GeoFEM-Cube/latency の T2K オープンスパコン（東大）512 ノードにおける実行結果（速報）

中 島 研 吾

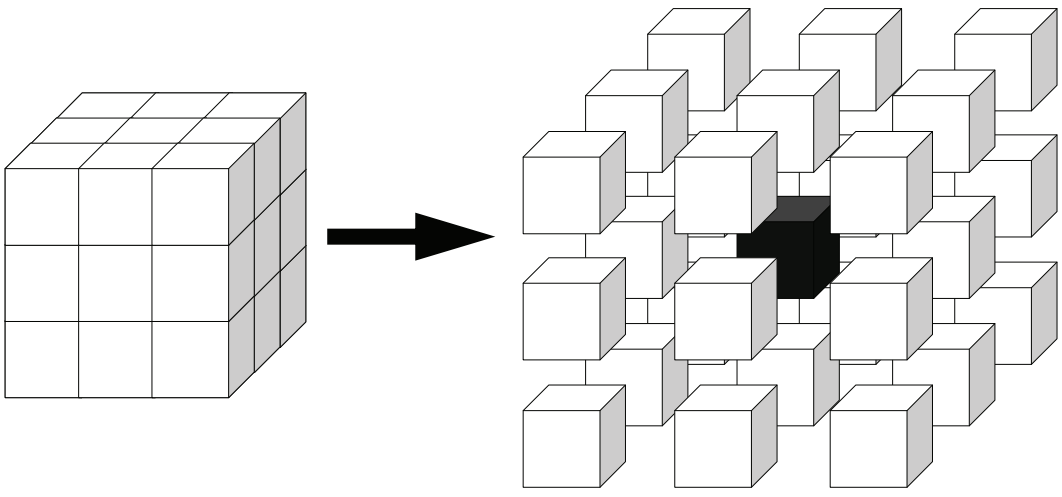
東京大学情報基盤センター

1. UT-HPC benchmark

東京大学情報基盤センターでは，2010 年 7 月より UT-HPC benchmark[1]を公開している。UT-HPC benchmark は実アプリケーションに基づき整備された大規模並列計算機システムの性能評価のためのベンチマーク群である。今回はそのうち，GeoFEM-Cube/latency を T2K オープンスパコン（東大）（Hitachi HA8000 クラスタシステム，以下 T2K（東大））512 ノード（8,192 コア）で実行した結果について報告する。

2. GeoFEM-Cube/latency

GeoFEM-Cube/latency は，固体地球シミュレーションのための並列有限要素法コード「GeoFEM[2]」の一部を抽出したもので，疎行列反復法の行列・ベクトル積計算において領域境界で発生する 1 対 1 通信をモデル化している。本ベンチマークでは図 1 に示すような立方体の集合において，各立方体と隣接する立方体間で大きさ 3～3×N^2 ワードの倍精度実数のメッセージの送信・受信を実施し，その時間を測定するものである。隣接する立方体の数は最大 26 であり（図 1 参照），隣接方法（面，辺，頂点）によってメッセージサイズが異なる。立方体位置により，隣接領域数も 7，11，17，26 と異なる。文献〔1〕で紹介した例では全ての隣接領域と 3×N^2 ワードの倍精度実数の通信を行うが，GeoFEM-Cube/latency では実際のアプリケーションと同じ通信パターンとなるように設定してある。



（a）立方体集合　　（b）立方体領域に隣接する立方体（最大 26 個）

図 1　GeoFEM-Cube/latency 計算モデルの概要

[1] http://www.cspp.cc.u-tokyo.ac.jp/ut-hpc-benchmark/
[2] http://geofem.tokyo.rist.or.jp/

GeoFEM-Cube/latency は図 2 に示すように，MPI_Isend，MPI_Irecv を使用した非ブロッキング通信に基づいている。N= 8，16，32，48 の場合ついて，自動的にデータを生成し，並列計算の実行が可能である。今回は T2K（東大）の 1～512 ノード，16～8,192 コアを使用して，16～8,192 プロセスの Flat MPI に基づき評価を実施した。プロセス間のメッセージ通信量の最大値は，それぞれ約 1.54 KB（N=8），6.14 KB（N=16），24.6 KB（N=32），55.3 KB（N=48）である。各 5,000 回（図 2 の ITERmax=5,000）の計算を実施して，一回の平均計算時間によって評価する。

```
      do ic  = 1, CASEtot
        N= 3*LENGTH(ic)*LENGTH(ic)

      do iter= 1, ITERMAX

      do neib= 1, NEIBPEtot
        call MPI_Isend (WS(N*(neib-1)+1), NN, MPI_DOUBLE_PRECISION,     &
     &                  NEIBPE(neib), 0, MPI_COMM_WORLD, req1(neib),    &
     &                  ierr)
      enddo

      do neib= 1, NEIBPEtot
        call MPI_Irecv (WR(N*(neib-1)+1), NN, MPI_DOUBLE_PRECISION,     &
     &                  NEIBPE(neib), 0, MPI_COMM_WORLD, req2(neib),    &
     &                  ierr)
      enddo

      call MPI_Waitall (NEIBPETOT, req2, sta2, ierr)
      call MPI_Waitall (NEIBPETOT, req1, sta1, ierr)

      enddo

      enddo
```

図 2　GeoFEM-Cube/latency の実行内容（主要部分）

3. Myrinet-10G リンク数の変更

T2K（東大）の各ノード間は Myrinet-10G（リンク当たり通信バンド幅：1.25 GB/sec）で結合されており[3]，本稿で使用されている「タイプ A」のクラスタでは，Myrinet-10G の 4 本のリンクから構成されている。「タイプ A」のクラスタではデフォルトでは 4 本のリンクを全て利用する（ノード間ピーク通信バンド幅：5.00 GB/sec）。また，環境変数「MX_BONDING」（デフォルト値＝4）を指定することにより，実行時にノード間通信で使用するリンク数を変更することができ，例えば「MX_BONDING=1」とすると，1 本のリンクしか使用しない。また，MX_BONDING の指定に関わらず，プロセス間通信量が 32KB を下回る場合のみ，自動的に 1 本のリンクのみ利用するように切り替わるような設定となっている（図 3 参照）。この判断にはある程度の時間を要するため，もしプロセス間通信量が 32KB

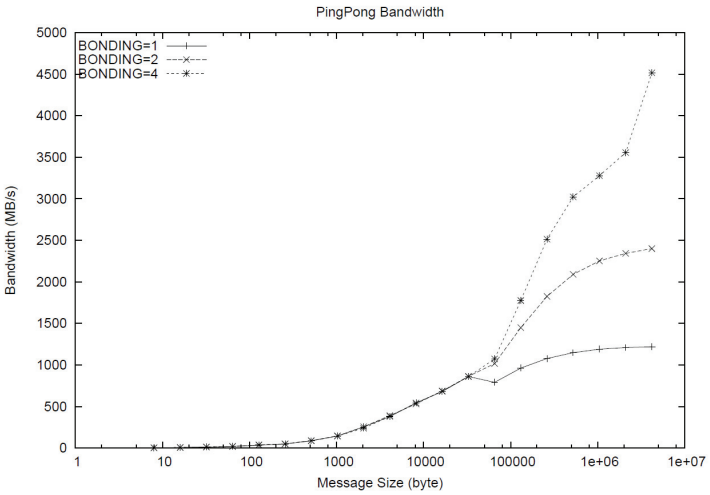


図 3　T2K（東大）ノード間実効通信バンド幅と Myrinet-10G リンク数（BONDING）の関係〔1〕
メッセージサイズが 32KB より小さい場合は
自動的に BONDING=1 となる
（松葉浩也博士（前東京大学情報基盤センター）による測定）

[3] http://www.cc.u-tokyo.ac.jp/service/ha8000/

を下回ることが予めわかっている場合には，MX_BONDING=1 としておいた方が効率が良い可能性がある。GeoFEM-Cube/latency の場合には N=32 から N=48 に変わる場合にプロセス間通信量（最大値）が 24.6KB から 55.3KB と増加し，32KB を超えている。

4. 計算結果

計算実施にあたっては，日立製作所製最適化 FORTRAN コンパイラ（オプション：-Oss –noparallel）を使用した。実行時の NUMA control としては，各ノードにおいて各ソケットのコアを 0 番から順番に埋め（図 4 参照），各ソケットのローカルメモリを使用する設定を使用した

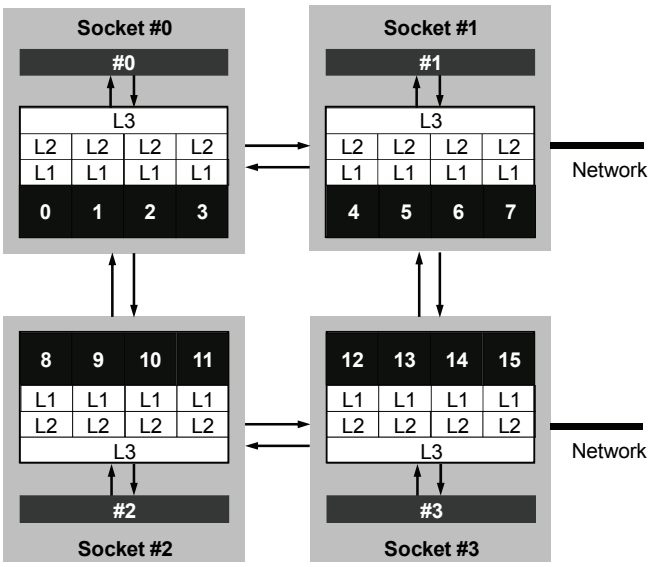
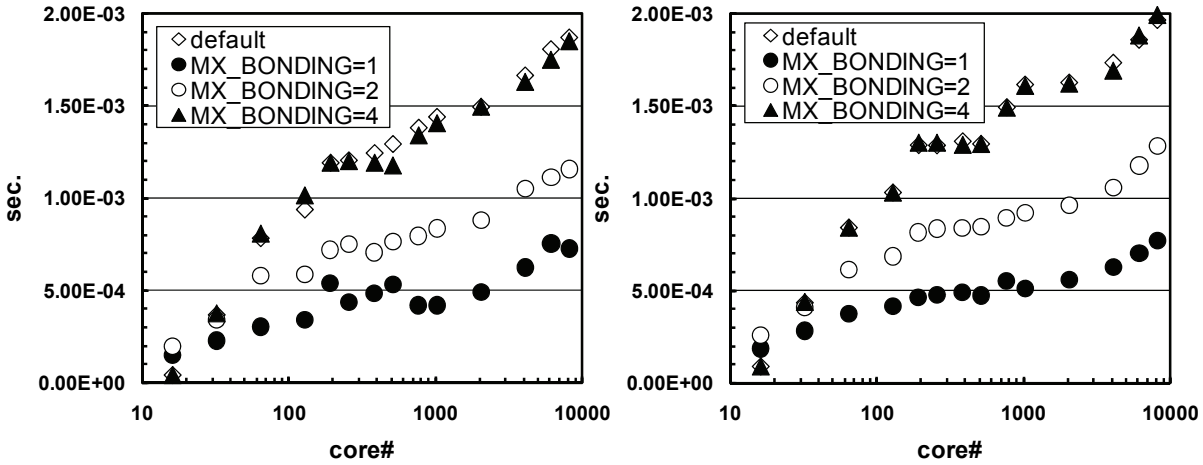


図 4　T2K（東大）のノード内構成

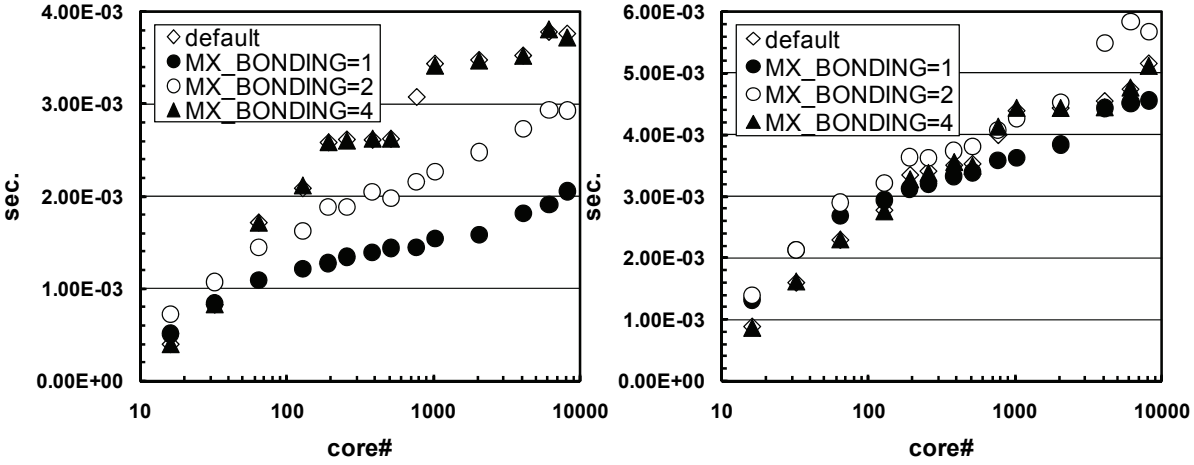


（a）N=8，プロセス間通信量：最大 1.54KB　　（b）N=16，プロセス間通信量：最大 6.14KB

図 5　GeoFEM-Cube/latency の結果（N=8，16），T2K（東大）1～512 ノード使用時

図 5 は N=8，N=16 の場合の測定結果である。通信量は N=16 は N=8 の約 4 倍であるが，実際の通信時間はほとんど変わらず，この測定結果はノード間リンク数設定のオーバーヘッド，および MPI プロセスの立ち上がり時のオーバーヘッド（レイテンシ）に相当しているものと考えられる。MX_BONDING=1 の場合が最も時間が短く，1,000 コアを超えるケースでは，

MX_BONDING=4 の場合と比較して 3 分の 1 程度の実行時間である。MX_BONDING の値を指定しない default の設定では MX_BONDING=4 となっているため，図 5 からもわかるように両者の実行時間は同じである。MX_BONDING=2 の場合は，MX_BONDING=1，MX_BONDING=4 の中間の値である。



（a）N=32，プロセス間通信量 : 最大 24.6KB　　（b）N=48，プロセス間通信量 : 最大 55.3KB

図 6　GeoFEM-Cube/latency の結果（N=32，48），T2K（東大）1～512 ノード使用時

図 6 は N=32，N=48 の場合の測定結果である。通信量の増加と共に実行時間も全体的に増加しているが，512 ノード（8,192 コア）においては N=48 の場合でも実行時間は N=8 の場合の 3 倍から 5 倍程度であり，最大通信量（約 35 倍）ほどの増加ではなく，ノード間リンク数設定，レイテンシによるオーバーヘッドの影響が大きいことがわかる。

N=48 では各ケースの処理時間はほぼ同じであるが，MX_BONDING=1 の場合がやや速い。これは，隣接 26 領域のうち，6 領域との通信量は 55.3KB であるが，他の 20 領域のうち 12 領域とは約 1KB（3×N ワード），8 領域とは 3 ワードの通信であるためである。

基本的な傾向は文献〔1〕で紹介したものと変わらないが，ノード数が増加するにつれて，ノード間リンク数設定，レイテンシによるオーバーヘッドの影響がより顕著になっていることがわかる。特にコア当たりの問題規模が小さい場合は注意が必要である。このような場合には，MX_BONDING=1 とした方がデフォルト設定（MX_BONDING=4 または 2）よりも性能が高い場合があるため，事前にベンチマークによってアプリケーションの挙動を把握しておくことが重要である。

参　考　文　献

[1] 中島研吾（2009）T2Kオープンスパコン（東大）チューニング連載講座番外編：Hybrid並列プログラミングモデルの評価（Ⅱ），スーパーコンピューティングニュース（東京大学情報基盤センター）11-5
http://www.cc.u-tokyo.ac.jp/publication/news/VOL11/No5/200909tuning.pdf/