PC Clusterの性能評価
～メモリ及びネットワーク帯域計測編～

幸谷智紀
tkouya@na-net.ornl.gov

平成 16 年 9 月 6 日

概要

本稿では，NetPIPE を用いて二つの PC cluster(“VTPCC”と “cs-pccluster2”) におけるメモリ転送性能 (memcpy 関数の転送性能)，及び TCP, MPI 転送性能を調査した結果について述べる。これら二つの PC cluster はどちらも各 node に x86-32 CPU を搭載した PC を用い，各 node は 1000BASE(GbE) によって結ばれたもので，OS には RPM 系 Linux distribution を利用している。但し，VTPCC は著者らの管理するものではなく，国立 N 大学 M 研究室 (高名なミステリ作家のいる研究室ではない) に設置されているもので，各 node は Dual Xeon プロセッサを搭載した SMP マシンとなっているのに対し，cs-pccluster2 は Pentium IV プロセッサを 1 個搭載した，著者らが組み立てた普通の PC によって構成されているという違いがある。この度，NetPIPE を用いてこの両者の通信性能を比較してみたところ，メモリ転送能力はそれ程差がないのに対し，cs-pccluster2 の TCP, MPI の転送性能は，VTPCC に比べて著しく劣ることが判明した。そこで，cs-pccluster2 には Windows(2003 と XP Pro SP2) を導入し，改めて通信性能を計測したところ，Vine Linux 環境よりも TCP, MPI 性能が向上することが確認できたものの，VTPCC に比べるとまだ劣るレベルであることが分かった。

1 初めに

我々は昨年 (2003 年) の 12 月に，2 代目となる PC Cluster である “cs-pccluster2”(図 1) を構築した [7]。これは次のようなハードウェア，ソフトウェア構成となっている。

CPU Pentium IV 2.8 GHz(with Hyper Threading)

RAM 512MB or 1024MB(cs-muse と cs-room443-05 のみ)

LAN 1000BASE-T(MPI 用)， 100BASE-TX(NIS/NFS 用)

Switch Dell PowerConnect 5212

OS Vine Linux 2.6r4

MPI mpich 1.2.5

cs-pccluster2 は研究兼教育用マシンとして，特段不満もなく様々な用途に活用されてきた。しかしベンチマークテストを行ってみると，ネットワークの性能に問題があることが判明した。

比較対象として，国立 N 大学 M 研究室に設置されている Dual Xeon Cluster，“VTPCC”を用いることで問

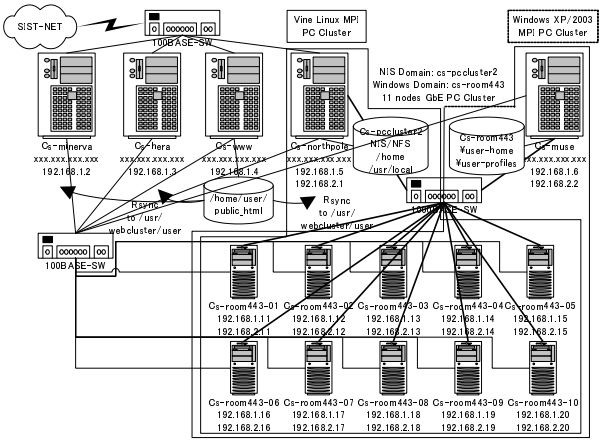

図 1: cs-pccluster2 のネットワーク配線図

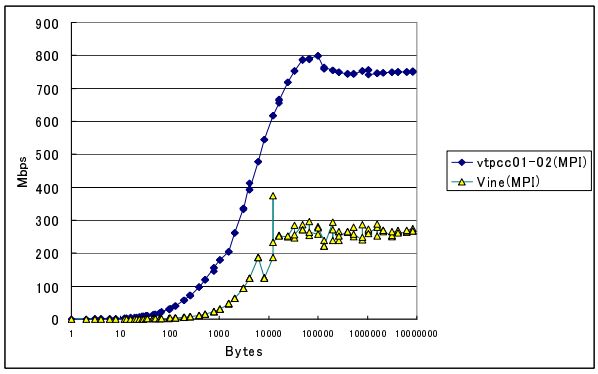

図 2: VTPCC と cs-pccluster2 における MPI 性能

題点が明確になった。VTPCC と cs-pccluster2 は，どちらも MPI には 1000BASE Ethernet(GbE) を使っており，RPM 系の Linux Distribution を導入している。しかし，NetPIPE[1] を用いて 2 node 間[1]の MPI 性能を計測してみると，VTPCC では 750Mbps(Mega bits/sec)〜800Mbps 程出ているのに対し，cs-pccluster2 では 250Mbps〜300Mbps 程度と，約 1/3 の性能しかないことが分かる (図 2)。

通信性能を向上させるため，特別にチューンアップした通信ライブラリを搭載した PC cluster 用の Linux Distribution として SCore[2] や Clustermatic[3] が存在していることは良く知られている。本来はそれらを用いて cs-pccluster2 を構築すべきであろうが，教育兼用のマシン群であるため，なるべく一般のアプリケーションが沢山同梱されている Distribution を使いたかったのである。それが裏目に出た結果，このような通信性能の劣化が現れたのであろう。

折角なので，この機会に cs-pccluster2 の方に Windows XP/Windows 2003 を用いた MPI PC cluster 環境をも構築し，Vine Linux 環境との比較検討を行うことにした。

2 NetPIPE について

NetPIPE[1] は同一マシン内におけるメモリ転送 (memcpy)，TCP, PVM, MPI などの様々な関数，プロトコルに対応した転送性能を計測するベンチマークテストツールである。ネットワーク転送性能を計測するツールは他にも存在するが，NetPIPE は，転送するバイト数を指数関数的に増加させ，各転送バイト数ごとに通信速度を Mbps(Mega bits/sec) で出力するという特徴を持っている。そのアルゴリズムは図 3 ようになっている。

```
T = MAXTIME
For i = 1 to NTRIALS
    t0 = Time()
    For j = 1 to NREPEAT
        if 自分が送信元であれば
            c byte のデータを送信
            c byte のデータを受信
        else
            c byte のデータを受信
            c byte のデータを送信
        endif
    endFor
    (略)
endFor
T = T / (2 * NREPEAT)
```

図 3: NetPIPE のアルゴリズム

指数関数的に c を増加させ，更にデフォルトではバイト数を c - 3, c, c + 3 と前後 3 バイトずらした場合も計測するようになっている。これにより，微妙なバイト数における cache ミスヒットや通信性能の劣化の検出が可能になっている。

今回は，VTPCC, cs-pccluster2 の Linux 環境においては，計測時において最新バージョンである 3.6.2 をソースコードからコンパイルして使用した。なお，Windows 版についてはバイナリの配布がなされていないようであったので[2]，memcpy(NPmemcpy)，TCP(NPtcp) については 3.6.2 のソースに著者がパッチを当てたもの[3]を，MPI(NPmpi) については mpich 1.2.5[4] に同梱されているものを，それぞれ Visual C++ 6.0 でコンパイルして使用した。

[1]VTPCC は 1node に 2cpu を積んでいるため，このベンチマークでは強制的に別 node との通信が発生するように実行している。

[2]配布元 [1] のサイトには一応 anonymous FTP へのリンクが張られていたのだが，計測時にはリンク先が確認出来なかった。

[3]パッチについては著者の Web ページ (http://na-inet.jp/) に公開してあるので詳細はそちらを参照されたい。

3 VTPCC(Xeon Dual Cluster) の計測

この Dual Xeon クラスタは自分の手元にあるものではないので，スペックの詳細は不明である。以下に把握できている部分のみ記しておく。

CPU Intel Xeon 2.8 GHz ×2

RAM 1024MB

LAN 1000BASE-T(MPI 用)， 100BASE-TX(NIS/NFS 用)

OS RedHat 8.0(kernel 2.4.20-18.8vt1smp)

MPI mpich 1.2.5

全部で 8 node(16 CPU) あるので，MPI では 16 process を使用することができる。

cs-pccluster2 の性能評価をするにあたり，この VTPCC の性能を比較対象として用いることにする。

3.1 メモリ帯域

まず，memcpy(NPmemcpy) の性能を計測する。Xeon, Pentium IV どちらも 1 次 cache, 2 次 cache を CPU 内に持つため，cache 内でのみデータ転送が行われる場合と，RAM とのデータ転送が発生する場合では通信速度 (この場合は転送速度というべきだが) は著しく異なることが予想される。

実際，オプションなしで NPmemcpy を実行すると，cache のミスヒットが発生している (らしい) 所では約 90Gbps から約 7Gbps へと激減していることがわかる (図 4)。前者の速度が cache メモリ転送速度，後者が RAM の転送速度であると予想できる。

NetPIPE にオプションを加えて cache のミスヒットが起こらないバイト数で実行した結果と，cache メモリの効果を無視した結果を図 5 に示す。これによって，前述したように，cache メモリを用いた時の転送速度と，RAM の転送速度予測が正しいことが分かる。

これらの結果から，例えば共有メモリ (OpenMP 等) を前提とした並列計算を行う場合，関数及び OS の API 呼び出しのオーバーヘッドを勘案すれば，cache 転送速度 (約 90Gbps) を得ることは難しく，むしろ RAM 転

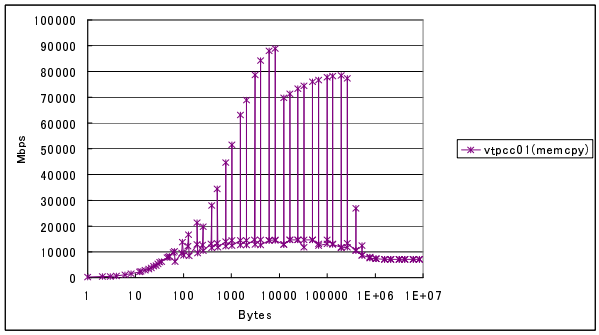

図 4: VTPCC01 における memcpy 性能 (netpipe オプションなし)

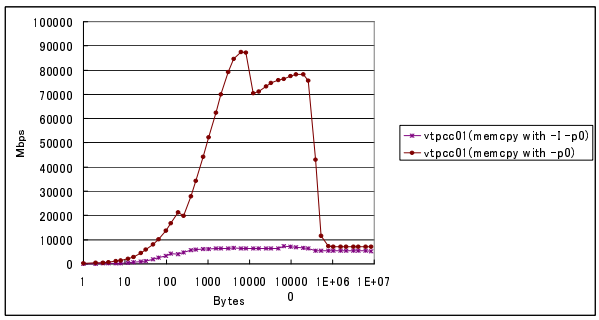

図 5: VTPCC01 における memcpy 性能 (cache 効果ありとなしの場合)

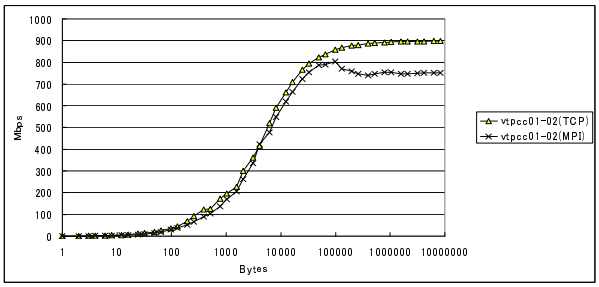

図 6: VTPCC01 と VTPCC02 間における TCP 及び MPI 性能

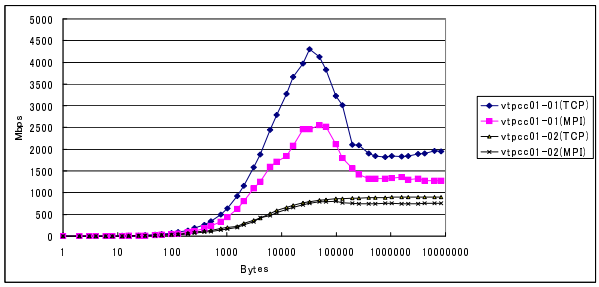

図 7: VTPCC01 内における TCP 及び MPI 性能

送速度 (約 7Gbps) 程度が最大転送速度と考えるのが妥当であろう。また，たとえ同一 node 内のプロセスであっても，TCP や MPI で通信を行う場合はこれより更に遅くなると考えられる。

3.2 TCP と MPI

では，GbE を介した別 node との転送速度はどの程度であろうか。本稿の冒頭で述べたように，MPI 転送速度 (MPI_Send/MPI_Recv 関数を使用) はかなり良好であることから，TCP 性能も GbE の最大性能に迫るものであることが予想される。これを裏付ける結果を図 6 に示す。

TCP 転送性能は最大で 900Mbps，最大 MPI 転送速度との差は 150Mbps 程度で，きれいな飽和曲線を描いていることが分かる。Dual Xeon という余裕のある CPU パワー，特別仕様の Switch(かどうかは不明) の効果，PM か GM といった通信ライブラリによるものであろうか？　後で示すように，我々のお手製 Cluster とは雲泥の違いがある。

では，同一 node 内で，つまり localhost に対して通信を行った場合の転送速度はどうなるか？　前述の予測通り，RAM 転送速度に比べて著しく劣ることが示された (図 7)。

最大転送速度だけ見ても，TCP は約 4.5Gbps, MPI に至っては 2.5Gbps しか出ていない。TCP の転送性能に対してはほぼ半減しているから，MPICH の実装はあまりよろしくない，ということなのであろう。tcplib[6] や MP_Lite[5] のような実装系が提唱されたのは，この MPICH のオーバーヘッドが契機となったようである。

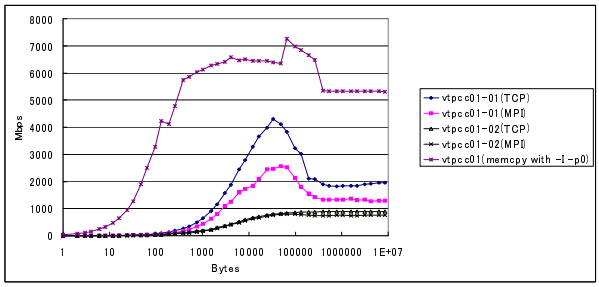

図 8: VTPCC における総合性能

3.3 総合評価

以上の結果をまとめたのが図 8 である。cache 転送速度はスケールが違いすぎるので除いてある。

この結果から分かるのは，MPIBNCpack のように，全ての通信を MPI で行っているライブラリでは，別 node 間通信が 800Mbps, 同一 node 内通信が 2.5Gbps となるのに対し，同一 node 内では通信を行わないようにすれば，少なくともこの部分の転送性能は 3 倍程度まで改善される可能性がある，ということである。

Dual CPU マシンが，通信性能に関しては著しくヘテロなことが，改めて良くわかるグラフである。

4 cs-pccluster2(Pentium IV Cluster) の計測

VTPCC と同様，cs-pccluster2 でもまずメモリ転送性能から計測し，次にネットワーク性能を計測することにする。Vine Linux と Windows との切り替えは，IDE HDD を差し替えることによって行っている。よって，両 OS において使用するネットワーク環境は全く同一

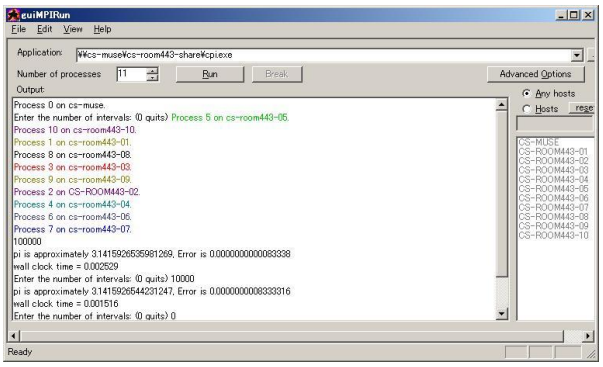

図 9: guiMPIrun による cpi.exe の起動

のものを用いていることになる。

参考までに，Windows を用いた場合の PC cluster 構築方法を簡単に記しておく。

1. 各 PC に Windows XP(SP2) をインストール
2. サーバ (cs-muse) に Windows Server 2003 をインストールし，Active Directory サーバをセット，Domain サーバとする
3. cs-muse 以外の各 PC を Domain に参加させる
4. mpich-1.2.5 の Windows 版を全 PC にインストール
5. guiMPIrun を用いて動作チェック (図 9)

なお，Vine Linux を用いての PC cluster 構築方法については著者の Web ページで公開してある文書 [7] を参照されたい。

4.1 メモリ帯域

まず cache 転送性能を見ることにする。cs-pccluster2 の場合，cs-muse とそれ以外のマシンとで，RAM 容量が異なり，前者が 512MB × 2 の構成，それ以外は 512MB × 1 となっている[4]。それが影響したのか，同じ Vine Linux 環境でも，cs-muse の方が 20Gbps 程度上がっている (図 10)。Windows XP, 2003 では殆ど同じ性能を発揮しており，これは VTPCC と比べて 10Gbps ほど低くなっている程度で収まっている。

では，cache の効果を無視して，RAM 転送速度のみを計測するとどうなるか? その結果を図 11 に示す。参考までに VTPCC の RAM 転送速度も記入してある。

[4]予算の関係上やむを得ず … 。

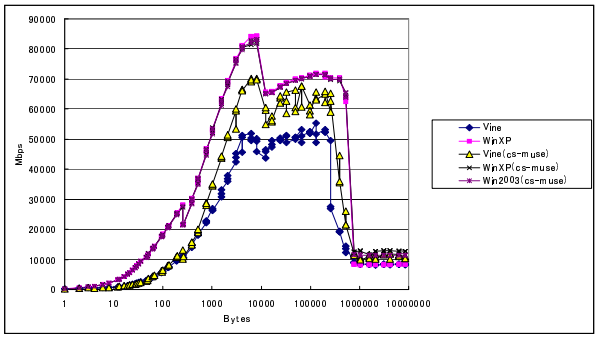

図 10: cs-pccluster2 におけるメモリ性能 (netpipe オプションなし)

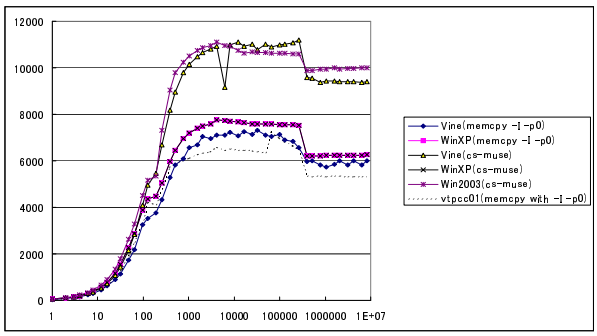

図 11: cs-pccluster2 におけるメモリ性能 (cache 効果なし)

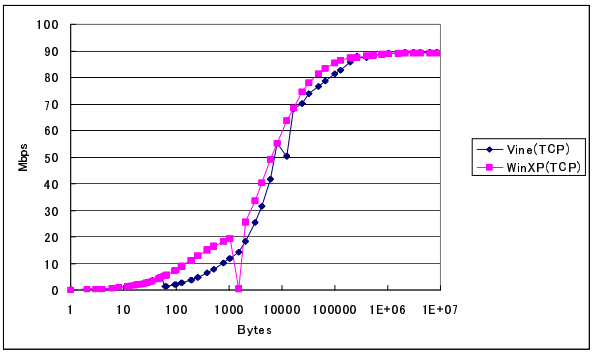

図 12: cs-pccluster2 における TCP 性能 (100BASE)

これで分かるのは，Windows XP と 2003 で差異があることである。それ以外では，Vine Linux, Windows とも大差なく，むしろ VTPCC よりも若干性能が良い。ということは，Windows 2003 と XP でメモリの扱いに違いがあるということなのか？ それとも Windows XP に SP2 を適用したことによるオーバーヘッドのせいなのだろうか？ Mother Board が異なるためか？　この辺りの理由は不明である。

4.2　TCP と MPI

本稿の冒頭に述べたように，cs-pccluster2 では GbE の性能，特に MPI を用いた場合は極端に悪いことが判明している。そこで Windows との性能比較をすることになったのだが，100BASE ではあまり優位な差は見られず (図 12)，どちらも 90Mbps 程度は出ていることが分かる。つまり，Vine Linux では GbE の性能が十分発揮されていないのではないか？という疑いが出てくる。

まず，GbE において Jumbo Frame を適用した場合と，そうでない場合との比較検討を行う。使用した Swicth の関係上，Jumbo Frame サイズは 4075bit に制限してある[5]。それぞれ Windows と Vine Linux で転送性能を計測した結果が図 13 である。

これによって，Windows 環境での性能の良さが明確となった。Jumbo Frame を用いても殆ど差は開かず，TCP では最大 800Mbps 程度の転送速度が確保できていることが分かる。

従って，MPI 性能も Windows 環境を用いることで

[5]もっと大きなサイズにも出来たのだが，Vine Linux ではこのサイズが最も性能が良かったのでそのようにしてある。

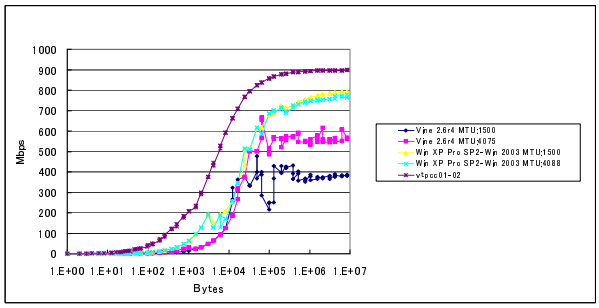

図 13: cs-pccluster2 における TCP 性能 (1000BASE)

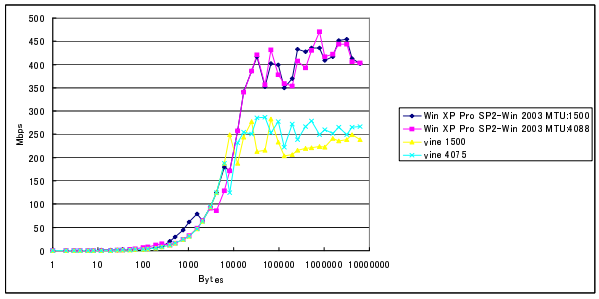

図 14: cs-pccluster2 における MPI 性能 (1000BASE)

向上することが期待される。その結果を図 14 に示す。

これにより，次のことが判明した。

- Vine Linux 環境では，GbE に Jumbo Frame を用いることで，TCP の性能向上を図ることが出来る。しかし，MPI ではあまり性能向上に寄与していない。MPICH のオーバーヘッドが，この TCP の性能向上分を打ち消してしまったためと思われる。
- Windows 環境では，Vine Linux 環境と比べて，TCP，MPI とも 200Mbps 以上の性能向上が見られる。

5　cs-pccluster2 はどこまで性能を向上できたか?

以上の結果より，cs-pccluster2 においても，Windows を導入することで何とか 400Mbps 程度の MPI 転送性能を得ることが出来ることが判明した。しかし，これとても VTPCC に比べると半分程度の能力しかない (図 15)。

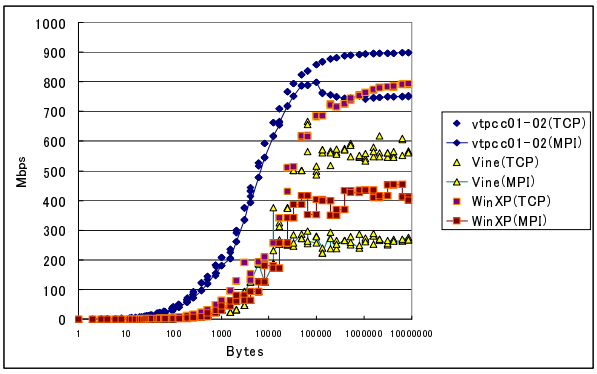

図 15: TCP 及び MPI 性能 (1000BASE)

何とか VTPCC の性能に肉薄できないものだろうか？

6　今後の課題

Vine Linux にとっては厳しい結果となったが，前述のように，Linux には PC cluster 向けの tune-up がなされた distribution があるのだから，本来はそちらを用いて計測するべきであったろう。また，MPICH によるオーバーヘッドの解消を目的とするライブラリも存在している。今後余裕があれば，それらを導入して性能比較を行ってみたい。

参考文献

[1] NetPIPE, http://www.scl.ameslab.gov/netpipe/

[2] SCore, http://www.pccluster.org/

[3] Clustermatic, http://www.clustermatic.org/

[4] mpich, http://www-unix.mcs.anl.gov/mpi/mpich/

[5] MP_Lite, http://www.scl.ameslab.gov/Projects/MP_Lite/

[6] TCPLIB,
http://grape.astron.s.u-tokyo.ac.jp/~
makino/softwares/tcplib/tcplib.html

[7] 幸谷智紀, Vine Linux による PC Cluster の構築 Version 2, http://na-inet.jp/na/mpipc2.pdf