



## 大規模計算におけるクラスタコンピューティングの可能性

—非経験的分子軌道計算の経験から—

長嶋雲兵 [物質工学工業技術研究所]

関口智嗣 [電子技術総合研究所]

### 広大な時空をこの目で —大規模計算機シミュレーション—

物質科学の時間的空間的広がりには広大で、また連続的である。空間的には原子・分子の $10^{-10}\text{m}$ の世界から $1\text{m}$ の我々人間のスケール。そして環境問題など地球スケールから $10^{10}\text{m}$ の銀河を越えた宇宙の果てまで。ほんの $1\text{cm}^3$ の水でさえ $10^{23}$ 個の水分子からなる。また化学反応の世界はフェムト ( $10^{-15}$ ) 秒の世界であり、宇宙空間での物質の生成/消滅は、 $10^{20}$ 秒のオーダーである。このような広大な時空の現象を、我々のスケールに拡大/縮小してみせてくれるのが、電子計算機を用いたシミュレーションである。

物質のマクロな性質には、物質の原子分子レベルのミクロな構造が強く反映する。原子分子レベルの現象は量子力学的であり、我々のスケールは古典力学的である。物質科学に限っていえば、ミクロな量子力学的世界と我々の世界の間を埋めるのは、統計力学や熱力学である。ただし、量子力学—統計力学—熱力学をそれぞれ正しく解いて、物質の手触りや光沢といった我々の五感で認識できる物質の性質を求めるのは至難の技である。とはいえ、量子力学の誕生以来、計算化学者が夢見てきたことは、計算機シミュレーションによって未知の分子の化学的性質を純理論的に予測し、さらに高機能を持つ新たな物質を作り出すことであった。

残念ながら、現状では扱える分子の大きさや種類という観点からはまだ十分とはいえませんが、非経験的分子軌道法や分子動力学などシミュレーション技法の充実と計算機の急速な進歩により、大学の研究機関だけではなく、創薬工業や化学工業など民間企業における新機能性物質開発研究の場においても、計算機シミュレーションは計算化学として着実に広まり始めている。これは、理論の改良や計算技法の発達によりさまざまな物理量の正確な計算が可能となったことと、1980年代の超大型計算機にも匹敵する計算能力を持つワークステーション (WS) の出現とインターネッ

トの普及による計算環境の改善に負うところが大きい。

非経験的分子軌道計算の非経験的という言葉の意味は、実験データのような経験的なパラメータをいっさい使わず、純理論的に計算だけで分子軌道を計算するという意味である。そのため、実験的に捕捉することの難しい化学反応における反応中間体の構造を明らかにして、反応メカニズムを明らかにしたり、星間分子などのように地球上では安定でない分子の物性を計算して、宇宙からの電波の解析を行うことなどに使われてきた。もともと計算機シミュレーションは計算機の中にあらゆる状況を作り上げることができるので、実験に比べ安全であり、また煩雑な実験の準備や手順を必要としない。従来は計算機資源 (CPU能力やメモリ容量など) の制限によって、取り扱える分子のサイズや計算できる物理量の精度などが十分ではなかったが、1990年代に入り計算機の能力の飛躍的向上を背景に、サイズや精度が向上したので、研究室レベルの研究だけでなく、化学工業や創薬工業の現場でも幅広く利用されるようになってきた。特に非経験的分子軌道法を用いた分子構造の精密計算は、X線構造解析などの実験の精度を超える。とはいえ、現状では現実の光合成系や各種薬・酵素反応系を考えるとまだまだ計算能力は足りない。このため、クラスタコンピューティングは計算科学の現場に強力な計算能力を供給するものとして大いに期待されている。

本稿では、物質開発の現場で広範囲に用いられている非経験的分子軌道法を例にとり、計算機の利用技術という観点からクラスタコンピューティングを考察する。

### アプリケーションから見た並列処理 —並列処理の長所・短所とその動向—

個々の演算器の性能向上は、半導体技術の成熟と共に困難になり始めている。最新の演算器では、1個の

命令をナノ秒のオーダーで実行するが、この間に光はわずかに30cmしか進むことができない。演算器の集積度をいかに高めて計算機を小型化するかが技術的課題として立ちはだかつてきている。その解決策として、演算器を並列に複数束ねることにより、高速性を実現したのが並列計算機である。ベクトル演算器を複数束ねた計算機がベクトル並列計算機、マイクロプロセッサを多数並べたシステムが超並列計算機システム (Massively Parallel Processor: MPP) である。いずれの形態においても、主記憶をCPU間で共有する共有メモリ型と各CPUにローカルメモリを持たせる分散メモリ型がある。クラスタコンピュータは分散メモリ型MPPである。

クラスタコンピュータのような分散メモリ型並列計算機のアプリケーションから見た特徴として、次の7項目がある。

- 1) 複数のCPUを同時に使うことができるため、1CPUに比べて時間あたりの計算能力が増大する。
- 2) メモリからのデータ読み出し速度が通常の計算では律速となるが、個々のプロセッサが同時にメモリにアクセスでき、実質的なメモリの読み出し能力がプロセッサ台数分だけ増大する。
- 3) 一般にプロセッサに搭載できるメモリ量は制限されているが、並列計算機では接続したプロセッサの台数に比例してメモリ空間が広がるため、多くのデータをメモリに格納することができる。
- 4) メモリに高速にアクセスするために、最近のプロセッサにはキャッシュメモリが用いられているが、単一のプロセッサではキャッシュメモリに収まらないデータでも、データの分散により処理するデータが小さくなり、各プロセッサのキャッシュメモリに収まり高速な処理が行える場合が多い。
- 5) 個々のプロセッサに直接ハードディスクが接続されている場合、それぞれのディスクへの独立なアクセスが可能になり、入出力の並列処理ができる。
- 6) 単純な逐次加算よりも、分散処理による部分的な加算では、計算精度が良くなる場合がある。
- 7) 廉価なマイクロプロセッサを多数用いることにより、高速計算システムを低コストで構築できる。

もちろん、並列計算機にも欠点はある。最終的な結果を得るためには、データを交換する通信をプロセッサ間で行わなくてはならない。この通信速度と各プロセッサに分散する数値処理の分量 (粒度) のバランスが悪いと、並列計算機のパフォーマンスを引き出すことはできない。粒度をできるだけ大きくするようにプログラムの構造を変えたり、通信の頻度を極力抑えるような工夫が必要であり、これがプログラムの並列化を困難にしている最大の理由である。ベクトル型スーパーコンピュータでの最大の成功は、コンパイラによる自動ベクトル化が比較的うまく働いた点にある。残念ながら現状では、過去に開発されたプログラムを自動並列化してくれるコンパイラは皆無ととっても過言ではない。

プロセッサ間通信を簡便に行うライブラリとして、PVM<sup>1)</sup> やMPI<sup>2)</sup> があるが、メッセージ通信を記述するプログラミングはいわば、アセンブラで並列プログラムを書くような面倒な作業であり、プログラム開発にとって大きな障害である。とはいえ、MPIの標準化により移植性の高いプログラムを開発できるようになりつつある。

分子軌道計算のみならず科学技術計算には、従来ベクトル型スーパーコンピュータが主として用いられてきたが、単一の計算機のパフォーマンスが限界に近づくにつれて、ベクトル計算機を数台から数百台結合したシステムが中心になっている。これらは、共同利用の大型計算機センタの軸マシンの地位を保持している。

計算機の動向で特筆すべき点は、マイクロプロセッサの急激な進歩である。世界的に有名で十分なデータを持つSpec Benchmark<sup>3)</sup> のSPECint92やSPECfp92などのデータを見るとおおむね、1年半ごとに性能が2倍になっている。また、単一プロセッサの実効性能をLinpack<sup>4)</sup> ベンチマークで見ると、一世代前のベクトル型プロセッサに匹敵する性能をマイクロプロセッサが達成しつつあることを示している。マイクロプロセッサは廉価でかつコンパクトなので、MPPのように数百から数千個のプロセッサを高速ネットワークで接続すれば、きわめて高い性能を実現できる。超高速計算機の実現には、開発コストの関係からクラスタコンピュータへの期待が高まっているが、数千にも及ぶプロセッサを効率よく稼働させる並列プログラムの開発が重要な課題となる。

### アプリケーションから見た クラスタコンピューティングの可能性 —過去・現在・未来—

#### ・過去：第一世代のクラスタコンピューティング

我々は、Thinking Machine Corp. CM2, Intel iPSC, nCUBE, MasParなど、超並列計算機の開発が華やかであった1992年あたりからクラスタコンピューティング技術に注目してきた。その大きな理由はスーパーコンピュータや超並列計算機がコスト的に見合わないことに加え、それらに特化したソフトウェアの可搬性に問題があったためである。特に非経験的分子軌道法のユーザはベクトル型スーパーコンピュータユーザからPCユーザまでの広がりがあり、コスト的な面もそうであるが、むしろソフトウェアの可搬性が大きな問題であった。

高価なスーパーコンピュータに匹敵する性能をいかに安く実現するか、またプログラムの可搬性をいかに高めるかの、1つの解答が、メッセージパッシングライブラリを用いたクラスタコンピューティングであった。我々が第一世代と呼ぶクラスタシステムは、休暇中や夜間の情報処理教室のWSや計算機センタの端末群など、ネットワーク上の遊休CPUを10Base5で並

表-1 クラスタシステム構築のためのメニュー

# CPU:	Intel Pentium, DEC Alpha, C21164/C212641, IBM Power-PC, SUN Sparc etc.
# Number of CPU:	16, 32 ..... 1024, .....
# Memory Size/CPU (MB):	128, 256, .....
# Network:	Giga Bit Ethernet, Myrinet, 100BaseT etc.
# OS:	Linux/Free BSD, Windows NT, Original etc.

列接続してクラスタを構成するものであった。それらは、ある条件では20から30台位のWSで、当時のスーパーコンピュータと同等かそれ以上の性能を実現することができた<sup>5)</sup>。そのため、この第一世代のクラスタ技術は、従来は高価であった科学技術計算のニーズに低コストで応えるシステムとして注目を集めた。これは、クラスタコンピューティング技術が、はるか先にある最先端の技術ではなく、今普通に手に入る安定した要素技術を使い、少し先に見えている技術を安価に、そして安定に実用化するものであったからである。

ソフトウェア的に見れば、この時代のクラスタコンピューティングの発展の理由の1つには、PVMやMPIなどのメッセージパッシングライブラリの急速な普及があげられるが、負荷分散など並列処理特有な問題に関しては、皮肉にも超並列計算機上で培われた技術の寄与が大きかった。

• 現在：第二世代のクラスタコンピューティング

第一世代のクラスタシステムは、もともと遊休の資源の利用を対象にしていたため、その上でのソフトウェア技術の開発などにとって必須な、実行時間などの信頼性の高いデータを得ることが難しかった。また、複数台の計算機の電源投入やシステム管理などを人手で行わねばならず、維持管理運営などに問題があった。そこで遊休の計算機資源を前提とするのではなく、専用の資源を持った並列システムを構築し、高い対価格性能比を維持しながら、大規模な問題を解くためのシステムが数多く作られている。第二世代のクラスタシステムとでもいふべきシステムであろう。もちろんこれらには、オートブート&シャットダウンなど維持管理運営のための機構が装備されている。また、大規模なシステムばかりでなく、個人や研究室レベルの利用のための比較的小さなサイズのクラスタシステムも多数作られている。さらにベストシステムズ社<sup>6)</sup>やビジュアルテクノロジー社<sup>7)</sup>のようにベンチャーとしてPCクラスタシステムを専門に制作販売する会社も現れてきた。分子科学の分野でも、サイエンティスツパラダイス社<sup>8)</sup>のように自社開発のPCクラスタシステムに、これも自社開発の分子軌道計算や分子動力学的並列処理プログラムをバンドルして販売する会社が現れている。これは、クラスタシステムの対価格性能比の向上もさることながら、MPIなどの普及でプログ

ラムの可搬性が向上したため、並列処理システムがある特定の限られた技術者のものでなくなり、その性能向上とあいまって、並列処理の実用的認知度が高まったためである。Spec BenchmarkのSPEC/HPG<sup>9)</sup>に採用されたGAMESSという非経験的分子軌道計算プログラムは、MPI版、PVM版、TCGMSG版を世界に無償で配布している。そのため世界中の多くのGAMESSユーザがPCクラスタを用いてクラスタコンピューティングを実用的に利用している。

• 未来：第三世代のクラスタコンピューティングへ向けて

クラスタコンピューティング技術は背景とする要素技術に依存するところが大きく、目的に沿った要素技術の選択が重要なポイントとなる。そのためクラスタコンピュータの特徴として、その構成の多様性がある。たとえば読者が今、クラスタを作る場合、表-1のようなメニューのうちからニーズに応じて基本的要素技術を選択することができる。

今後とも新しいCPUや通信機構などの提案が目白押しなので、このメニューの多様性は急速に増していくだろう。

今後のクラスタシステム構築に関しては、クラスタに特有な技術開発が重要である。たとえばプロセッサの状況を把握するライブ/パフォーマンスモニタ、稼働状況を把握し、自動負荷分散・スケジューリングを行う機構、OSのような、あたかもクラスタシステムを1つのシステムとして管理運営するためのツール群などである。さらにクラスタは汎用ネットワークを利用するが、そのため、バリア同期やブロードキャストデータ転送などに弱点がある。それを克服するための付加的なハードウェアの開発・実装が有効であると考えられる。効率のよい管理運用のために、リモートブート/シャットダウン機構は必須な機能であろう。

信頼性の向上に関する努力も必要である。我々のクラスタでは、ディスククラッシュやメモリの初期不良、バックプレーン能力不足によるネットワークスイッチのトラブルなどに手を焼いた。また、NFSによるファイルシステムを前提としているためアプリケーションによっては、I/Oコンフリクトが起き、それがパフォーマンスネックとなり大幅な性能低下を引き起こすことがあった。信頼性向上のためディスクレスまたは半導体ディスクの採用も今後の方向の1つである。アプリケーションソフトウェアのチューニングの際には、各CPUの環境や構成をジョブごとに常に一定とするツールの必要性も痛感した。将来的には大規模クラスタに対するネットワークブートの方式を確立する必要がある。さらに多数台によって構成される大規模なシステムへ拡張するには、設置面積や空調および電力の問題も重要である。たとえば設置面積の縮小のためには、Single Board Computer (SBC) の採用なども積極的に行うべきである。大規模システムであれ

表-2 クラスタコンピュータを用いた分子構造最適化計画

CPU	グリシン (110基底)		チオフェン (248基底)	
	wallclock	speedup	wallclock	speedup
1	30m10s	1.00	8h40m5s	1.00
2	14m20s	2.10	4h10m28s	2.08
4	10m47s	2.79	2h1m46s	4.27
8	8m12s	3.67	59m6s	8.80
16	8m42s	3.46	40m21s	12.88

表-3 用いたクラスタの諸元

CPU: Alpha21164A 533MHz, Cash: 1 <sup>st</sup> on-chip 8K, 2 <sup>nd</sup> on chip 96K, 3 <sup>rd</sup> on board 2MB, Memory/CPU: 128MB, Network: 100BaseT (Fast) Hub OS: Linux, Message Passing Lib.: MPI
--

ばあるほど、実装密度の向上が必要となる。

クラスタ技術によって高性能で大規模な並列処理システムが比較的安価で簡単に構築される時代がすぐそこにある。そこで期待されるのはクラスタコンピューティング技術の発展を基盤とした、ソフトウェア技術としての並列化技術の発達である。

くどいようだがクラスタコンピュータ上では、利用可能CPU数が増えれば単純に性能が増えるわけではなく、高い効率を得るためには、実アプリケーションプログラムの動作特性に合わせたクラスタ構成をとる必要がある。また逆に実アプリケーションプログラムを、実行させるクラスタコンピュータに最適化することも必要である。OpenMPやMPIなど、並列アプリケーションのコンパイラレベルでの世界標準に準拠しながら、並列処理対応ソフトウェアとライブラリの開発によって、さらに広範囲なアプリケーションをクラスタコンピュータ上で効率よく実行させていく環境の整備が望まれる。

### クラスタコンピューティングと 非経験的分子軌道計算

#### • 非経験的分子軌道計算のアルゴリズム

非経験的分子軌道計算は、計算自体の並列性が高くと高く、ベクトル計算機上で高い性能をあげた。そのプログラムの並列化は、1980年代のはじめに試みられている。その後MPPやWSクラスタの出現とともに並列処理の有効性が広く認識され、今日では、著名な分子軌道計算のプログラムはほとんど並列化されるに至っている。その基本的な考え方は、きわめて単純である。

分子軌道計算の手順は、まず、1) 分子構造と分子軌道を記述するための基底関数を初期データとして入力する。基底関数は、通常原子の電子状態を計算して求められる。

2) これを用いて分子軌道の初期値を近似的に作成し、次に近似的分子軌道を用いてFock行列を生成する。Fock行列の行列要素は以下のとおりである。

$$f_{ij} = \langle i | h | j \rangle + \sum P_{kl} (2 \langle ij | 1/r | kl \rangle - \langle ik | 1/r | jl \rangle)$$

第1項は電子と原子核の相互作用を表し、第2項は電子同士の相互作用を表す。第2項の和は $k > l$ に関してとられる。P<sub>kl</sub>は、近似分子軌道から作られる電子の分布を表現する密度行列であり、 $\langle ij | 1/r | kl \rangle$ と

$\langle ik | 1/r | jl \rangle$ は電子同士の斥力を表現している。インデックスi, j, k, lは基底関数を表す。この式からも分かるように、Fock行列の生成には、基底関数の数nの4乗に比例する演算量を必要とする。

3) このFock行列を対角化し固有値と固有ベクトルを得る。得られた固有ベクトルが分子軌道であり、固有値は分子軌道に属する電子の運動エネルギーとなる。

Fock行列は近似分子軌道を用いて作られるので、Fock行列を作る近似分子軌道とFock行列の対角化によって得られる分子軌道(固有ベクトル)が一致するまで、このプロセスは繰り返される。このプロセスをSCF (Self-Consistent Field) サイクルという。

さらに、分子構造最適化の場合は、分子構造を変えながら、このプロセスを繰り返す。分子構造の変更のためにエネルギー局面の原子核の位置に関する微分の計算が必要となるが、この部分も2電子積分の微分の計算となるので基底関数の数nの4乗に比例する演算量を必要とする。

#### • 非経験的分子軌道計算の並列処理

通常の計算では、計算時間の実に90%以上が2電子積分の計算とFock行列の生成に費やされているが、この部分が並列化されれば著しく計算時間が短縮できる。2電子積分は互いに独立なので、まず2電子積分の処理を複数のプロセッサに分散させる。各プロセッサでは、指定された2電子積分を計算し密度行列P<sub>kl</sub>を掛けて、部分的なFock行列を作製する。現状のプログラムの多くは、各プロセッサの処理がすべて終了した段階で、部分的Fock行列をマスタープロセッサに集め、完全なFock行列とし対角化するのである。各プロセッサで基底関数や原子核の座標などの共通データを、スレーブプロセッサに2電子積分の計算に先立って送信する。SCFサイクルに入ると、密度行列を同様に送信し、2電子積分の計算を開始する。並列処理が終了すると、マスタープロセッサに部分Fock行列を転送し完全なFock行列作成後対角化を行い、収斂するまでこの過程を繰り返す。

表-2にPCクラスタを用いた分子軌道計算による分子構造最適化計算のスピードアップの例を示す。分子はアミノ酸の一種であるグリシン (110基底) と有機伝導物質として注目されているチオフェン (248基底) のオリゴマーである。表-3に用いたPCクラスタの諸元を示した。

分子構造最適化計算はFock行列の生成と対角化の他に、原子核の位置に関するエネルギー微分の計算のために2電子積分の微分を計算する。初期構造の質にもよるが、原子核の数をNとすると大体N回の分子軌道計算が必要となる。分子構造最適化は、分子軌道計算以上に演算量を必要とするが、2電子積分の微分の計算も2電子積分の計算と同様に並列化の効果が非常に高い。この結果は、初期構造などによるので、分子を変えてもいつもこのような結果がでるわけではないが、クラスタコンピュータの実力を測る目安にはなる。

グリシンではCPU数が2の時わずかながらスーパーニアスピードアップが見られる。これは、並列化によって1CPUあたりの問題サイズが小さくなり、行列のほとんどが3次キャッシュに載ってしまったためと思われる。しかしながらCPU数16の方が8の場合より遅くなる。これは問題サイズが16台を使うには小さすぎて演算粒度と通信処理のウエイトが逆転していると考えられる。このように小さな問題では、通信性能も高い必要がある。

一方、チオフェンの場合は、問題サイズが大きいので16台程度でも著しい性能の低下は見られない。朝研究室に来て1時間ぐらいで入力を作り、その後ジョブを投入し、会議や講義をして、夕食後に結果が得られて、寝るまでに解析をするのと、ジョブ投入後コーヒーを飲みながらWebで新聞でも読んでいると答えが出て、昼ご飯前にもう1つジョブを投入できるのでどちらがよいのだろうか?? (会議や講義で忙しいシニアの研究者には前者がよいかもしれない。でもやっぱり早いに越したことはない)。

### おわりに

## —クラスタ・ネットワーク・コンピューティングへの期待—

現在の非経験的分子軌道計算は積分計算とFock行列の生成が並列処理されている。並列処理によってこれらのステップの高速化が実現されると、行列対角化が次の問題として浮かび上がってくる。行列対角化はベクトル計算機で効率よく実行されることが知られているので、積分計算とFock行列の生成はクラスタコンピュータで実行し、Fock行列の対角化はベクトル計算機でというように、異なるタイプの計算機をネットワークで接続した不均質なネットワークコンピューティング環境で、いかにして分子軌道計算を高速化するかが今後の課題となろう。残念ながら、今のところ不均質なネットワークコンピューティング環境で、分散並列処理を効率よく行う計算機利用技術は確立されていない。

不均質な環境での分散処理には、並列プログラミング、プロセス間通信、プログラムの構造化など克服すべき課題は多くあるものの、すでに我々のグループで

進められているNinf<sup>10)</sup> やGlobus<sup>11)</sup> などの広域分散並列処理システムの開発も急速にすすめられており、ソフトウェア技術の発展により不均質な環境での分散処理におけるさまざまな課題は、次第に解消されていくに違いない。ソフトウェアとハードウェア双方に対する最近のコンピュータ技術の発展をみると、このようなクラスタ技術を含むハードウェアの進歩とプログラムの改良の相補的な発展による計算時間の大幅な短縮と問題サイズの拡大が期待できる。これは、非経験的分子軌道法の適用範囲を拡大し、分子設計や創薬などにとどまらず、さらには新材料や新エネルギー原料などの創出に向けた実用的な大規模計算が、今以上にさまざまな場所で行われるようになることを予感させる。分子科学分野におけるクラスタ・ネットワーク・コンピューティング技術への期待は大きい。

クラスタコンピューティングの利点はユーザがそのニーズとコストに合ったシステムをいくつかの選択肢の中から基本要素を選択し、それらを容易に組み合わせて作り上げるといふ、いわばプラモデルを作るような楽しさがある。クラスタは、性能という点では必ずしもハードウェアの潜在的な能力を十分に引き出しているわけではない。やはりMPPや専用計算機システムには劣るが、その度合いがコストに見合うかどうかのポイントとなる。クラスタコンピュータの性能はコストパフォーマンスという視点から評価されるべきもので、従来の高性能計算機システムとは少し趣がちがう。我々も、第一世代のSUN/IPCによるクラスタやAlpha WSによる第二世代のWizクラスタ<sup>12)</sup>、その他のマイクロクラスタによる経験をもとに、より広いニーズに応えられるクラスタ技術の開発に貢献していくつもりである。

謝辞 電総研の中田秀基研究官、建部修見研究官、高木浩光研究官、東工大の松岡 聡助教授、小川宏高助手、合田憲人助手、新情報技術開発機構の佐藤三久主任研究員、NEC基礎研の高田俊和主任研究員、地球環境産業技術開発機構の山本博志主任研究員といろいろ情報交換をさせていただいた。ここに謝意を表する次第である。

### 参考文献

- 1) <http://www.epm.onrl.gov/pvm>
- 2) <http://www.mpi-forum.org/>
- 3) <http://www.specbench.org/>
- 4) <http://www.netlib.org/>
- 5) Nagashima, U., Hyugaji, S., Hosoya, H., Sekiguchi, S. and Satoh, M.: An Experience with Super-Linear Speedup Achieved by Parallel Computing on a Workstation Cluster: Parallel Calculation of Density of States of Large Scale Cyclic Polyacenes, *Parallel Computing*, Vol.21, pp.1491-1504 (1995).
- 6) <http://www.bestsystems.co.jp/>
- 7) <http://www.v-t.co.jp/>
- 8) <http://www.spara.co.jp/>
- 9) <http://www.specbench.org/hpg/>
- 10) <http://ninf.etl.go.jp/>
- 11) <http://www.globus.org/>
- 12) 益口摩紀, 長嶋雲兵, 関智嗣, 建部修見, 佐藤三久: アルファワークステーションのクラスタetlwizの性能評価, *IPJS SIG Notes*, Vol.98, 98-HPC-70 (1998).

(平成10年9月28日受付)

