

**様式 1 : HPC 特別プロジェクト報告書**

プロジェクト名 : T2K オープンスパコンへのインヤン地球ダイナモコードの移植

代表者 (所属) : 陰山 聡 (海洋研究開発機構)

地球シミュレータ用に開発・最適化した地球流体系のコードを HA8000 システムに移植し、その速度計測と最適化の効果を調べた。このコードは球殻領域内の磁気流体力学 (MHD) 方程式を有限差分法で解く地球ダイナモシミュレーションコードである。計算格子はインヤン格子、総格子点数は 8 億 1 千万である。このコードは地球シミュレータ (ES) 用に開発・最適化したコードで、言語は Fortran90、並列化手法は 2 次元領域分割による Flat MPI、上記の格子点数で ES の 4096 コア上で 15.2TFLOPS の性能がある。

**移植について** : 最適化のためいくつか「手術」を施したコードも作ったが、結局、ES 用のソースコードをほぼそのまま使ったものが最も速かった。変更したのは以下の 2 点だけである : (i) use mpi を include 'mpif.h' に変更。 (ii) MPI\_ALL\_REDUCE で MPI\_IN\_PLACE が使えなかったので、ダミーのバッファを使うよう変更。

**性能について** : 最大ノードでの記録のみ記すと、

- 512 ノード x 16 プロセス : 1.8 TFLOPS (ピーク性能の 2.4%)
- 512 ノード x 8 プロセス : 1.5 TFLOPS (ピーク性能の 4.0%)
- 512 ノード x 4 プロセス : 0.89 TFLOPS (ピーク性能の 4.7%)

**最適化について気がついたこと** :

- 3 次元配列のサイズ、特に最初の次元の大きさに速度が敏感に影響する。16 の倍数がよいようだ。半径方向の格子サイズを 512 から 513 にしただけで速度が 10% も落ちた。
- メモリロードを減らすために補助変数を使わないようにするのは効果がある。ただし、これは格子サイズが小さい時のみ。
- MHD 方程式の基本場をまとめて 4 次元配列を作ったが、高速化の効果はなかった。
- 空間の一つの次元を分割して、各小領域を順次解く方法は効果があった。
- ループのアンローリングは効果的。しかしそれはコンパイラにまかせれば十分。
- 桁落ちで極端に速度が遅くなるようだ。

**可視化について** : 我々が開発している HPC 用並列可視化ソフトウェア Armada は、グラフィックス関係のハードウェアを必要としない。スカラーデータに対しては等値面、ボリュームレンダリング、スライスを、そしてベクトルデータに対しては流線、矢印表示等の可視化が可能である。下の図はそのサンプルである。今回 8192 コアでの実行を確認した。

