.org/>
- [4] <http://www.cs.berkeley.edu/projects/titanium/>
- [5] <http://www.naregi.org/>

(つだ ともこ：名古屋大学情報連携基盤センター学術情報開発研究部門)

(ながい とおる：名古屋大学情報連携基盤センター大規模計算支援環境研究部門)

(はせがわ あきうみ：中京大学生命システム工学部

前名古屋大学情報連携基盤センター大規模計算支援環境研究部門)

(いしい かつや：名古屋大学情報連携基盤センター大規模計算支援環境研究部門)