不均一クラスタ上での並列Linpackの性能に関する検討

指導教官：市川 周一　　学籍番号 003101 岸本 芳典

1 はじめに

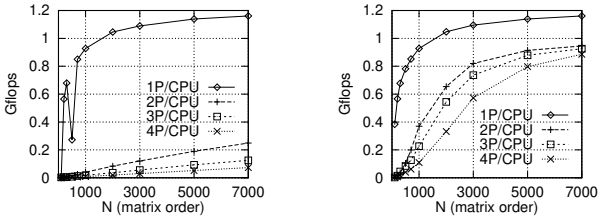
多くの科学技術計算アプリケーションでは，行列などの線型代数演算の高速化は極めて重要である．線型代数演算のうち LU 分解の性能測定には，Linpack ベンチマークが広く用いられている．特にクラスタなどの分散メモリ型並列計算機には，これを MPI(Message Passing Interface) を用いて並列化した HPL(High Performance Linpack Benchmark)[1] が多く利用される．

クラスタは高速ネットワークで要素計算機 (PE) を結合した仮想の並列計算機で，システム要素が均一の PE による構成が一般的である．HPL はこれを前提として各 PE に負荷を均等に分割するが，不均一な PE を持つクラスタでは負荷の不均衡によって性能上の問題を生ずる．

図 2(a) は，不均一なクラスタ上で HPL を実行した結果である．HPL(N=10000) で各 PE の単体性能を測定した場合，Athlon 1.33GHz (1.15 Gflops) は PentiumII 400MHz (0.27 Gflops) の約 4 倍高速となる．ところが，Athlon × 1 + PentiumII × 4 の性能は PentiumII × 5 の性能に等しい．これは，高速な Athlon は演算を先に終了するが，アイドル状態になって PentiumII の処理の終了を待つためである．

2 複数プロセスによる最適化

この問題を避けるには各 PE の速度に応じた負荷を与えればよい．笹生ら [2] は HPL を修正して負荷の均衡を図ったが，その方法では応用 (HPL) や実装に依存してしまう．高速な PE に複数の MPI プロセスを起動する方法ならば，均一なクラスタ用に書かれた他の並列応用にも容易に適用できる．ただし同一 PE 上に複数プロセスを起動するとオーバヘッドで性能が低下するため，笹生らは性能低下が著しいとして棄却している [2]．



(a) MPICH-1.2.1　　(b) MPICH-1.2.2

図 1: 複数プロセス起動時の性能

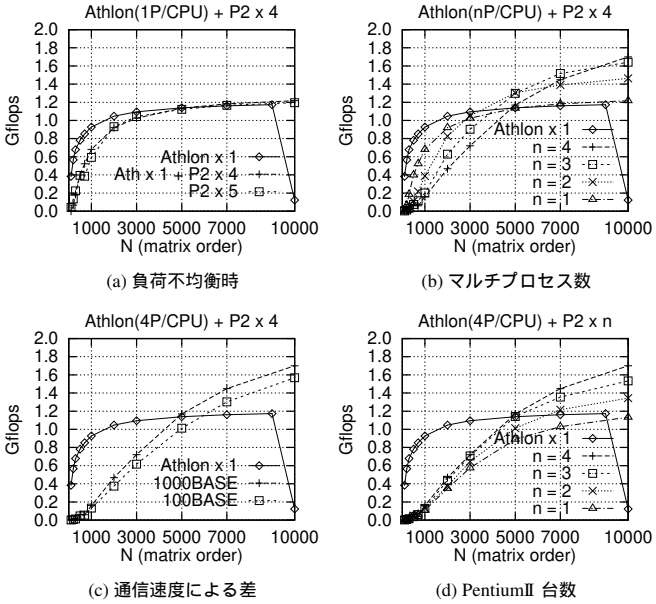
図 1 に，Athlon 上に 1 ~ 4 プロセスを起動した場合の性能を示す．MPI の実装としては MPICH を選んだ．MPICH-1.2.1 では複数プロセス起動によって 8 ~ 9 割近い性能低下が見られるが，1.2.2 では 2 ~ 3 割の低下にとどまっており実用に耐え得る性能を示す．

3 最適な実行方法とは

不均一クラスタで HPL を実行する場合，可能な構成の中から最良の選択肢を発見する必要がある．高速な PE のみで構成するか，低速な PE も使用するか，さらに高速な PE 上ではプロセスをいくつ起動するのか，等である．

そこで，これらの選択肢が性能に及ぼす影響を調べた．測定は，Athlon 1.33GHz × 1 と PentiumII 400MHz × 4 を 1000baseSX で結合したクラスタで行った．

図 2(b) は，Athlon 上のプロセス数と性能の関係である．行列のサイズ N によって最適なプロセス数は異なっている．例えば，$N \leq 3000$ では Athlon 1 台だけが良く，$3000 \leq N \leq 5000$ では PentiumII × 4 も利用し Athlon 上に 2 プロセス，$5000 \leq N \leq 9000$ では同じく 3 プロセス起動するのが良い．しかし，このような切り替え点 (サイズ N) は一定でなく，ネットワークや計算機構成に依存する．例えばネットワークが 100baseTX なら切り替え点の N は大きくなり (図 2(c))，PentiumII の利用可能台数が 4 より小さい時も切り替え点の N は大きくなる (図 2(d))．



(a) 負荷不均衡時　(b) マルチプロセス数

(c) 通信速度による差　(d) PentiumII 台数

図 2: ヘテロクラスタの各パラメータにおける性能

4 おわりに

HPL の複数プロセス実行により負荷不均衡が改善されることが確認された．また，起動プロセス数・PE 構成の最良の選択肢は問題サイズ N によって定まり，その関係は通信速度や使用可能な PE 数により変化することが判明した．

不均一なクラスタの性能はシステム構成の様々な要素を含むため，近似アルゴリズムによる性能の予測は難しい．

今後は，(1) クラスタの性能を自動的に測定し，(2) 近似モデルを用いて選択肢の切り替え点の予測を行ない，(3) 最適な実行環境を自動的に選ぶ手法の開発に取り組んでいく．

参考文献

[1] A. Petitet et al. “HPL – A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers,” http://www.netlib.org/benchmark/hpl/.

[2] 笹生 健, 松岡 聡, 建部 修見, “ヘテロなクラスタ環境における並列 LINPACK の最適化,” 情処研報 2001–HPC–86, pp.49–54 (2001).