

WAN 上のクラスタを用いた並列有限要素解析の性能予測

Predictability on the performance of parallel FEM using clusters on WAN

林 雅江¹ 奥田 洋司^{1,2}

Masae Hayashi¹ and Hiroshi Okuda^{1,2}

¹ 東京大学人工物工学研究センター

² 東京大学工学系研究科システム創成学専攻

¹Research into Artifacts, Center for Engineering, The University of Tokyo

²Dept. of Systems Innovation, Graduate School of Engineering, The University of Tokyo

Abstract: With the rapid growth of WAN infrastructures and development of Grid middleware, the cluster-of-clusters has become a realistic methodology for executing computation-demanding applications. While distributed computing or loosely connected applications has been successfully ported to the Grid environment, few tightly connected applications such as parallel finite element analysis (FEA) have been attempted. In this paper, we focus on an iterative solver, which is often used in FEA: the conjugate gradient method. By using both predictions and numerical experiments we evaluate the performance of the CG method parallelized via domain decomposition.

序論

有限要素解析は、社会基盤・産業基盤分野における有力な設計ツールの一つである。そうした基盤分野のひとつである原子力工学においては、非常に高い安全性が求められる上、実試験が困難なため、数値シミュレーションによる代替実験の期待が高い[1,2]。実大・実環境シミュレーションがのぞまれる原子力プラントでは約 10^5 以上のコンポーネントから構成され、まるごとモデル化しようとするれば数百 TB のメモリが必要になる[2]。そうした膨大な計算需要に対してグリッドコンピューティング[3,4]の利用が期待されてきた。グリッドコンピューティングは、ネットワーク上の様々な情報・計算リソースを動的に結びつけ、仮想的な統合環境として安全に利用しようとする情報基盤技術である。しかし、並列有限要素解析のように通信・同期頻度の高い、いわゆる密結合アプリケーションにおいては、流体構造解析や解析と可視化の連携アプリケーションといったように複数のプログラムから構成され、個々のプログラムをネットワーク上の異なる計算資源に割り当てるようなプログラミングスタイルがグリッド利用の主流にあり [5]、一つの大規模計算を複数の計算資源で実行するメタコンピューティングとしての利用は実績が少ない[6]。

本研究では、一つの大規模有限要素解析のためのグリッド利用を主眼とし、近年の WAN 環境の性能向上と PC クラスタの普及によってその実用性が期待される Cluster-of-Clusters 環境（以下 C-of-C）に注目し、クラスタ間で最適なプロセス数を分配するのに役立つ性能予測手法を提案し予測手法の評価ならびに並列 FEA のための C-of-C 利用に関して実用性評価を行う。

Cluster-of-Clusters における並列有限要素解析の性能予測

C-of-C という通信コストの高い環境で並列有限要素解析という同期および通信の頻度が高い並列計算を実行することになるため、計算コストに対し、高い通信コストが懸念される。並列有限要素解析において計算コストと通信コストのバランスは並列手法となる領域分割で得られる分散メッシュに大きく依存する。実用的な利用には適切な分割数の設定が不可欠であり、分散メッシュから得られる情報を基に計算コストと通信コストを予測する手法について提案する。

想定する実行環境と実行方法

想定する実行環境は実際に利用できる 2 台のクラ





スタで結ばれる C-of-C 環境である。2 台のクラスタはインターネットで結ばれ、一台は産業技術総合研究所（茨城県つくば市）のクラスタマシンで F32 と呼ばれるクラスタマシン(128PE, 64node, Intel Xeon 3.06GHz, 4GB Memory/node, Gigabit Ethernet, Redhat Linux8.0)であり、もう一台は東京大学柏キャンパス（千葉県柏市）にある SC と呼ばれるクラスタマシン (32PE, 32node, Intel Pentium4 3.00GHz, 2GB Memory/node, Gigabit Ethernet, Debian GNU Linux3.1)である。地理的には約 25Km で追跡可能な数だけで traceroute コマンドによる中継点は 15 箇所あった。

次に、実行の条件であるが、領域分割に基づく並列化を行う並列有限要素解析において、1 領域を 1 プロセスと扱い、1 プロセスを 1 プロセッサに割り当てたこととした。また、全体のプロセス数のうち半数ずつ 2 台のクラスタに分配することを原則とする。

例題

例題として用いるのは表 1 に示す、バイクの部位（エンジン、フレーム）やドリル、ポンプの 4 種類のテストモデルを用いた弾性静解析である。節点規模は最小のフレームで約 52 万節点（約 160 万自由度）、最大のポンプで約 3,600 万節点（約 1 億 1,000 万節点）と幅広い問題を扱う。並列計算の際のプロセス数モデルのサイズに合わせて設定しており、表の最下行にて示す。

表 1 解析モデル

	Frame	Engine	Drill	Pump
No. of Nodes (d.o.f)	520,000 1,600,000	580,000 1,730,000	1,700,000 5,100,000	36,700,000 110,100,000
Element type	2 nd order tetrahedron	2 nd order tetrahedron	1 st order tetrahedron	2 nd order tetrahedron
Model image				
No. of procs.	2,4,6,8,16,32	2,4,6,8,16,32	4,6,8,16,32,64	64,128

たとえば、フレームは 2,4,8,16,32 の 4 通りの並列プロセス数による計算を行うとし、ポンプでは 64, 128 プロセスでの並列計算を想定するものとする。並列有限要素解析では、内計算における集団通信と行列ベクトル積における隣接プロセス間での一対一通信があるが、一対一通信の通信相手となる隣接プロセスの数は単に分割数だけでなくその形状にも大きく依存する。そのことを示すのが図 1 である。図 1 では、各モデルが、並列計算される分割数（フレームなら 2,4,8,16,32）で有限要素メッシュを領域分割したときに 1 プロセスに割り当てられる剛性マトリックスの非ゼロ成分の数と隣接プロセスの数を示し

ている。各マーカは各モデルに対応しており、各プロット近傍にてモデルの頭文字と分割数を表示する。例えば、フレームの方がエンジンではおなじ分割数において非ゼロ成分の数はほぼ同等であるのに対してより隣接プロセス数に差があることがわかる。

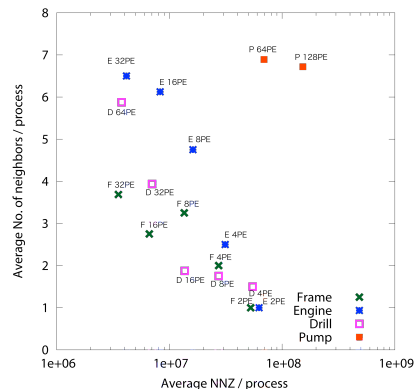


図 1 1 プロセス当たりの隣接プロセス数と係数行列の非ゼロ成分の数

予測手法

利用する並列有限要素解析プログラムは領域分割に基づく既存のものであり本研究では構造解析ソフトウェアの FrontSTR[7]を利用することを想定し、そのなかで行列方程式の反復解法部分（共役勾配法）に費やされる通信時間と計算時間について予測する。

通信時間の予測

領域分割に基づく並列化が行われるため、各プロセスもつ剛性行列や右辺ベクトルは、割り当てられた部分領域に関する節点に関してのみ持つことになる。したがって、行列ベクトル積や内積計算で通信が必要となる。テストアプリケーションである FrontSTR においては、内積は全プロセス間で MPI_DOUBLE_PRECISION 一つのデータサイズが集団通信の MPI_Allreduce（以下 Allreduce と呼ぶ）によって行われ、行列ベクトル積では隣接プロセス間で共有する節点に関するデータの送受信が一対一通信の MPI_Isend, MPI_Irecv, MPI_Waitall（以下行列ベクトル積一回当たりには生じる隣接プロセス間の MPI_Isend, MPI_Irecv, MPI_Waitall を一まとめにして Isend/Irecv と呼ぶ）で行われる。これら典型的な二つの通信に対して通信単体を実験環境で実行し、その結果を予測に用いることにする。このとき、クラスタ間では常に等しい数のプロセスを割り当てることにする。通信実験の結果を図 2,3 に示す。ここで、内積に関する Allreduce では通信データが 8 B 固定であることから通信に参加するプロセス数のみを変化させている。一方、行列ベクトル積に関する Isend/Irecv では、通信データサイズは領域分割に

おける隣接領域間の袖領域に依存するため、通信実験において一対一通信を同時に行うプロセスの数とデータサイズの両方を変化させて実験を行った。各図ではクラスタ間通信に加えクラスタ内通信の結果を比較のため載せている。

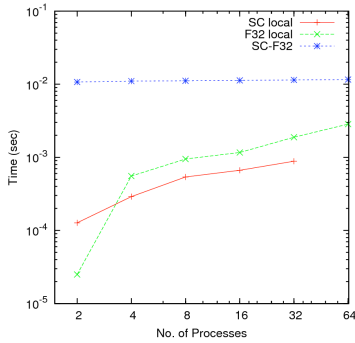


図 2 Allreduce による通信実験の結果

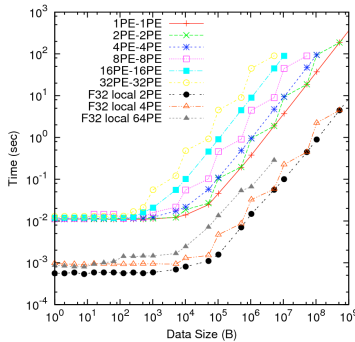


図 3 Isend/Irecv による通信実験の結果

利用する解析コード内の CG 法アルゴリズムにおいて一反復あたりの内積は 3 回、行列ベクトル積は 2 回行われる。内積計算に関わる通信時間をまず見積もる。Allreduce の実験結果 (図 2) を見ると、クラスタ間通信に注目すれば、2 プロセスでの通信時間は 0.0109[s] であり、64 プロセスでの通信時間は 0.0113[s] であったことからプロセス数の違いに対する通信時間の差は小さいことがわかる。内積一回当たりの通信時間を 0.0113[s] として一反復あたりの通信時間を 0.034[s] と見積もる。一方の Isend/Irecv にかかる通信時間予測では、各モデル、各分割数によって通信回数やデータサイズが異なるため、各ケースで得られる分割メッシュを基に算出する。その際、図 3 で得たデータ量対通信時間のグラフを線形近似し、切片 α をレイテンシ、傾き β をバンド幅の逆数とし、 n [B] のデータサイズに対する通信時間を $\alpha + \beta n$ [s] で算出する。これを全通信相手に対して総和を取

ることで Isend/Irecv の通信時間を見積もる。また行列ベクトル積は 2 回あるので行列ベクトル積に関する通信時間はここで算出された通信時間を 2 倍したものとし、Allreduce の通信時間との合計によって通信時間全体を見積もられる。

計算時間の予測

CG 法アルゴリズムを構成する主要な計算は前処理、行列ベクトル積、内積であるがなかでも、疎行列を扱う有限要素法においては非ゼロ成分のみを記憶する CRS (Compressed row storage) 形式などを利用するため行列ベクトル積では間接メモリ参照が頻発することから、通信コストのほぼ全体を占める。Vuduc からは疎行列ベクトル積 (以下 SpMV) においてピークパフォーマンスに対してどの程度の性能が得られるかについてモデル化している [8]。そのモデルを用い、本実験環境においては、SpMV 一回あたりの計算時間 T_{SpMV} を (1) 式で見積もる [9]。

$$T_{SpMV} = \frac{NNZ \times 2}{P \times 5\%} \quad [s] \quad (1)$$

ここで、 P [flop/s]、 NNZ はそれぞれマシンのピークパフォーマンスと係数行列の非ゼロ成分数である。さらに、一反復あたりの計算コストにすると、2 回の行列ベクトル積に加え、前処理の計算コストが加わる。前処理には局所 ILU(0) 法とその安定化のための Additive Schwartz 領域分割法が使われており、行列ベクトル積、前進代入、後退代入の計算コストと同等のものに解釈することができる。そこで、前進後退代入を行列ベクトル積 1 回分とみなし、一反復あたりの計算量は行列ベクトル積 3 回分として見積もる。

性能評価実験

前節で述べた性能予測手法を評価するため、実際に計算を実行したときの通信時間と計算時間と比較する。

実行方法

FrontSTR は MPI (Message Passing Interface) を用いて並列化が成されている。今回 WAN 環境での MPI プログラムの実行となるため、WAN を含むグリッド環境での MPI プログラムの実行を可能にする MPI 実装系の一つである GridMPI を利用した。MPI ライブラリとしてこうしたグリッド環境に拡張された MPI の実装系を利用することで既存の MPI プログラムがグリッド環境で修正することなく実行可能とな

る。

実験結果

図4にフレームからポンプまでの4つのテストモデルに対して計測した通信時間および計算時間を示す。また同じグラフに予測から得られた通信時間および計算時間も示す。ポンプ以外の3つのモデルにおいては、計算時間と通信時間の傾向をよくとらえていることがわかる。計算時間についてまず見てみると、プロセス数が小さいときに計算時間の予測が計測結果を大きく下回るケースがポンプ以外の3つのケースで見られた。このずれは、(1)式における計算性能を全てのケースにおいてピークの5%と一定に設定していることが原因と考えられ、更なる調整が必要である。また、ポンプの計算時間については予測が大幅に上回ってしまっており、特に1プロセス当たりのメモリ容量が大きい場合の計算時間の予測手法については更なる考察が必要である。一方、通信時間についてはフレームからポンプまで全4つのモデルにおいてほぼ等しいオーダーで予測できている。ただし、エンジンの32プロセスやドリルの64プロセスなどプロセス数が大きく設定された場合に生じる通信時間の急激な増加が予測できていないことがわかる。

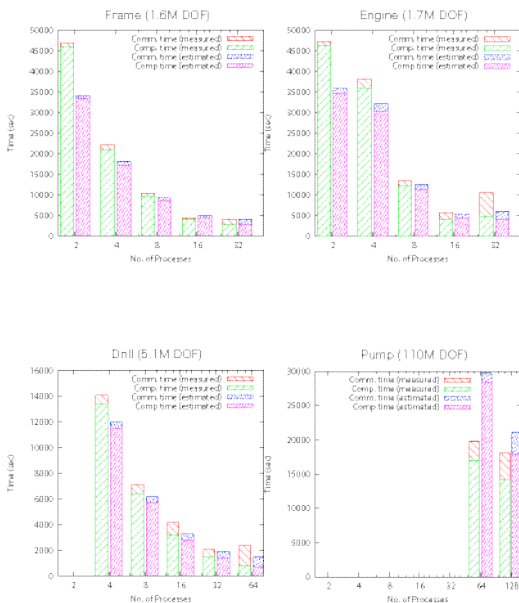


図4 各モデルにおける数値実験結果と予測値の比較

結論

並列有限要素解析の実行環境としてC-of-C環境を有効に利用するためにはあらかじめ最適な負荷分散を設定することが重要であり、本予測手法によって

分散メッシュから得られる情報と基礎的な通信性能実験から計算コストおよび通信コストの傾向が予測可能であることを示した。更なる実用性向上にはこうした性能予測手法の充実が求められる。また、本予測手法は2台のクラスタ環境で評価されたが、3サイト以上に増えた場合にも拡張可能なものである。より現実的かつ実用的な環境としてサイト数がより大きな環境を見据えた予測手法の評価、改善が必要であり今後の課題とする。

謝辞

本研究は、ApGridの研究活動の一環として行われました。産総研グリッド技術研究センターをはじめApGridに参加される全ての団体・研究者に感謝の意をここに表します。

参考文献

- [1] 吉村忍, CREST チームによる原子力発電プラントの地震耐力予測シミュレーション, 第19回CCSEワークショップ, 2008.
- [2] 西田明美, 松原仁, 田栄, 羽間収, 鈴木喜雄, 新谷文将, 中島憲宏, 谷正之, 近藤誠, 原子力プラントのための3次元振動仮想振動台の構築, 日本原子力学会和文論文誌, vol.6, No.3, pp376-382, 2007.
- [3] I. Foster et al., "The GRID: Blueprint for a New Computing Infrastructure", Morgan Kaufmann Publishers, 1998.
- [4] I. Foster and N. Karonis, "A Grid-Enabled MPI: Message Passing in Heterogeneous Distributed Computing Systems", In Proc SC'98, 1998.
- [5] Y. Suzuki et al., Development of Three-dimensional Virtual Plant Vibration Simulator on Grid Computing Environment ITBL-IS/AEGIS, Journal of Power and Energy System JSME, Vol. 3, No. 1, pp.60-71, 2009.
- [6] M. Muraoka and H. Okuda, Feasibility Study of Parallel Finite Element Analysis on Cluster-of-Clusters, Journal of Computational Science and Technology, Vol.3, No.1, pp.77-88, 2009.
- [7] H. Okuda, Middleware for Developing Parallel Finite Element Applications, International Conference on Computational Methods (ICCM2007), Conference Abstracts, pp. 240-240, 2007.
- [8] R.W. Vuduc. Automatic performance of sparse matrix kernels. PhD thesis, UC Berkeley, Computer Science Division, 2003.
- [9] M. Muraoka and H. Okuda, Feasibility study and predictability on the performance of parallel FEM using Cluster-of-clusters on WAN, Journal of Computational Science and Technology, Vol.3, No.2, pp.460-475, 2009.