

EasyISTRの高速化(大規模メッシュ対応) (EasyISTR ver 3.04.190923)

大規模メッシュを扱う上でネックとなる箇所

1. メッシュ変換
2. メッシュファイルの読み書き

1. メッシュ変換 (unv → Fistr形式)

現状は、(unv → abaqus)変換後、(abaqus → Fistr)に変換している。

これを、直接(unv → Fistr)変換のみに修正。

unv → abaqus変換は、既存のCAELinuxに付属されていたものを流用していた。

unv → Fistr変換スクリプトを新たに作成した。

Linuxの場合 (220万要素:tetra 1次要素)

ver	変換時間	備考
3.02	1分50秒	
3.03	1分29秒	アルゴリズム修正 (文字列演算→list演算に変更)
3.04	33秒	↑ (unv → Fistrに直接変換)

2. メッシュファイルの読み込み

高速化する方法として、multiProcessing (並列処理) を使用。
メッシュファイルをテキスト形式で読み込み、以後の処理 (文字→数値変換) は、

!NODEの読み込み

!ELEMENTの読み込み

その他 (!NGROUP, !SGROUP等の読み込み)

に分けて各processに割り当てて処理 (最大3並列) する。

systemのcpu数を確認し、1コア～3コアで並列処理する。

これにより、約1.8倍程高速化が実現。

各processから転送されるデータ量が多いので、通信負荷が大きく、1.8倍程度の高速化。

220万要素の読み込み時間

15秒→8秒

メッシュ変換の場合は、独立処理できる箇所が少なく、並列化の効果が薄い。