# 2P134　T2K Tsukuba システムにおける GAMESS の性能評価

([1] 筑波大 計算科学研究センター, [2] 自然科学研究機構 分子研)
○梅田宏明 [1,2], 佐藤三久 [1]

## 序

近年大規模並列計算機システムのプラットフォームとして、マルチコア・マルチソケット構成の並列計算機システムが広く使われるようになってきている。このような構成の計算機では NUMA (Non-Uniform Memory Access)と呼ばれる非対称な共有メモリ型マルチプロセッサアーキテクチャが利用されている。これまで広く利用されてきた対称型マルチプロセッサアーキテクチャ(所謂 SMP, Symmetric Multiprocessing)とは共有メモリ型である点で同じであるが、各ソケットに対しメモリが配置されているため全てのメモリアクセスが必ずしも等価にならない点で大きな違いがある。このようなアーキテクチャ上での並列計算では、これまでとは違う配慮が必要となっており、OpenMP/MPI ハイブリッド並列化など様々な研究がなされてきている。計算化学の分野の代表的な分子軌道計算プログラムの一つである GAMESS [1]では、データサーバを持つなど単純なフラットな MPI プログラムとは異なった並列プログラムとなっており、NUMA システム上での動作は不明な点も多く、このようなシステム上で高速に動作させるためのノウハウが必要とされている。本発表では、NUMA システムである T2K Tsukuba システム[2]上で GAMESS の性能評価と解析を行なう。

## GAMESS の並列化

GAMESS は並列化については、独自の通信レイヤーである Distributed Data Interface (DDI)を用いたプロセスレベルの並列化が行なわれている。DDI では、broadcast や global sum 等の集合通信や send/recv のような一対一の通信の他に GET/PUT/ACC といったリモートメモリアクセス(RMA)機構が実装されている(version 1)。またサブグループの概念が導入されフラグメント分子軌道法計算で利用されている(version 2)。最近では、マルチプロセッサノードの普及に対応してノード内共有メモリを利用する機構が実装され、Coupled Cluster 計算においてより大きな計算をオンメモリで高速に計算できるようになってきている(version 3)。現在 GAMESS と共に配布されている DDI には TCP/IP socket, MPI-1/socket mixed, MPI-1, ARMCI [3]等による実装が用意されており、様々な計算機システム上で効率的な並列計算が可能である。sockets, mixed, mpi 等の実装では、RMA を実現するために計算プロセスとは別にデータサーバプロセスが必要となる(図 1)。このためプロセッサコアあたり 2 プロセスが動くことになり、データサーバへのアクセスが多い計算では性能に影響を与える可能性がある。一方 ARMCI を利用した場合はノードにつき一つのデータサーバスレッドが起動される。

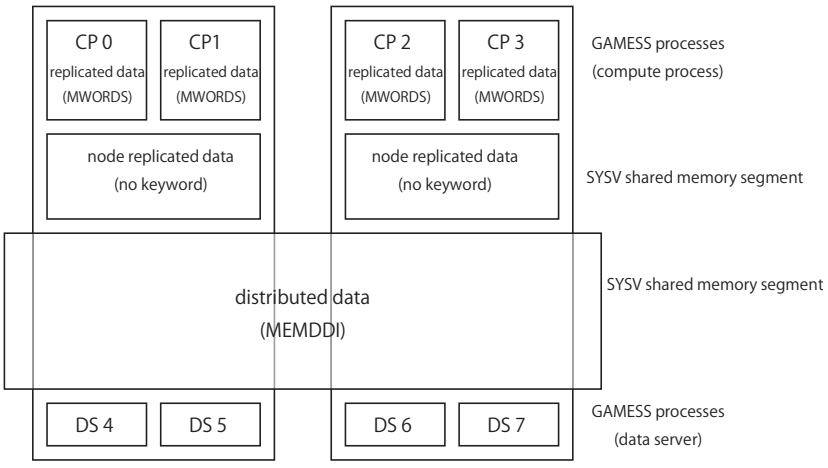


図 1 標準的な DDI の構成図(readme.ddi より)

データサーバは主に動的負荷分散におけるグローバルカウンタと積分変換における中間データの保持のために利用されている。また、それぞれの利用法に対する代表的な計算手法としては HF 計算と MP2 計算があげられる。特に通信量が大きいのは後者であるため、本発表では主として MP2 計算をターゲットとして性能評価を行なった。

## T2K Tsukuba システム

T2K Tsukuba システムは、各ノードに 4 つの Quad core AMD Opteron プロセッサ(16 プロセッサコア)を搭載した並列計算機システムであり、32GByte のメモリを共有している。個々のノードは 4 本の InfiniBand による fat-tree ネットワーク構造により接続されており、どのノードに対しても常に高速な通信が期待できる。ノード内に計算用のディスク領域やスワップ領域を持たないが、InfiniBand で接続された Lustre ファイルシステムにより高速な I/O を実現している。InfiniBand ネットワークを利用する方法として、mvapich1 等の MPI ライブラリの他に IP over InfiniBand (IPoIB)がある。

NUMA システムにおいては、メモリ及びプロセスをどのコア又はソケットに割り当てるかが重要なファクターとなってくる。これをコントロールするために Portable Linux Processor Affinity (PLPA) [4]を DDI ライブラリに組み込み、プログラム起動時に設定するよう実装した。

## テスト計算および結果

性能評価のためのベンチマーク計算として MP2 計算を実行した。計算には T2K Tsukuba システムの 4 ノード(64 プロセッサコア)を用い、分散共有メモリを使うアルゴリズムを利用している。表 1 にその結果を示す。processor affinity (PA)とは、プロセスをどのプロセッサコア上で動かすかを指定するもので、CORE は特定のプロセッサコアにバインドする設定、ANY はバインドしない設定に相当する。PA について、計算プロセス(CP)とデータサーバ(DS)それぞれに対し設定を行なった。太字になっているのは、それぞれの DDI 実装においてデフォルトで設定される PA 設定である。この結果は processor affinity が性能に対し非常に大きな影響を与えており、適切にこれを設定することが重要であることを示している。また ARMCI による実装では、データサーバがノードごとに一つだけしか起動されないことで計算プロセスに与える影響が比較的小さいことが予想され、このため他の実装に対し有利になったと考えられる。

表 1 T2K Tsukuba システム 4 ノードによる MP2 テスト計算の実行時間

| DDI | **socket** | socket | socket | mixed | mixed | **mixed** | mpi | mpi | **mpi** | **armci** |
|---|---|---|---|---|---|---|---|---|---|---|
| Network | **IPoIB** | IPoIB | IPoIB | mvapich1 | mvapich1 | **mvapich1** | mvapich1 | mvapich1 | **mvapich1** | **mvapich2** |
| PA(CP) | **ANY** | CORE | CORE | ANY | CORE | **CORE** | ANY | CORE | **CORE** | **CORE** |
| PA(DS) | **ANY** | ANY | CORE | ANY | ANY | **CORE** | ANY | ANY | **CORE** | **CORE(0)** |
| CPU/s | **72.4** | 67.5 | 72.8 | 82.8 | 87.4 | **127.0** | 78.6 | 73.4 | **77.6** | **97.6** |
| WALL/s | **108.5** | 99.5 | 240.0 | 108.8 | 122.2 | **273.6** | 113.7 | 107.3 | **240.9** | **99.2** |


[1] GAMESS (General Atomic and Molecular Electronic Structure System), http://www.msg.chem.iastate.edu/gamess/

[2] ARMCI (Aggregate Remote Copy Interface), http://www.emsl.pnl.gov/docs/parsoft/armci/

[3] T2K Open Supercomputer Alliance, http://www.open-supercomputer.org/

[4] PLPA (Portable Linux Processor Affinity), http://www.open-mpi.org/projects/plpa/