

解析におけるHPC (ハイパフォーマンスコンピューティング) 技術

🔗 解析, 並列計算, 計算機, ネットワーク, 有限要素法

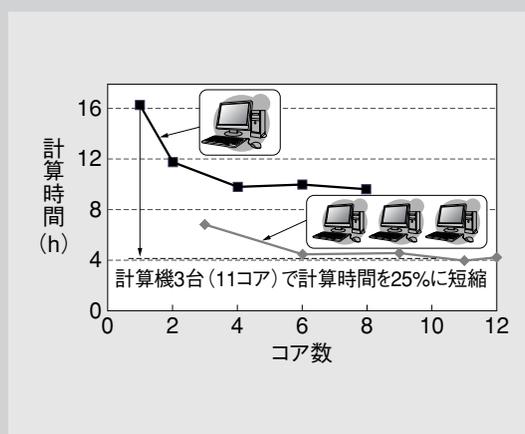
* 江尻光良 Mitsuyoshi Ejiri * 佐々木聖夏 Kiyoka Sasaki

概要

構造・流体解析により仮想的に検証を行い, 試作回数を減らす取り組みが進み, 解析計算の高性能化(高速化・大容量化)が求められている。

HPC (High Performance Computing) 技術とは解析対象の問題を複数の小さな問題に分割し, 複数の計算機を用いて並列に計算を行う技術であり, 高性能な計算が可能となる。

効率的に高性能化を行うためには分散構成を解析問題に応じて変える必要がある。問題のタイプやサイズや分散構成を変えて計算時間を計測した。HPCの効果に差があるが, 計算時間がおおよそ数分の一に短縮されることが確認された。



HPC技術による計算時間短縮

1. ま え が き

製品には高性能と低環境負荷の両面が求められるため, 多岐にわたるこれらの課題を開発時に十分考慮する必要がある。このため, 開発の上流段階において解析シミュレーション技術を駆使して, 設計の最適化を進めている。これらの実現には, 3次元CADデータを用いて形状を正確に模擬した大規模モデルや非線形問題, 複数の検討課題の最適化などに対応する必要があるため, 計算量が膨大となっている。

一方, 計算を行うソフトウェアとハードウェアについても, オペレーティングシステム (OS) が32bitから64bit化したことや, 演算装置 (CPU) のマルチ化やマルチコア (計算ユニットであるコアを複数持つ) CPUの普及により進歩している。

本稿では計算を複数のCPUやコアに分散・並列

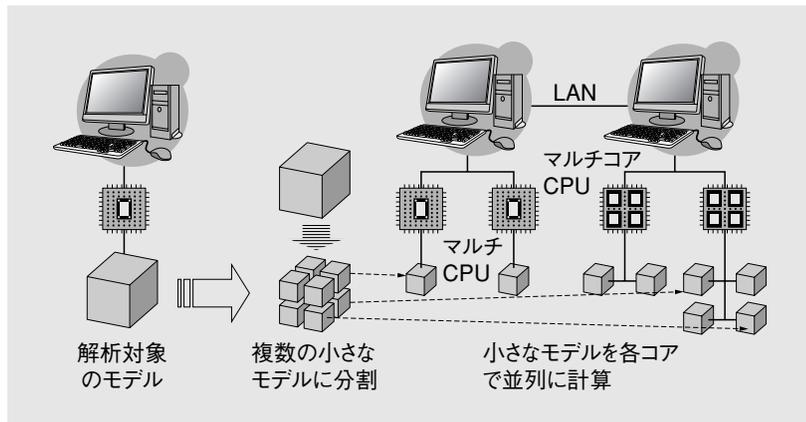
化して, 高速・大規模化を実現するHPC (High Performance Computing) 技術とその効果について紹介する。

2. 解析の大規模化・高速化手法

HPC技術とは, 第1図に示すように解析モデルを複数の小さなモデルに分割し, マルチコアCPUや複数のCPUを搭載した1台の計算機, あるいはネットワークに接続された複数台の計算機を用いて分散したモデルを並列に計算する技術である。これにより高速且つ大容量という高性能な計算を行うことができる。この代表的な方式は第2図に示すようなSMP (Symmetric Multi Processing) とMPI (Message Passing Interface) がある。

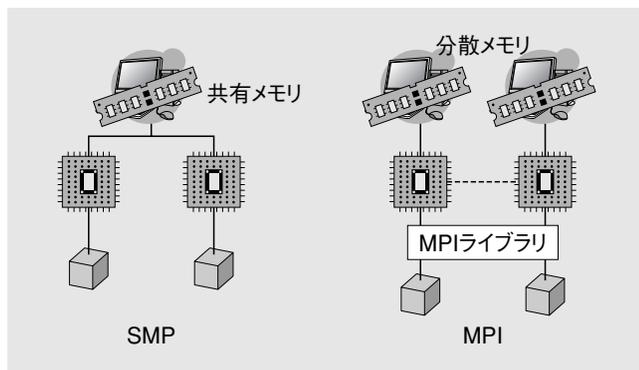
SMPはメモリを共有し, すべてのコアに対して対称的・均一的に処理が割り付けられる並列処理方式である。一般的に2~4コアで最大効率となる

*基盤技術開発部



第1図 分散処理

HPC技術における分散処理のイメージを示す。解析モデルを分割し各CPUコアに割り当て、並列に計算することにより計算時間を短縮する。



第2図 SMPとMPI

HPCの2つの方式であるSMPとMPIの違いを示す。SMPはメモリを共有して計算処理を行うが、MPIはコアごとにメモリを独立して使用し、メッセージ通信でデータのやり取りを行う。

サイズS サイズX

モデルサイズ	S	M	L	X
自由度	19万	38万	67万	147万

(a) 固有値解析・線形応力解析用の例題

モデルサイズ	L
自由度	73万

(b) 非線形構造解析用の例題

第3図 検証用の例題

検証に用いた解析例題のイメージと問題のサイズ(自由度)を示す。(a)の固有値解析(PCG LANCZOS法)及び線形応力解析のモデルはモータのフレームであり、サイズをS, M, L, Xで表記する。(b)の非線形構造解析のモデルは電機であり、大変形・弾塑性を解析する。

と言われている¹⁾。

MPIは分散メモリ型の並列計算でデータを通信するための仕様標準である。複数のコアがメッセージ通信を行うことにより協調して計算処理を行うもので、MPICH2などのプログラムライブラ

リが提供されている。基本的にはインターコネクトと呼ばれるネットワークを介してネットワーク分散MPIとして動作するが、1台の計算機内の複数のコアでローカルMPIとして動作させることもできる。ローカルMPIは同一計算機のメモリを分割して使用するため小規模な計算に適しており、メッセージ通信が計算機内で行われるためデータ転送速度も速い。

一方、ネットワーク分散MPIは各計算機のメモリを利用するため大規模なモデルに適しているが、ネットワークを経由してデータ転送を行うため通信速度の影響を大きく受ける。インターコネクトの代表的なものとしては、Myrinet (10Gbps)・Quadrics (7.2Gbps)・InfiniBand (6.6Gbps)などが提供されているが、低速ではあるがコストパフォーマンスの良さからGigabit Ethernet (1Gbps)が使われることも多い。

モデルを分割する方法は、モデル形状の部位によって分割する領域分割と、数値計算のマトリックス(代数式)を分割するマトリクス分割がある。構造解析では後者のマトリクス分割が使われることが多い。

3. 並列計算についての基礎検討

基礎検討は汎用有限要素法ソフトウェアのANSYSに並列計算機能を付加するMechanicalHPCを追加して行った。本ソフトウェアはSMP及びMPIに対応している。

第3図に検討を行った解析モデルを示す。通常的设计でよく用いられている固有値解析と線形応力解析である線形構造解析、及び大変形・弾塑性を考慮した非線形構造解析を例題として行った。また、同一形状でメッシュサ

イズを変えて、自由度で示すように4種類のモデルサイズによる影響について比較した。第1表に主要な並列化と計算機の使用コアパターンを示す。表のパターン名は、先に示した並列化の方式と使用した計算機の台数を示し、コア数は計算に用い



第1表 検証パターン

計算時間の検証を行った分散構成と使用する計算機数とコア数の例を示す。検証結果ではパターン名とコア数の組み合わせで構成を表す。

パターン名	コア数	計算機1使用コア	計算機2使用コア	計算機3使用コア
SMP	6	□□□□□□		
	8	□□□□□□□□		
MPI1	2	□□		
	8	□□□□□□□□		
MPI2	2	□	□	
	4	□□	□□	
	6	□□□	□□□	
MPI3	3	□	□	□
	6	□□	□□	□□
	9	□□□	□□□	□□□

第2表 計算機一覧

検証で使用した計算機の仕様を示す。

計算機名	仕様(CPU, メモリ, OS)
A	Xeon 2.67GHz (8コア×2CPU) 16GB, WindowsXP 64bit
B	PentiumD 3.2GHz (2コア) 4GB, WindowsXP
C	PentiumD 3.0GHz (2コア) 4GB, WindowsXP
D	Xeon 3.06GHz (2CPU) 2GB, WindowsXP
E	Athlon64 1GHz 1GB, WindowsXP

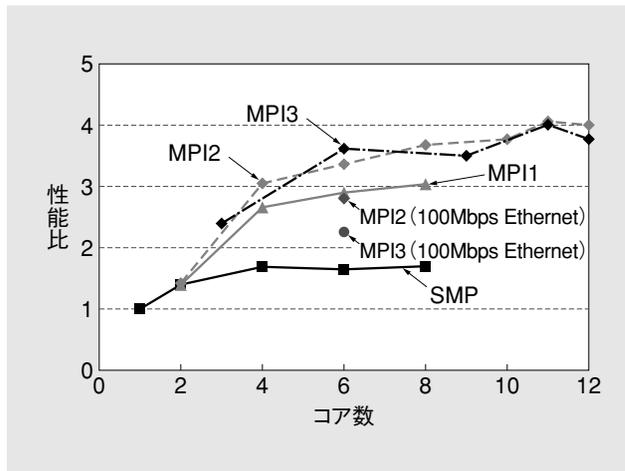
たすべてのコア数を合計した値である。

第2表に使用した計算機の代表的な仕様を示す。インターコネクトは、基本的に1Gbps Ethernetを使用している。

比較する指標には、実行時間(Elapsed Time)を計測し、1コアのみ使用した場合を基準とした計算速度の比である性能比を採用した。

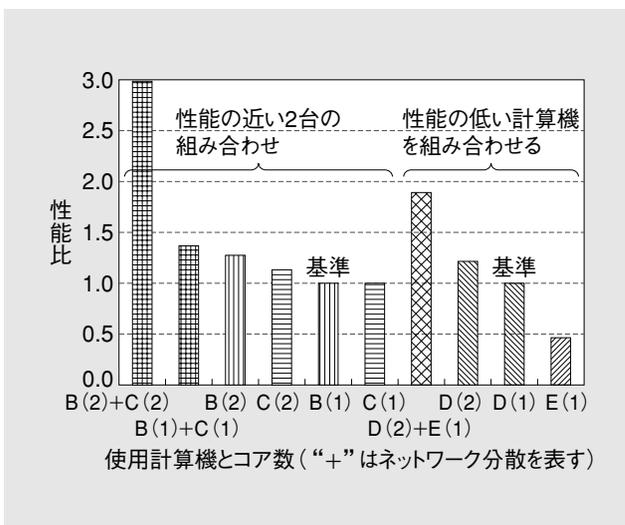
3.1 実行形態による比較

(1) コア数と分散方式 非線形構造解析について、計算機Aにおいて使用コア数と分散方式を変えて比較した結果を**第4図**に示す。ここで、MPI2の記号例は**第1表**に示すように、並列化方式がMPIで計算機が2台構成であることを示す。SMPは4コアで性能向上が飽和することがわかる。一方、MPI1~3はSMPに比べ性能比が高いが、コア数が6個以上になると向上の割合が低下することが分かる。これは分割したことによる分割境界の整合性に関する処理、通信量のオーバーヘッドの影響などと考えられる。



第4図 非線形構造解析の結果

計算機Aを使用した非線形構造解析において、パターンごとに使用コア数を変えて性能を比較した。



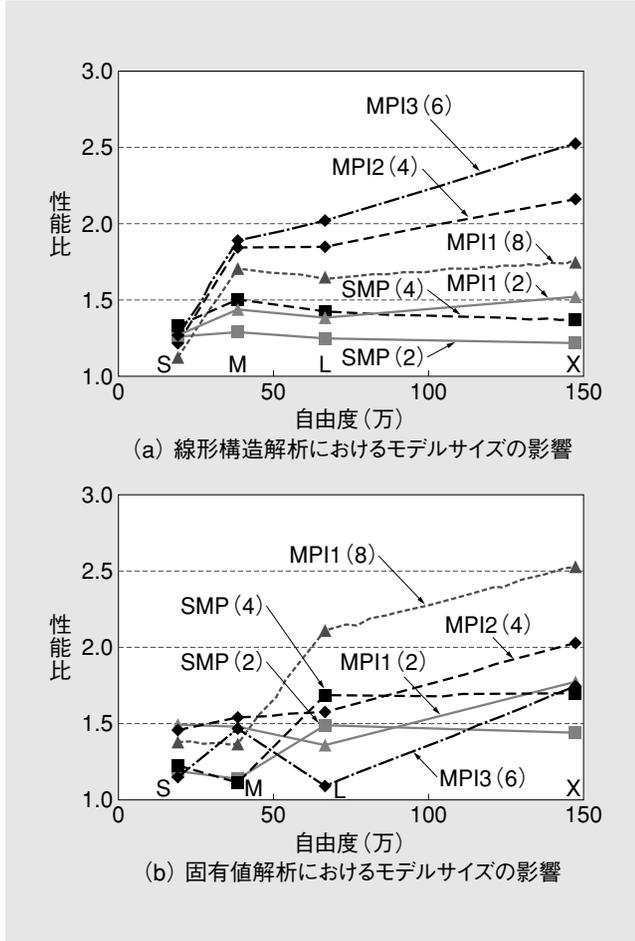
第5図 ネットワーク分散の効果

非線形構造解析において、2台の計算機を使用したときのMPIの効果を示す。性能の近い2台を使用した場合と、性能の低い計算機を組み合わせた場合について基準と比較した。横軸の項目名は計算機名と使用コア数で表す。

また、インターコネクトの速度を変え100Mbps Ethernetとした場合で、3台構成としたMPI3と2台構成のMPI2を比較すると、MPI3の方が効果が低い。この原因は、使用する計算機の増加により、データ通信量が増えるためと考えられる。

(2) ネットワーク分散の効果 次に、2台の計算機においてMPIの効果を検証した結果を**第5図**に示す。ここで、横軸の項目はX(Y)として表し、Xは計算機名(**第2表**)を、Yは使用コア数を示している。また、“+”記号はネットワーク分散で2台の計算機を使用することを意味している。

1台で2コア使用したローカルMPI(**第5図**のB(2)及びC(2))に比べて2台で1コアずつネット



第6図 モデルサイズの影響

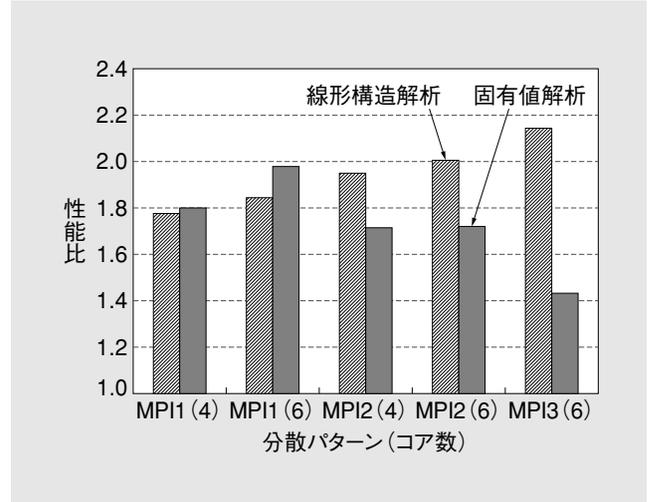
1台から3台の計算機を使用したときのモデルサイズによるMPIの効果への影響を示す。モデルサイズは自由度で表す。検証パターンはパターン名(第1表)と使用コア数で表し、代表的な例のみ記載した。

ワーク分散させた場合(第5図のB(1)+C(1))は若干性能が向上している。更に4コア使用する(第5図のB(2)+C(2))と2倍以上の性能向上が得られる。比較的遅いインターコネクトを使用した場合でもネットワーク分散による効果は大きい。また、遅い計算機と組み合わせた場合(第5図のD(2)+E(1))も効果が認められる。

3.2 解析の種類と分散構成による検証

計算機Aを3台使用し、使用コア数の構成を変化させ計測した。解析の種類は第3図の非線形構造解析・固有値解析・線形構造解析で、固有値解析と線形構造解析においてはモデルサイズも変化させた。

(1) 解析の種類とサイズの比較 固有値解析と線形構造解析に対して、検証パターン(第1表)ごとにモデルサイズと性能比の関係を比較した結果を第6図に示す。SMPはサイズによる性能の変化はほとんどないが、MPIは大きな問題において



第7図 MPIの効果

固有値解析及び線形構造解析において、使用コア数を4又は6に固定し計算機の台数を1台から3台まで変えた場合のMPIの性能を示す。モデルサイズの影響を除くためモデルサイズごとの結果を平均化した。右の項目ほどネットワーク分散の度合いは高くなる。

より性能比が大きくなることが分かった。

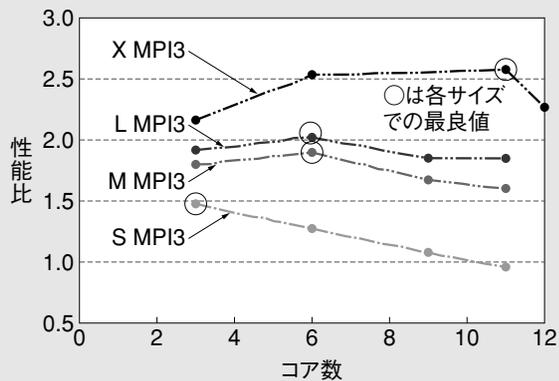
(2) ネットワーク分散MPIの効果 同じ使用コア数で計算機の台数を1台(ローカルMPI)から3台まで変えた場合のMPIの性能比を検討した結果を第7図に示す。ここでは、4種類のモデルサイズの性能比を平均化して示した。

線形構造解析では分散により性能比も向上するが、固有値解析では分散により性能比が低くなる逆の傾向となることが分かった。

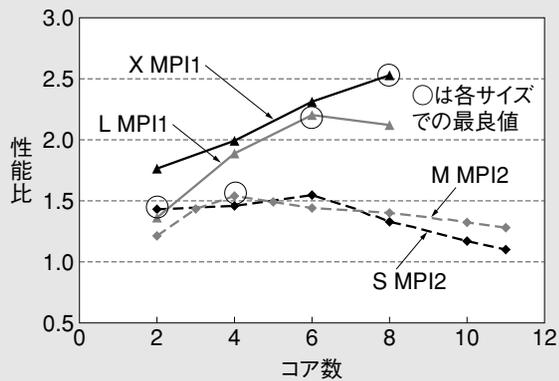
(3) 性能向上の最大値の比較 問題の種類別、サイズ別に性能向上の傾向をまとめて検討した結果を第8図に示す。図中のX MPI3などの記号例は第3図に示したサイズXのモデルを第1表のMPI3の検証パターンで計算したことを示している。モデルサイズ別に最大の性能向上が見られるコア数を○印で囲んでいる。これから、サイズが大きいほど、コア数を増やした効果が高くなる。しかし、コア数を必要以上に増やすと性能が低下する傾向が得られた。固有値解析ではサイズが大きい場合はMPIの効果が高いが、線形構造解析に比べて効果は低い。問題の種類によらずサイズが小さいとコア数増加による効果は低い。

第8図の結果からモデルサイズに対して今回の検証で最適であったコア数の関係を第9図に示す。

なお、これらの図では性能の良い代表的なパターンのみ記載したが、他のパターンも同様の傾向を示しているので割愛した。



(a) 線形構造解析における性能の傾向



(b) 固有値解析における性能の傾向

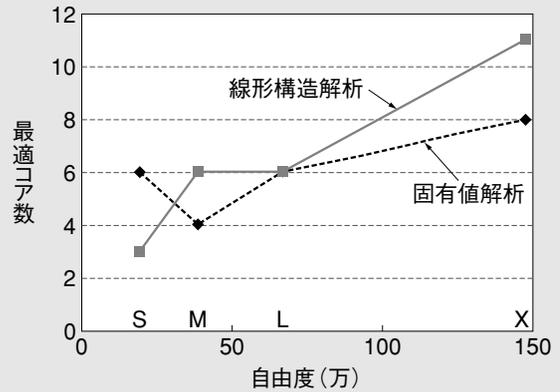
第8図 性能向上の傾向

計算機Aを使用した固有値解析及び線形構造解析において、モデルサイズ別に、パターンごとに使用コア数を変えて性能を比較した。検証パターンはサイズとパターン名（第1表）で表し、代表的な例のみ記載した。

4. ま と め

汎用有限要素法プログラムANSYSでHPC技術を用いた場合の解析について、問題の種類、問題の大きさ、更に使用する計算機の条件において効率的な計算方法の検討を行い、以下の結果が得られた。

- (1) MPIの方がSMPよりも効果は大きい。
- (2) 構造解析においては、ローカルMPIよりもネットワーク分散MPIの方が効果は大きい。
- (3) 固有値解析ではモデルサイズに依存するが、傾向としてはローカルMPIの方が効果は大きい。
- (4) インターコネクトは1Gbps Ethernetでは十分な性能が出ているが、100Mbps Ethernetでは性能が低下する。



第9図 モデルサイズと最適コア数

固有値解析及び線形構造解析において、モデルサイズによる最適な使用コア数を示す。

- (5) 本検討の範囲では、最も効果が大きかった非線形構造解析の場合、3台の計算機で最大約75%の計算時間短縮（本例題では平均約16時間から約4時間へ短縮）を実現できた。

5. む す び

ネットワーク分散により計算時間の大幅な短縮を行うことができるが、解析のモデルにより効果が相違することが分かった。今後、技術計算の効率化並びにこれを用いた設計の最適化支援技術の開発を進めていく所存である。

《参考文献》

- (1) ANSYS社製品リリース11.0アップデートセミナー資料Vol.2

《執筆者紹介》



江尻光良 Mitsuyoshi Ejiri
ソフトウェア技術及び解析技術の開発に従事



佐々木聖夏 Kiyoka Sasaki
解析技術の開発に従事