超大規模計算時代における並列処理とポスト処理の問題点と解決手法

深沢圭一郎[1, 2, 3]、森江善之[1, 3]、南里豪志[1, 3]、村田健史[4]

1. 九州大学情報基盤研究開発センター
2. 九州大学宙空環境研究センター
3. CREST, JST
4. NICT

Feb. 15, 2012

KYUSHU UNIVERSITY

KYUSHU UNIVERSITY 2011
100th Anniversary

Contents & Motivation

◆エクサスケール計算時代における問題

□超大規模並列計算における問題

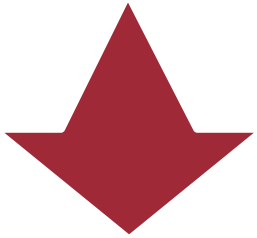
- ✓ 並列処理に伴うメモリ使用量
- ✓ 並列数増加による並列化効率の低下
- ✓ 不適切ランク配置による通信衝突

□ポスト処理の問題

- ✓ 大規模データのI/O処理
- ✓ 大規模データの解析
- ✓ 大規模データを保存するストレージ

超大規模並列計算における問題点

- エクサ時代における超大規模並列では現状のMPIではメモリの問題で実現が難しい
- また並列化効率の悪化により並列計算の意味をなさなくなることが懸念されている。

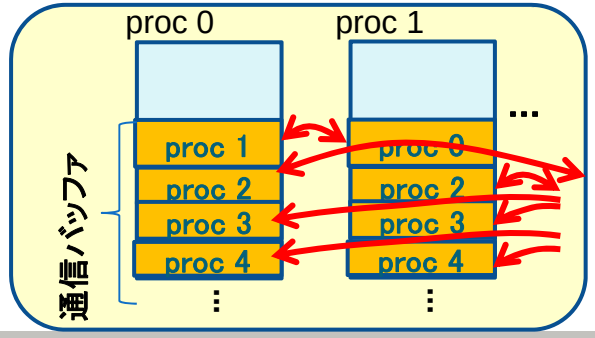


通信バッファ領域を総メモリ容量の10%以内に抑えながら、実アプリケーションで数千万～数億プロセスまでの性能向上を維持することを目標にJST CRESTで研究が始まっている

エクサスケールに向けた通信ライブラリの課題1

◆数億プロセスに耐えるメモリ管理機構

- エクサ時代でもプロセスあたりのメモリ容量は1～10GB程度と想定される
- 現状の通信ライブラリを利用した場合、一億プロセスにおける使用メモリ量
  - i) 各プロセスで全プロセスの情報を管理
    = 1プロセス 4Byte でも 400MB/プロセス
  - ii) 通信相手プロセス毎にバッファを用意
    = 1プロセス 1KBでも、最悪 100GB/プロセス



各プロセス用に
通信バッファを用意

省メモリライブラリを
開発中
MPI3には実装?

エクサスケールに向けた通信ライブラリの課題2

◆数億プロセスにおける並列化効率の改善

- Weak scalingにおける並列化効率@HA8000
  - i) 1024プロセスでmpi_senderecvを行う場合
    実アプリケーションで90%の効率
  - ii)8192プロセスでmpi_senderecvを行う場合
    実アプリケーションで85%の効率

この劣化は線形的に見えており、10万プロセスで並列化効率が50%を切ってしまう→並列化する意味が無くなる

通信の非同期化＋通信回数の削減
（深沢、他、HPCS2012）

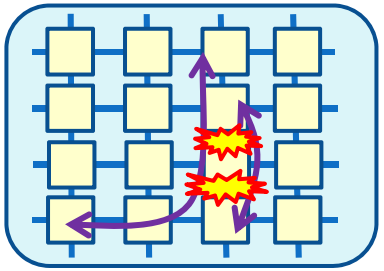
エクサスケールに向けた通信ライブラリの課題3

◆数億プロセスにおけるノード間通信衝突の問題

- 物理ノード配置とプロセス配置の関係
  エクサではクロスバーを使用することは困難
  →ファットツリー、メッシュ、トーラスなどを採用

実行時に決定する要素で性能が大きく変動
（プロセス配置による通信衝突や通信距離変化）

通信衝突



通信衝突はタスク配置に依存する
タスク配置最適化により通信衝突を回避可能

タスク配置最適化の実験

◆実験の条件

- 隣接通信の通信パターンにホップ数から最適化する手法（TAHB）と通信発生時間から最適化する手法（TACCC）を適用したときの通信性能の比較
- メッセージフローシミュレータ*を用いて実行時間を計測
- 7x6x6の格子の隣接通信(周期境界)
- 2-level, 16-aryのファットツリーへのマッピングを行う
- ファットツリーの多重度を1, 2, 4, 8 16と変更
- メッセージサイズ=100　BW=1.0

*出典：矢崎俊志, 石畑宏明, ``メッセージフローに基づくネットワークシミュレータ MFS の評価," 2011 年ハイパフォーマンスコンピューティングと計算科学シンポジウム(HPCS2011) 論文集, pp.1-9 (2011).

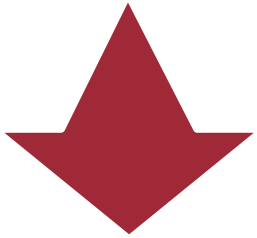
シミュレータによる予測通信時間結果

	Default	Random	TAHB	TACCC
多重度1	6400	9300	5040	4640
多重度2	3200	4800	2840	2420
多重度4	1700	3200	1740	1380
多重度8	900	1900	1160	800
多重度16	600	1200	780	500

- デフォルトタスク配置に対して最大で37%の性能向上
- TAHBに対しても最大56%の性能向上
- ランダムタスク配置に対しては最大2.4倍の性能向上
- 多重度が下がるほどTACCCによる性能向上率が増加する
  →多重度が低いほど通信衝突の可能性が増え、どのリンクで通信衝突が発生しているかを考慮できるため。

エクサスケールにおけるポスト処理の問題点

- エクサ時代における超大規模計算では扱うデータサイズもエクサスケールであり、今までのポスト処理技術では対応が難しい。
- そもそもエクサスケールの計算は必要なのか。



惑星磁気圏シミュレーションではまだまだ計算パワーが必要。
エクサでやっとMHDからVlasovに移行できるかのかも。

電磁流体コード -1

◆宇宙プラズマを取り扱う方程式(1)

□Vlasov方程式

- 無衝突Boltzmann方程式とMaxwell方程式から成るプラズマの振る舞いを最も正確に表現できる方程式系

  速度分布関数$f(\mathbf{x}, \mathbf{v}, t)$を考えると、

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} + \frac{q}{m}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f}{\partial \mathbf{v}} = 0$$

  これとMaxwell方程式を連立して解く。

- しかし、位置(x, y, z)3次元、速度(v_x, v_y, v_z)3次元と時間から成る非線形方程式系で、解くことが困難
  $f(x, y, z, v_x, v_y, v_z, t)$を計算するにはメモリ不足
  （例えば、1000^6=8PBのメモリ）

電磁流体コード -2

◆宇宙プラズマを取り扱う方程式(2)

□MHD (Magnetohydrodynamics)方程式

– Vlasov方程式のn次モーメント取ることで、求められる。
0次(速度空間で積分)、1次(**v**かけて積分)、2次(**v**^{2}かけて積分)より、

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\mathbf{v}\rho)$$

$$\frac{\partial \mathbf{v}}{\partial t} = -(\mathbf{v}\cdot\nabla)\mathbf{v} - \frac{1}{\rho}\nabla p + \frac{1}{\rho}\mathbf{J}\times\mathbf{B}$$

$$\frac{\partial p}{\partial t} = -(\mathbf{v}\cdot\nabla)p - \gamma p \nabla\cdot\mathbf{v}$$

を得る。これらと磁場の誘導方程式

$\frac{\partial \mathbf{B}}{\partial t} = \nabla\times(\mathbf{v}\times\mathbf{B})$　をまとめてMHD方程式という。

この場合、$1000^3 \times 8$=64GBのメモリ

土星磁気圏MHDシミュレーション

◆ 沿磁力線電流と渦構造の関係

□沿磁力線電流の強い箇所から磁力線を伸ばすと. . .

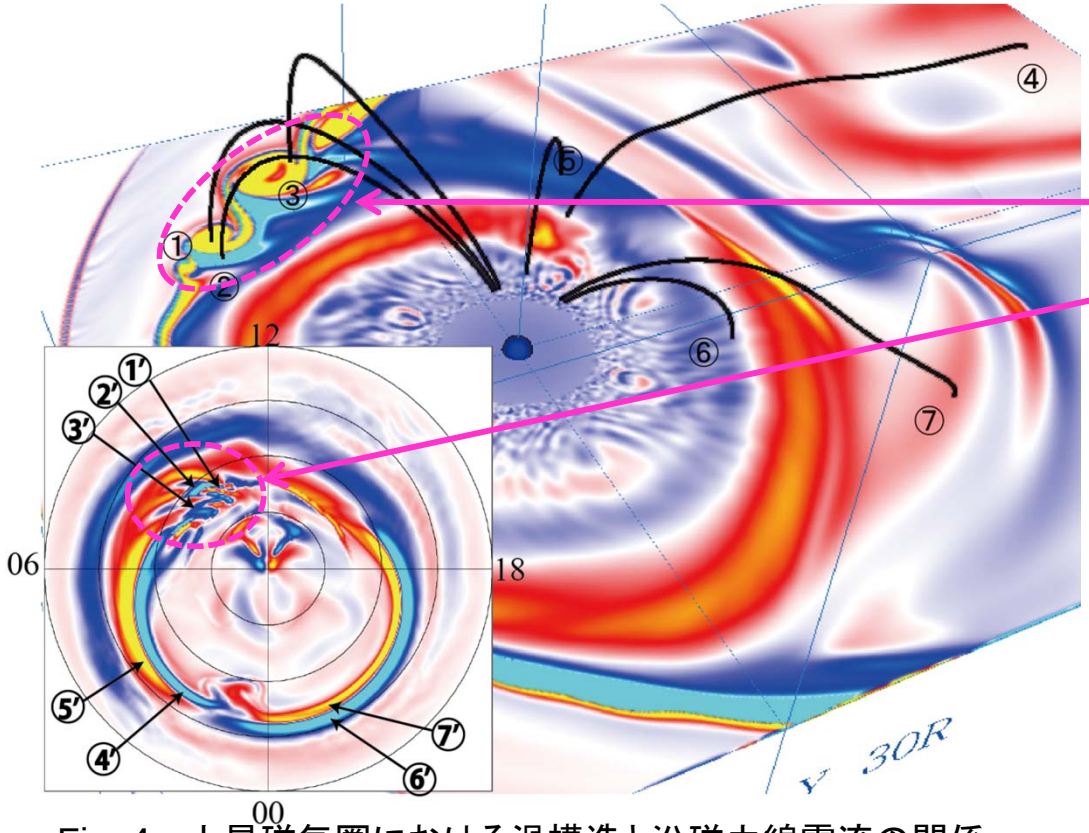


Fig. 4. 土星磁気圏における渦構造と沿磁力線電流の関係
[*Fukazawa et al., JGR*, 2011, accepted]

大規模計算による高精度なシミュレーションにより、土星渦構造と斑点オーロラ構造の関連性を初めて示唆。

観測結果に似た構造を再現。

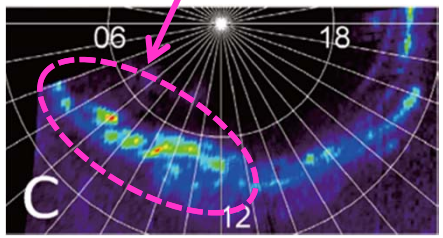


Fig. 5. 土星南極におけるオーロラの輝度[*Grodent et al., 2011*]

さらに高精度の計算へ

◆オーロラのような現象を精度良く解く必要がある

- 今までの計算では0.1 R_S（R_Sは土星半径 = 60,300 km）の格子幅を利用。
- マクロとミクロの遷移領域（MHD近似の限界領域）を計算するためには、0.01 R_Sの格子幅→$15{,}000^3$程度のグリッドが必要。

  ➡ 1タイムステップで200TBのデータ容量、時間方向を考えると200×100=20PBは最低限必要

- Vlasovで解く場合、位置、速度空間で1000^6のグリッドがあれば現状のMHDシミュレーションスケールは計算可能

  ➡ 1タイムステップで8PBのデータ容量

大規模計算におけるI/O処理

◆I/Oデータの巨大化

- 現在で1データで100GB程度、数TBも現実サイズ
  →SATA3で6Gbpsだとして、100GBで130秒、
  1TBで1300秒かかる（理論値）。
  →1ノードのメモリ量を超えるため、書き出しが
  出来ない。
  （ほとんどの分散メモリマシンは64GB/node）

プロセスorノード毎の分散読み書きだしが必須
→1000ノードで1TBを書き出すと、1ノード当たり1GB
を担当（6Gbpsでも1.3秒程度で書き出し可能）

大規模計算のポスト処理1

◆巨大な分散データをどう解析するか

- 現状100GBのデータは100MBずつ1000個に分散書き出しされている
- 大部分の可視化アプリは分散読込非対応
  →100GBに結合させるためにはメモリが100GBを超える計算機が必要
  →あるにはあるが、一般的に基盤センターには無い

分散データ対応並列可視化アプリケーションが必要
現状ではAVS/Express PCEが一つの解

AVS/Express PCEの評価

◆42GBの分散データを64並列で可視化

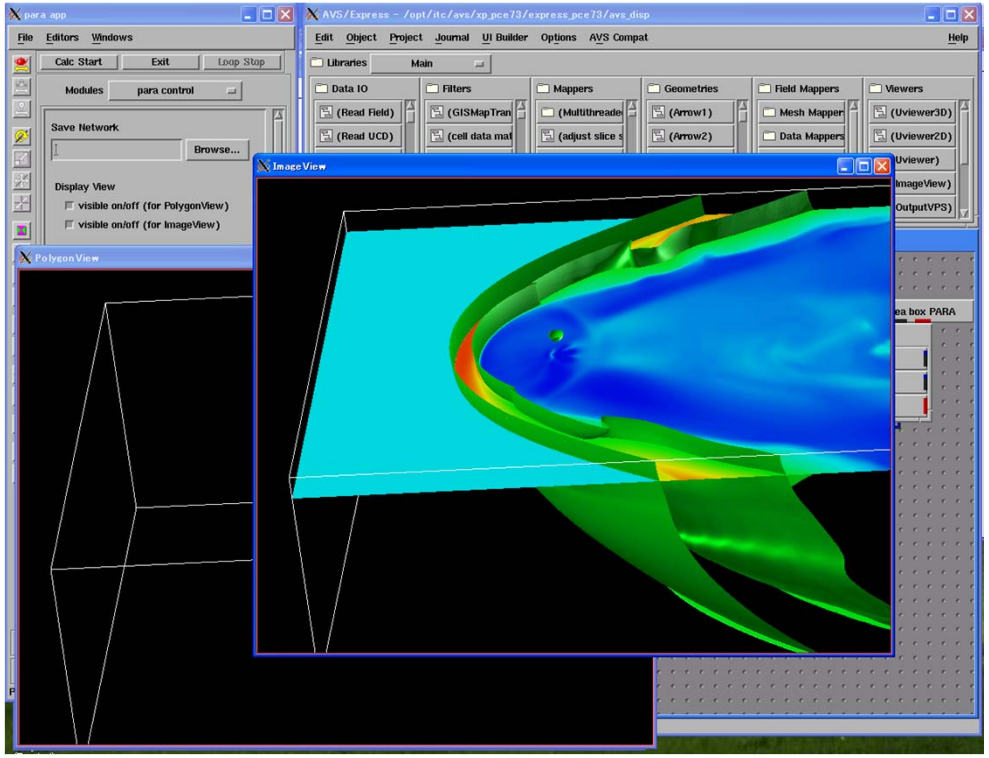


Fig. AVS/Express PCEでの土星磁気圏可視例
[深沢、他、東京大学スーパーコンピューティングニュース、Vol. 13、No. 5、39-45,2011]

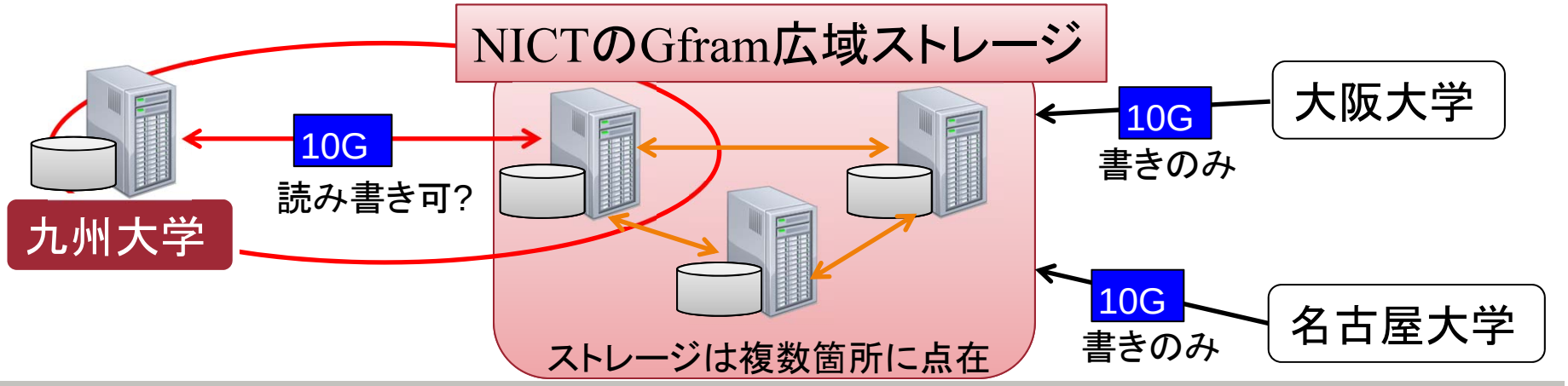
- ✓ ファイルを読み、可視化結果が表示されるまでに約2分かかる。
- ✓ 画面の視点を変えてみると、非常に動きが遅く、画面がフリーズすることもある。
- ✓ 各ウインドウのサイズを変更すると、動作が重くなりスムーズにサイズ変更が行えないことがある。
- ✓ スライス面の変更やisosurfaceの閾値変更などは30秒程度で完了。
- ✓ 視点変更はGPUの問題と考えられ、それ以外は42GBを読込GUIで可視化していることを考えれば及第点。

大規模計算のポスト処理2

◆巨大な計算結果をどこにおくか

- 現状で1シミュレーションで100GB×300=30TB
  - →基盤センターに置いておけない
  - →置き場所があっても転送が面倒

基盤センターに直結したグリッド型ストレージをNICTが整備



新しい計算結果の転送手法

◆ GPGPUとUDT通信を使った計算システム(@SC10)

1. GPUを使いシミュレーションを行う@NICT小金井
2. 計算結果をメモリ上から直接SC10の会場のサーバ上のメモリにコピー
3. サーバでメモリ上のデータを 使って(3次元)リアルタイム可視化@SC10会場

SC07、08、09においてUDT転送実験を行い、計算結果のディスク書き出しの遅延が大きく、計算が通信に追いつかないことを確認済み。今回はGPUを使い高速に計算を行い、ディスクI/Oを用いず、結果の転送を行った。
→MAX9Gbps、平均6Gbpsで転送し、可視化までシームレスに実現

Summary

◆エクサスケール計算時代における問題

□超大規模並列計算における問題

- ✓ 並列処理に伴うメモリ使用量→開発中
- ✓ 並列数増加による並列化効率の低下
  →非同期＋通信自体の削減
- ✓ 不適切ランク配置による通信衝突→配置の最適化

□ポスト処理の問題

- ✓ 大規模データのI/O処理→分散処理
- ✓ 大規模データの解析→分散データ対応並列可視化
- ✓ 大規模データを保存するストレージ
  →分散ストレージ＋高速データ転送