

Windows を用いた PC Cluster 上における並列多倍長数値計算 ライブラリ MPIBNCpack の性能評価

幸谷智紀*

Performance Evaluation of MPIBNCpack on a PC cluster using Microsoft Windows

Tomonori Kouya*

1. 初めに

我々は GMP[1] 及びその基本関数を利用した MPFR[2] をベースに、小規模 PC cluster 向けの並列多倍長数値計算ライブラリ MPIBNCpack[6] を開発してきた。GMP/MPFR は長らく UNIX 環境でのみ利用されてきたが、今年 (2004 年) になって Visual C++.net と NASM を用いて、GMP 4.1.3 を Windows 環境に移植するためのソースキットが公開された [5]。我々はこれに基づいて最新の GMP 4.1.4 を Windows に移植し、cs-pcluster2 (Pentium IV 2.8GHz, Windows XP Pro SP2 と Vine Linux 2.6r4 が動作, 10PEs) の 1node を使ってベンチマークテストを行ってみた。結果、乗算のパフォーマンスはオリジナルと変わりなく、除算については短い桁数では最大 2 倍近く悪化しているものの、桁数を長くするにつれてオリジナルとの差が縮まってくる事が判明した (図 1)。

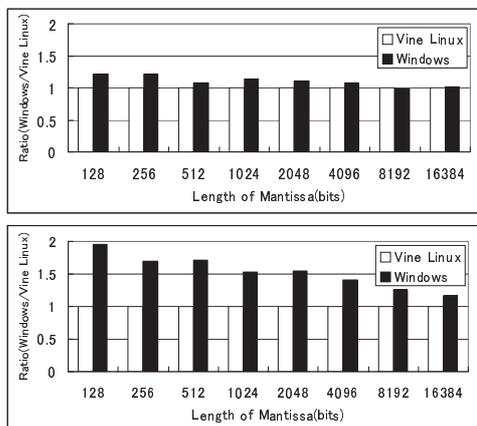


図 1: Windows 版 MPFR の多倍長乗算 (上) と除算 (下) の性能

また、Windows 環境でも利用できる MPICH2 Ver.1.0 が本年 11 月にリリースされている。これについても、NetPIPE[3] を使って MPICH 1.2.5 (Windows, Vine

Linux) との比較ベンチマークを行ったところ、Windows 環境においては MPICH 1.2.5, MPICH2 共、Linux よりも最大 100Mbps 以上通信性能が良くなる事が判明した。

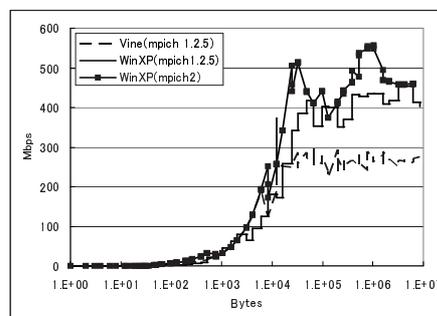


図 2: Linux と Windows における MPICH の通信性能

従って、Windows 版 GMP/MPFR と MPICH2 を用いて MPIBNCpack を動作させると、短い桁数で並列度の低い計算を実行すれば多少パフォーマンスは落ちるが、並列度を上げ、長い桁数で計算を行えば、Vine Linux cluster で実行するより良いパフォーマンスが得られると期待される。実際、同じ cs-pcluster2 を Vine Linux cluster 環境 (以下、Linux 環境) と Windows cluster (以下、Windows 環境) に切り替え、それぞれの環境で補外法を用いた数値積分と、積型 Krylov 部分空間法 (BiCG, CGS, BiCGSTAB, GPBiCG) の並列計算性能を比較したところ、予測通りの結果が得られた。本稿ではこれを示す。

2. MPIBNCpack の性能評価

補外法を用いた並列数値積分法と、積型 Krylov 部分空間法の結果を以下に示す。

2.1 補外法を用いた数値積分

我々は harmonic 数列を用いた補外法に基づく GBS アルゴリズムに対し、小規模 PC cluster 向けの改良を行ったものを提案した [7]。このアルゴリズムを用いて、Windows 環境と Linux 環境でベンチマークテスト

*静岡理科大学
Shizuoka Institute of Science and Technology

を行った結果を図3に示す。使用したのは Kahaner の 21 番目のテスト問題 (積分区間 $[0, 1]$, 被積分関数 $f(x) = \cosh^{-2} 10(x-0.2) + \cosh^{-4} 100(x-0.4) + \cosh^{-6} 1000(x-0.6)$) であり, 10 進 50 桁相当 (2 進 167bits) で計算を行った。

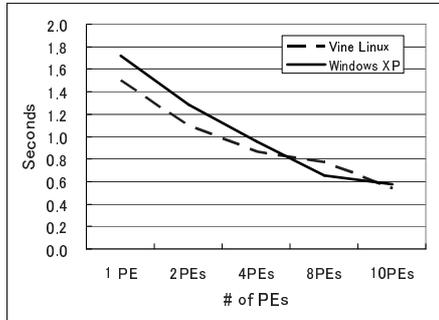


図 3: 並列数値積分法の性能 (10 進 50 桁相当)

我々のアルゴリズムでは, 並列度が上がると自動的に分割数も増加するため, 全体の通信時間も増加する。従って, PE 数が増えるにつれて, 通信性能の良い Windows 環境の方が有利になり, Linux 環境との差が近接してくることがわかる。

2.2 Krylov 部分空間法

Krylov 部分空間法は丸め誤差に対して非常に敏感で, 計算精度を増やすと反復回数が大きく変化することが知られている。今回は長谷川 [8] が取り上げた非対称行列に対応した積型解法, BiCG, CGS, BiCGSTAB, GPBiCG 法を用いてベンチマークテストを行う。使用したのは 150×150 の Toeplitz 行列 ($\gamma = 1.7$) であり, 真の解は $[0.1 \dots 149]^T$, 停止条件は $\|r_k\|_2 < 10^{-20} \|r_0\|_2 + 10^{-50}$ とし, 前処理は行っていない。ここでは 10 進 1000 桁相当 (2 進 3322bits) で計算した結果を示す (図 4(Linux 環境), 図 5(Windows 環境))。

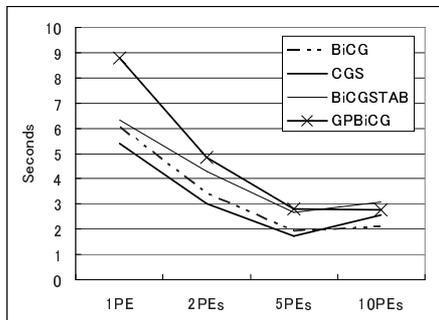


図 4: Linux 環境における積型解法の性能 (10 進 1000 桁相当)

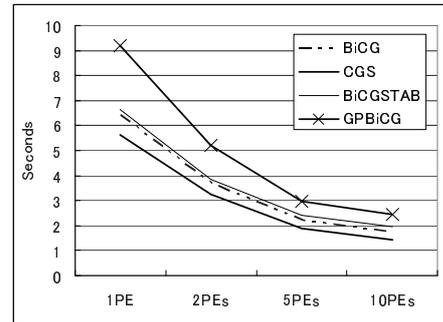


図 5: Windows 環境における積型解法の性能

反復回数は BiCG 法が 134 回, CGS 法が 111 回, BiCGSTAB 法が 117 回, GPBiCG 法が 113 回であり, これにそれぞれの解法の反復 1 回あたりの計算時間が乗されて全体の計算時間が決定される。桁数が多いため, 全体的にはかなり良好な並列性能が得られているが, PE 数を 10 にすると, 通信性能が勝る Windows 環境の方が, Linux 環境よりも全体の計算時間を減らしていることがわかる。

3. 結論と今後の課題

以上のベンチマークの結果, 予測通り, 桁数を増やし, 並列度を上げると, Windows 環境の方が若干パフォーマンスが向上することが判明した。今後の課題は, SMP, Multi-Core 向けに, MPIBNCpack に 1node 内での並列性を持たせるようにすることである。その際には今回の移植作業での経験を生かし, 最低でも Linux, Windows 双方に対応したものになるようにしたいと考えている。

謝辞

MPIBNCpack の Windows 環境への移植に動機を与えてくれた越須賀正雄・鈴木文明両卒研生に感謝する。また, GMP 及び MPFR の Windows 環境への優れた移植作業を行った Dr. Gladman, GMP/MPFR を開発し育ててくれた Free Software Community に感謝する。

参考文献

- [1] GNU Multiple Precision Library, <http://swox.com/gmp/>
- [2] MPFR, <http://www.mpfr.org/>
- [3] NetPIPE, <http://www.scl.ameslab.gov/netpipe/>
- [4] MPICH, <http://www-unix.mcs.anl.gov/mpi/mpich/>
- [5] GMP and MPFR for Win32, <http://fp.gladman.plus.com/computing/gmp4win.htm>
- [6] MPIBNCpack, <http://na-inet.jp/na/bnc/>
- [7] 幸谷智紀・鈴木千里, 補外法を用いた並列多倍長数値積分法の実装と性能評価, FIT2004, 2004.
- [8] 長谷川秀彦, 4 倍精度演算における積型反復解法の比較, LA 研究会, 2000.