第一回オープンソースCAEワークショップ

東京大学版 T2Kオープン スーパーコンピュータHA8000へのOpen∇FOAMのインストールと並列計算

今野 雅　(東京大学)

masashi.imano@gmail.com

はじめに

- 東京大学版T2Kオープンスーパーコンピュータ
- オープンソースCFDツールボックスOpenFOAM

をイントール

- 39万、290万、2,300万格子といった3レベルで

の非圧縮性・非等温LES解析

- 8ノード、128プロセッサ迄の並列化効率を調査

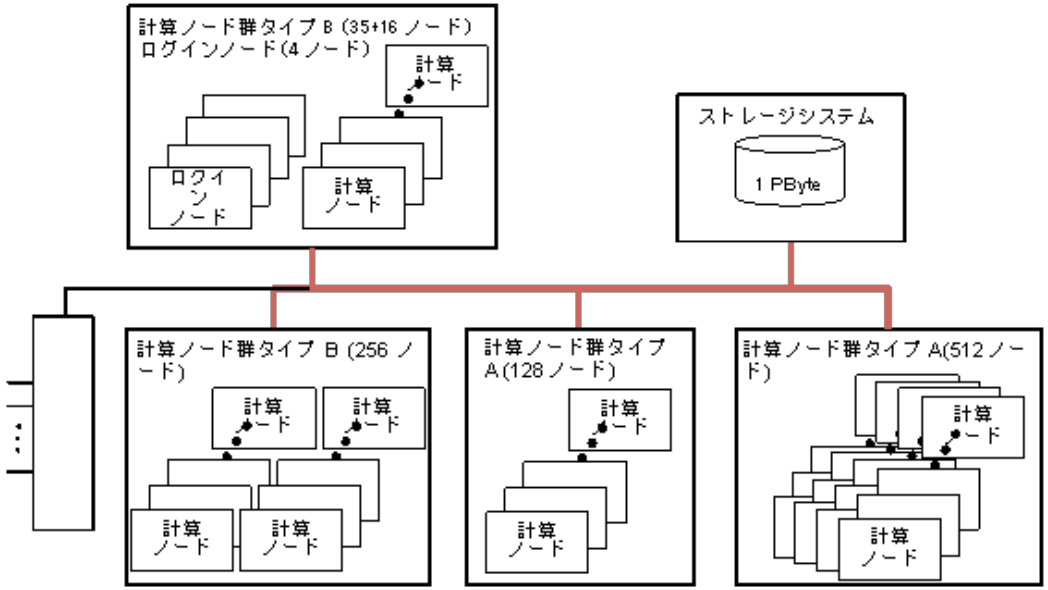
東京大学版T2Kオープンスパコン HA8000

(引用元:東京大学情報基盤センターHA8000クラスタシステム利用の手引)

東京大学版T2Kオープンスパコン HA8000

全体構成

総理論演算性能	147.1344TFLOPS
総主記憶容量	31.25TB
総ノード数	952
ストレージ装置容量	1PB(RAID6)



東京大学版T2Kオープンスパコン HA8000

ノード概観

普通のラックマウント型サーバPC

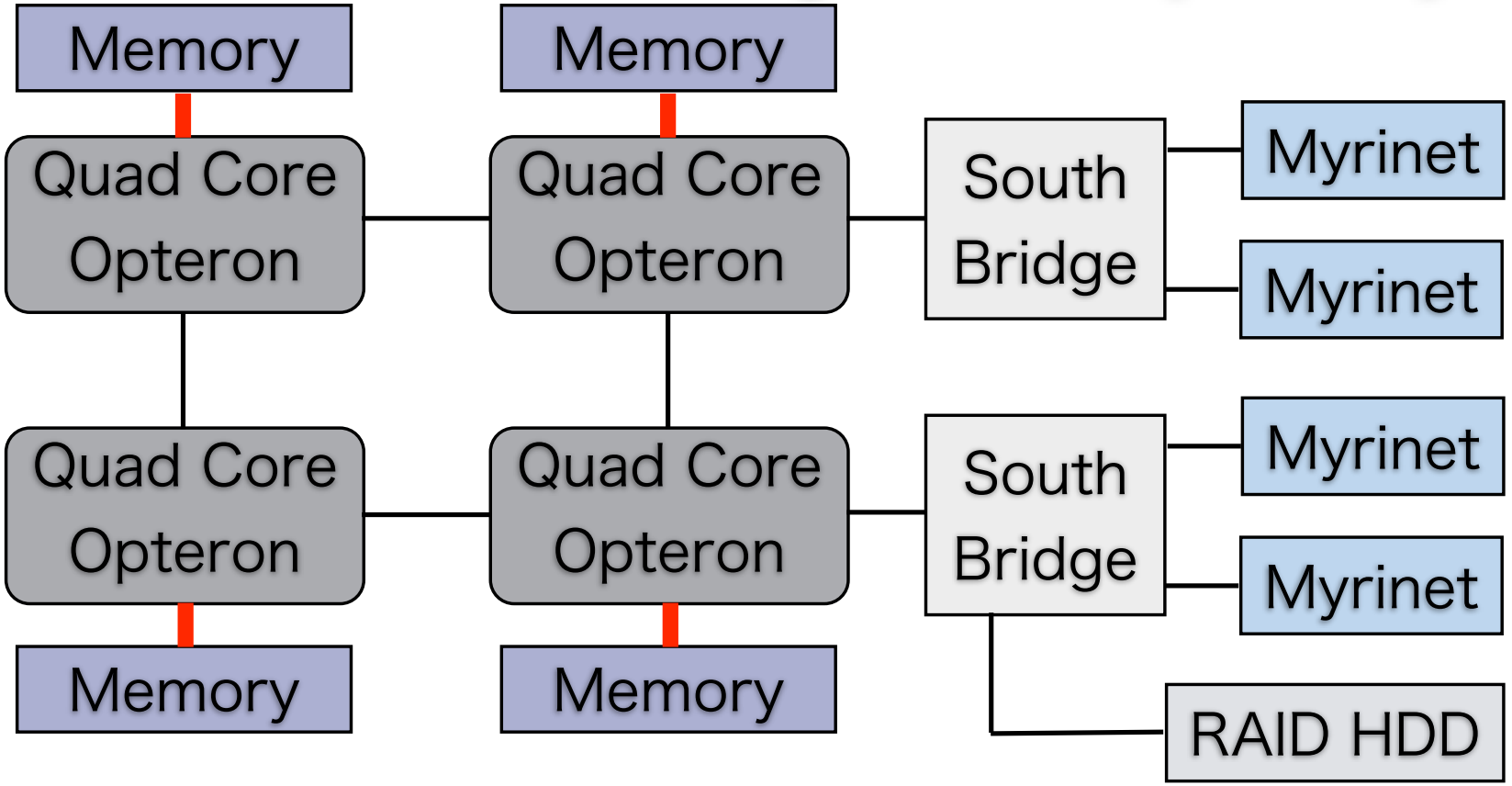
東京大学版T2Kオープンスパコン HA8000

ノードのスペック

ノード	理論演算性能	147.2GFLOPS
	プロセッサ数	4 (Quad Core)
	コア数	16
	主記憶容量	32GB(936ノード) 128GB(16ノード)
プロセッサ	プロセッサ数 (周波数)	AMD Opteron (2.3GHz)
	キャッシュメモリ	L2: 512kB/コア L3: 2MB/コア
	理論演算性能	9.2GFLOPS/コア

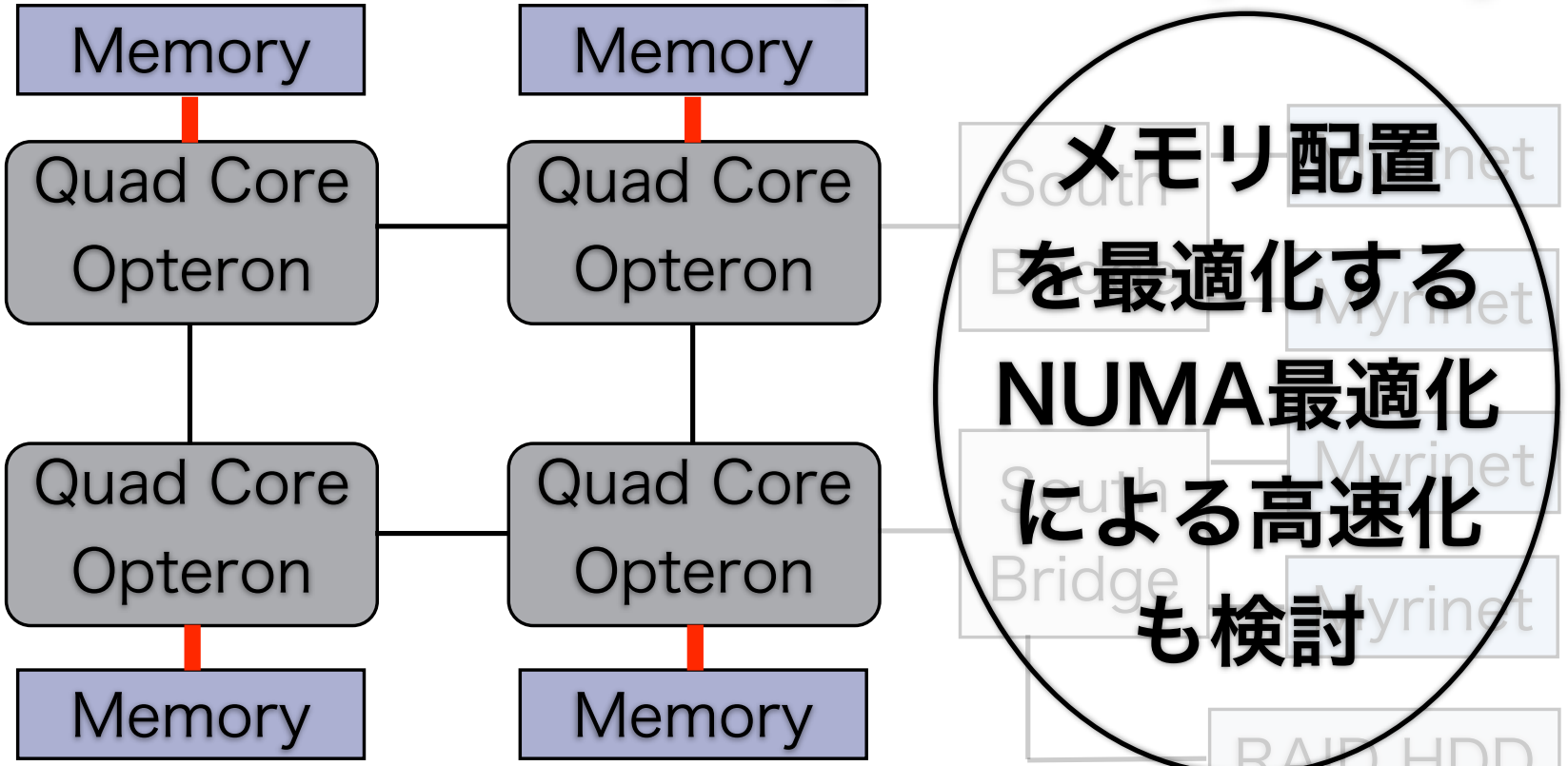
東京大学版T2Kオープンスパコン HA8000

アーキテクチャ
Non-Uniform Memory Access (NUMA)



東京大学版T2Kオープンスパコン HA8000

アーキテクチャ
Non-Uniform Memory Access (NUMA)



東京大学版T2Kオープンスパコン HA8000

ネットワークのスペック

構成	フルバイセクションバンド ネットワーク
ネットワークカード	Myrinet-10G
カード当りのバンド幅	1方向当り1.25GB/s (双方向に通信可能)
ノード当りのバンド幅 (カード数)	タイプA : 5GB/s (4本) タイプB : 2.5GB/s (2本)

東京大学版T2Kオープンスパコン HA8000

本研究で使用したネットワーク

- タイプA(5GB/s)は主にグループ利用のクラスタ
- タイプB(2.5GB/s)は主に個人利用のクラスタ
- 本研究ではタイプB (バンド幅がタイプAの半分)
- タイプAでは並列化効率等の性能が向上すること

が期待されることに予め注意されたい

東京大学版T2Kオープンスパコン HA8000

OS

- 従来のスパコン(SR11000、OS: AIX)では OpenFOAMのコンパイルすら難しかった
- **HA8000はスパコンなのにOSは普通のLinux!**

(具体的にはRedHat Enterprise Linux 5)

- 並列計算無しならバイナリ版がそのまま動く
- 並列計算にはMyrinet用のMPIライブラリをリンクするようにして、ソースからコンパイルする必要あり

並列計算環境

- OpenFOAM Version 1.5.x (2008/9/2時点)
- コンパイラ: Intel製C、C++ (Ver. 10.1.017)
- 最適化: -O3 -xO -no-prec-div
- 検討したオプション中では最速(後述)
- MPIライブラリ: Myrinet用のMPICH

並列計算環境

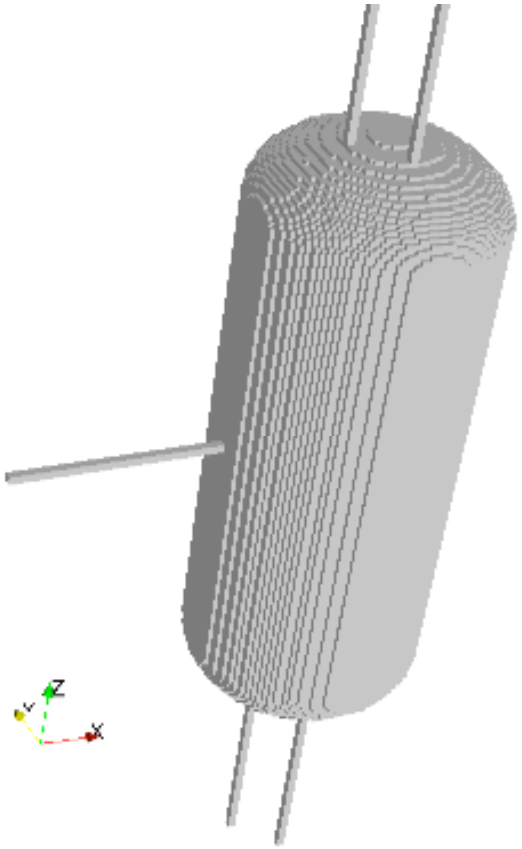
- 今回OpenFOAMのコードのカスタマイズは無し
- OpenMPのディレクティブも入れてない
- 並列化手法はピュアMPI
- プロセッサ間の通信量を減らすfloatTransfer

は、圧力方程式の反復回数が増えて、逆に計算時間が長くなるので、使用しなかった

計算条件

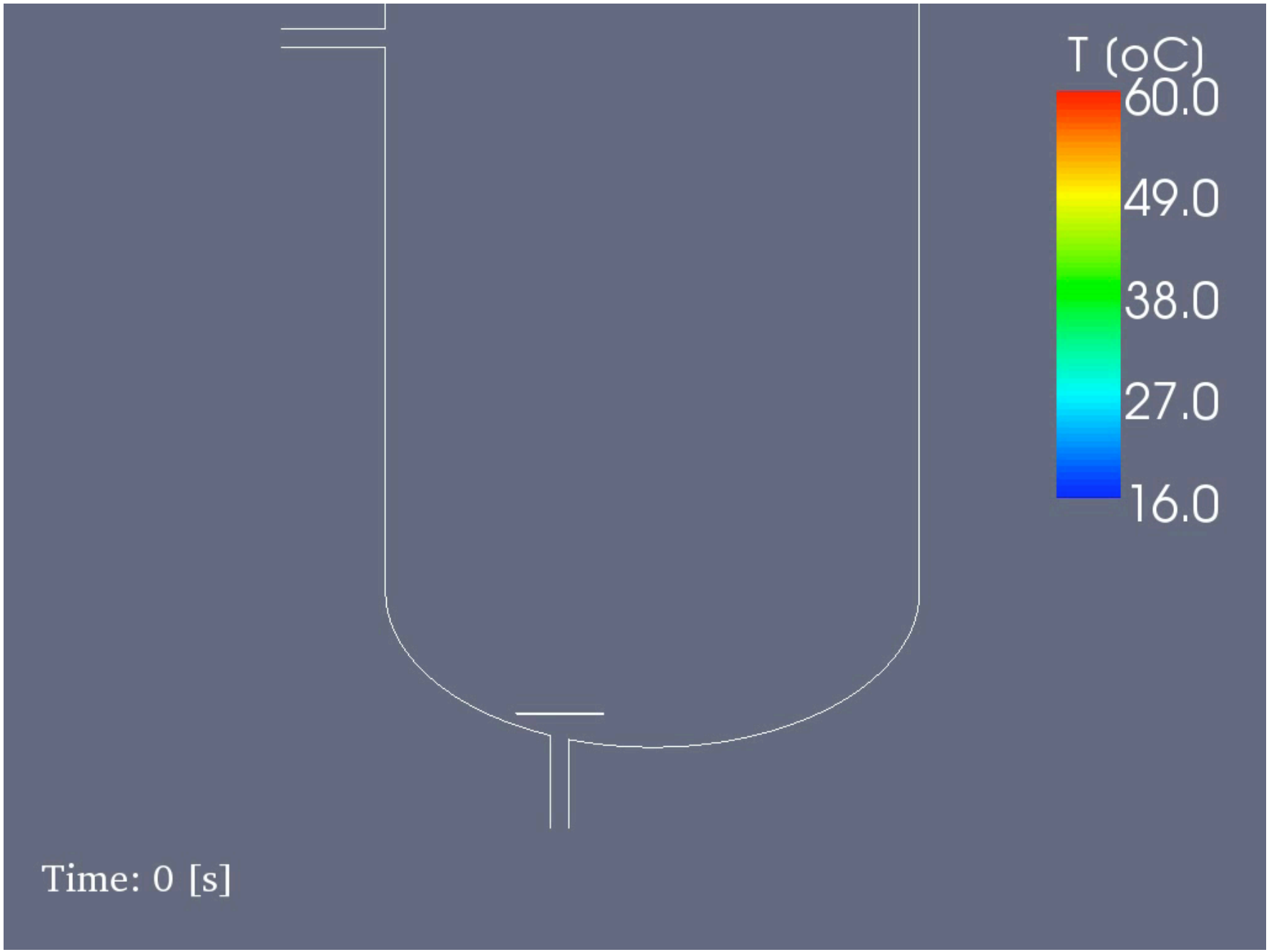
給水口	U,T：実測値、乱れ無し
中間取出口	U：実測値、T：勾配無し
貯湯槽表面	U：一般化対数則 T：外気との熱貫流率(1[W/m^{2}K])
乱流モデル	LES (Smagorinskyモデル)
スキーム	移流：二次中心、時間：一次
ソルバ	圧力:AMG、その他BiCGStab

計算対象ケース



境界：階段状近似

ケース	格子数	格子幅 [mm]	時間刻み [s]
Case-S	39万	10	0.02
Case-M	290万	5	0.01
Case-L	2,300万	2.5	0.005


T (oC)
60.0
49.0
38.0
27.0
16.0
Time: 0 [s]

並列計算ケース

プロセッサ数	ノード数	1ノード当り	NUMA最適化
1	1	1	Off
2	1	2	On
4	1	4	On
8	1	8	On
16	1	16	On
32	2	16	On
64	4	16	On
128	8	16	On,Off

コンパイル・オプションの検討

Case-L(格子数2,300万)、128プロセッサ

オプション	計算時間
-O3 -xO -no-prec-div	1054
-O3 -xW -no-prec-div	1056
-O3 -xO -no-prec-div -ip	1065
-O3 -xO	1112

-xO : Opteronでも動作する SSE3最適化

-xW: SSE2最適化

-no-prec-div: 除算の精度を上げる最適化無効

-ip: インライン化等の最適化 (-ipoは動作せず)

並列化効率の算出法

スピードアップ比

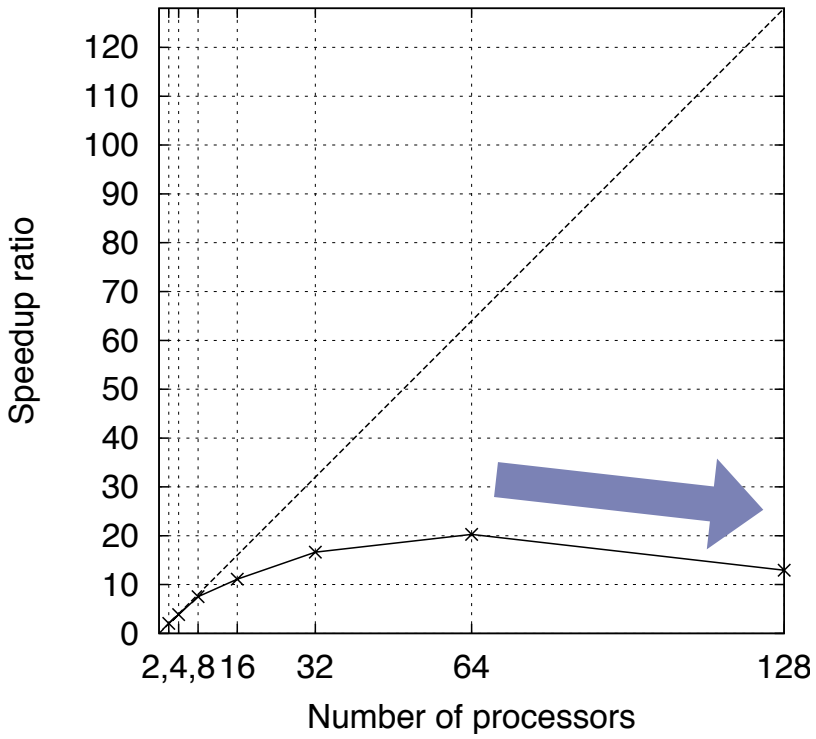
$$= \frac{1\text{プロセッサ時の計算時間}}{N\text{プロセッサ時の計算時間}}$$

並列化効率

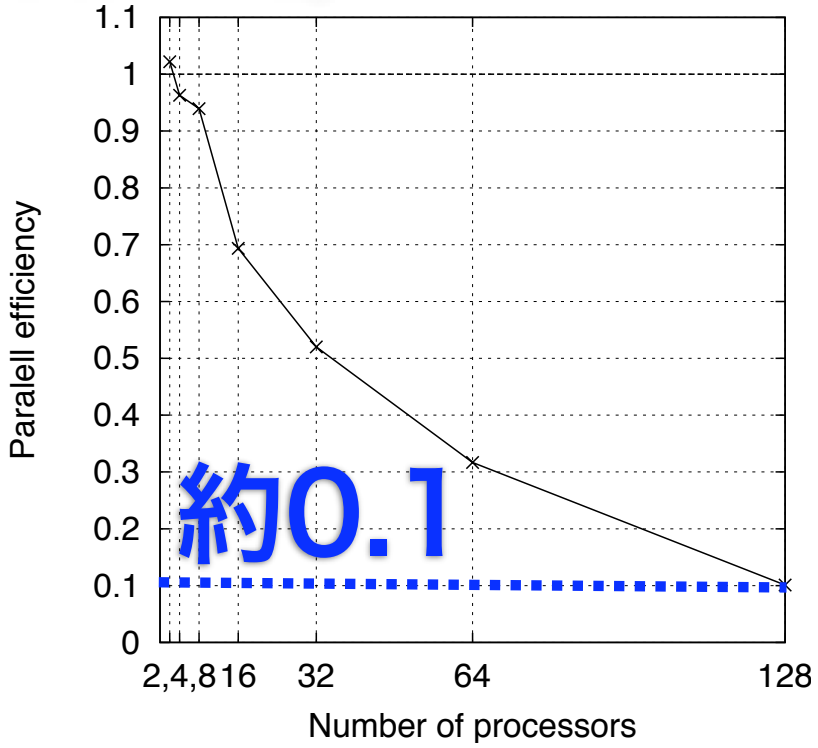
$$= \frac{\text{スピードアップ比}}{\text{プロセッサ数}N}$$

並列化効率

Case-S (格子数39万)



スピードアップ比



並列化効率

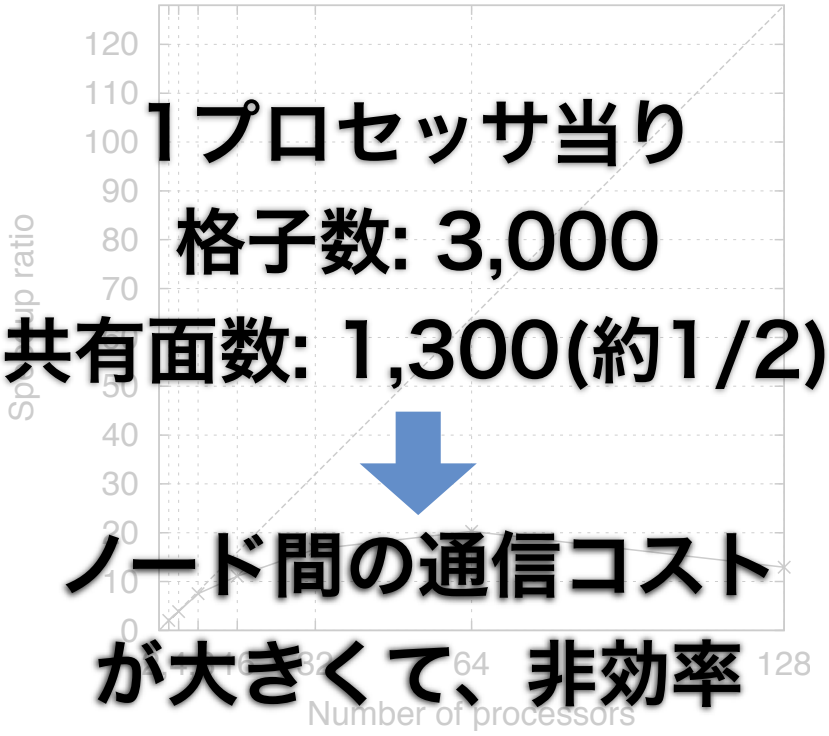
並列化効率

Case-S (格子数39万)

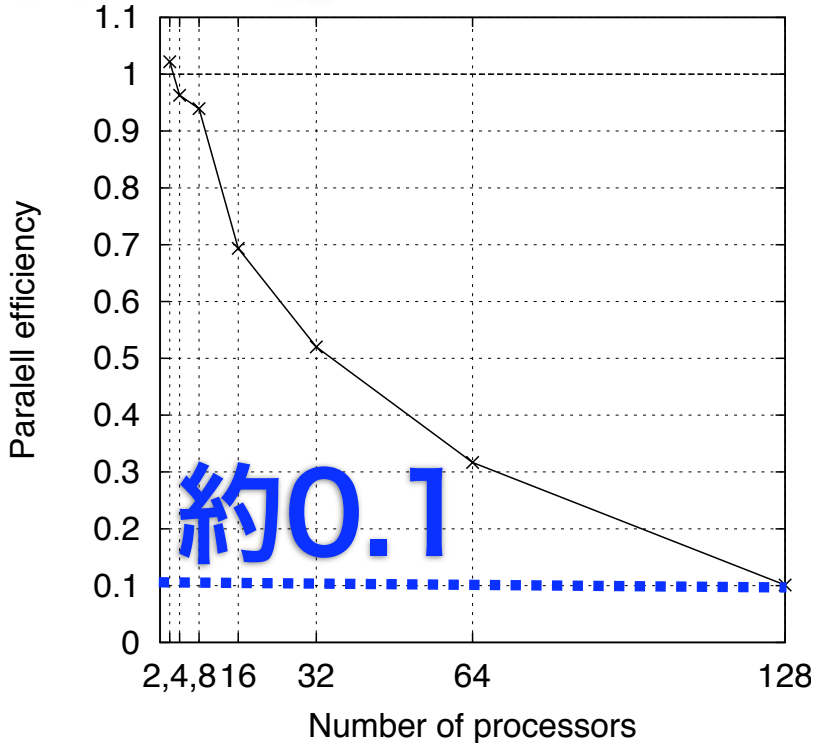
1プロセッサ当り
格子数: 3,000
共有面数: 1,300(約1/2)

↓

ノード間の通信コスト
が大きくて、非効率



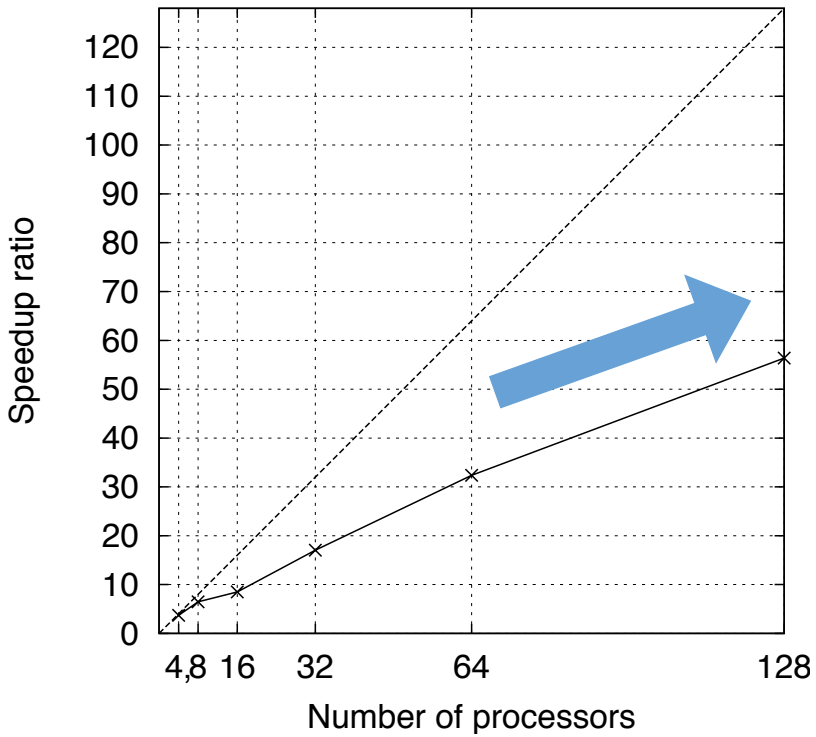
スピードアップ比



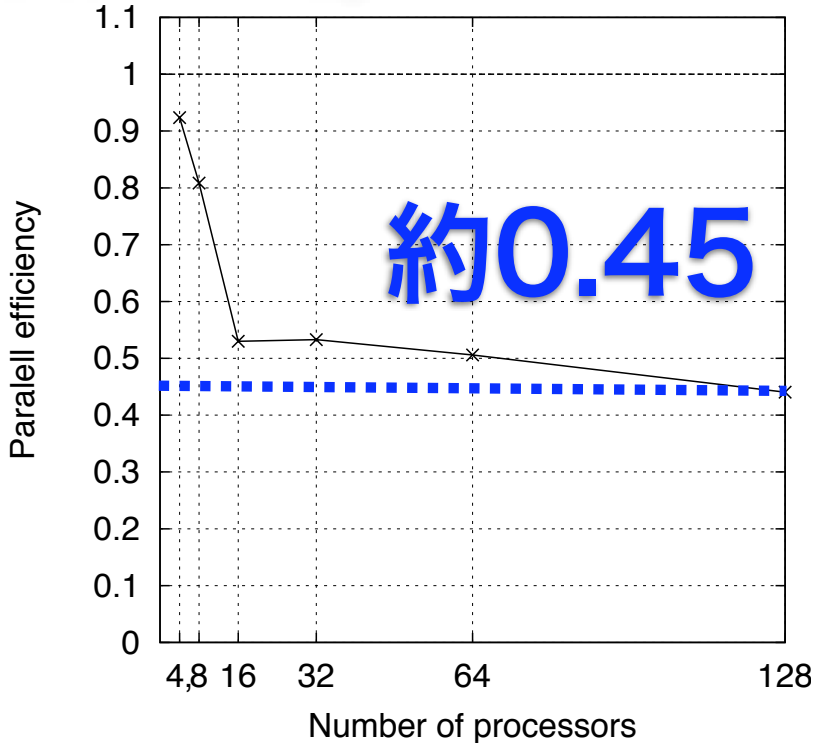
並列化効率

並列化効率

Case-M (格子数290万)



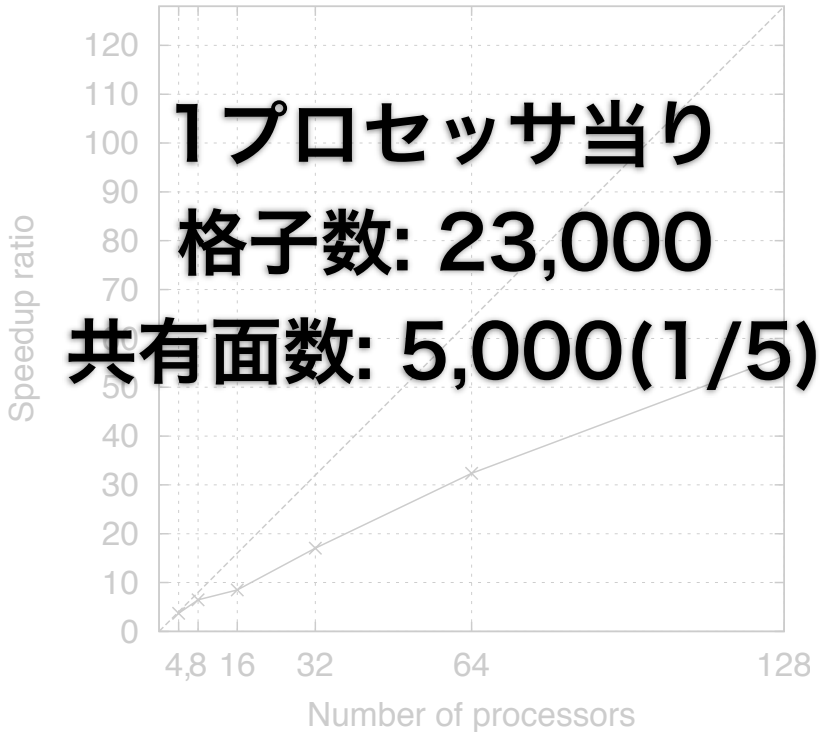
スピードアップ比



並列化効率

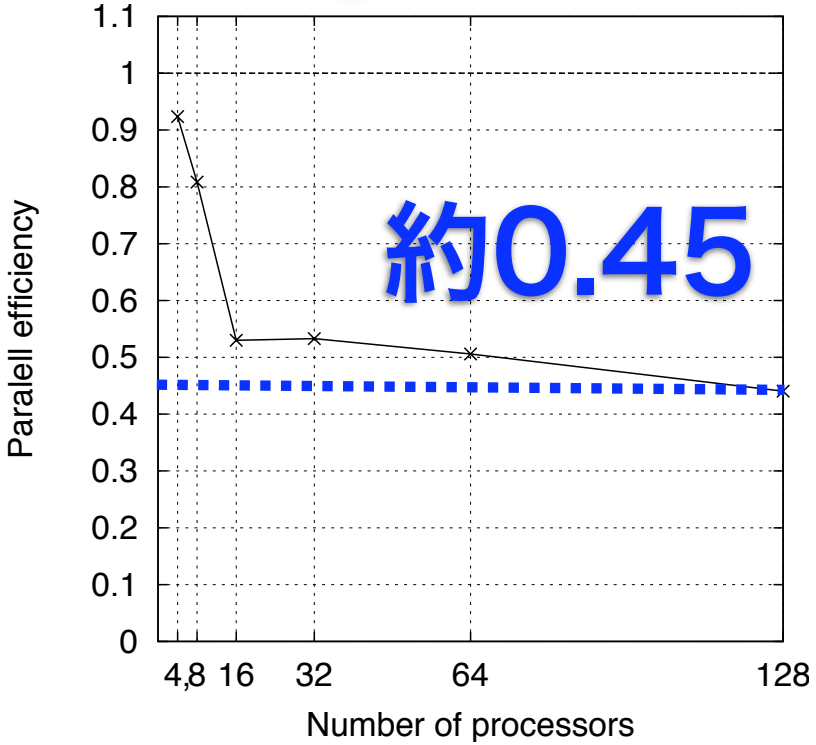
並列化効率

Case-M (格子数290万)



1プロセッサ当り
格子数: 23,000
共有面数: 5,000(1/5)

スピードアップ比

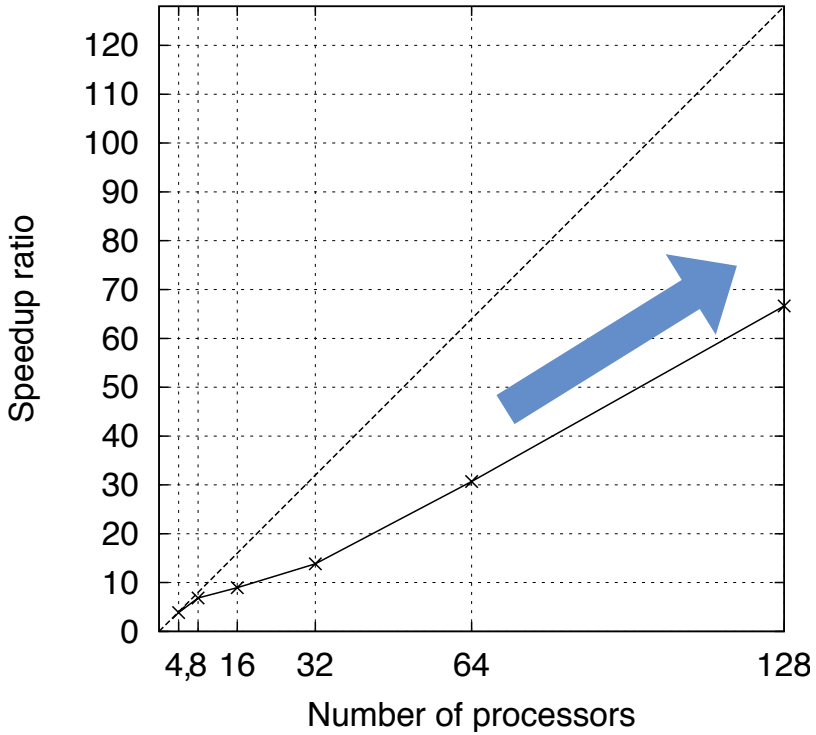


約0.45

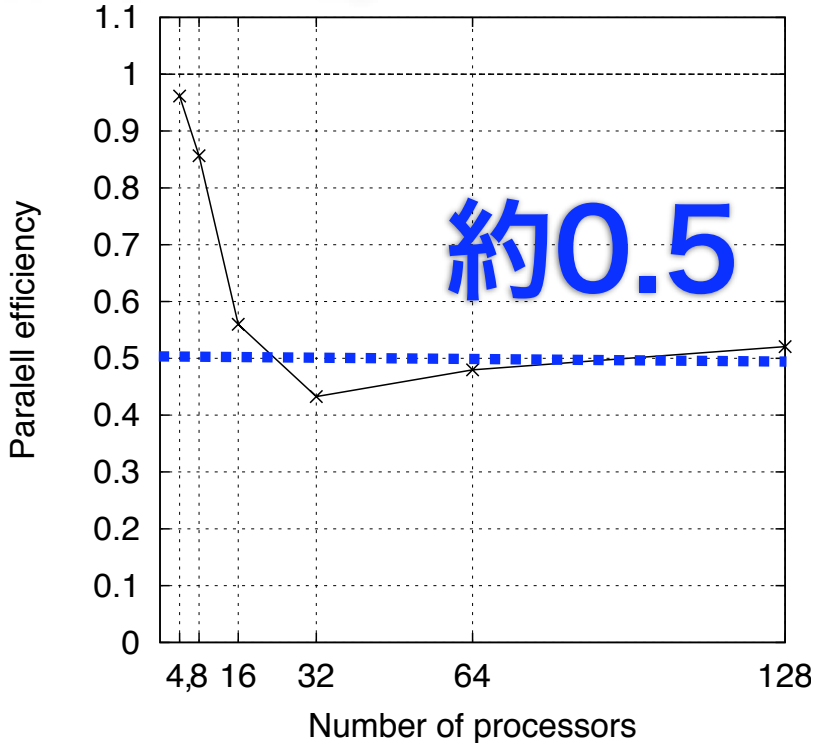
並列化効率

並列化効率

Case-L(格子数2,300万)



スピードアップ比



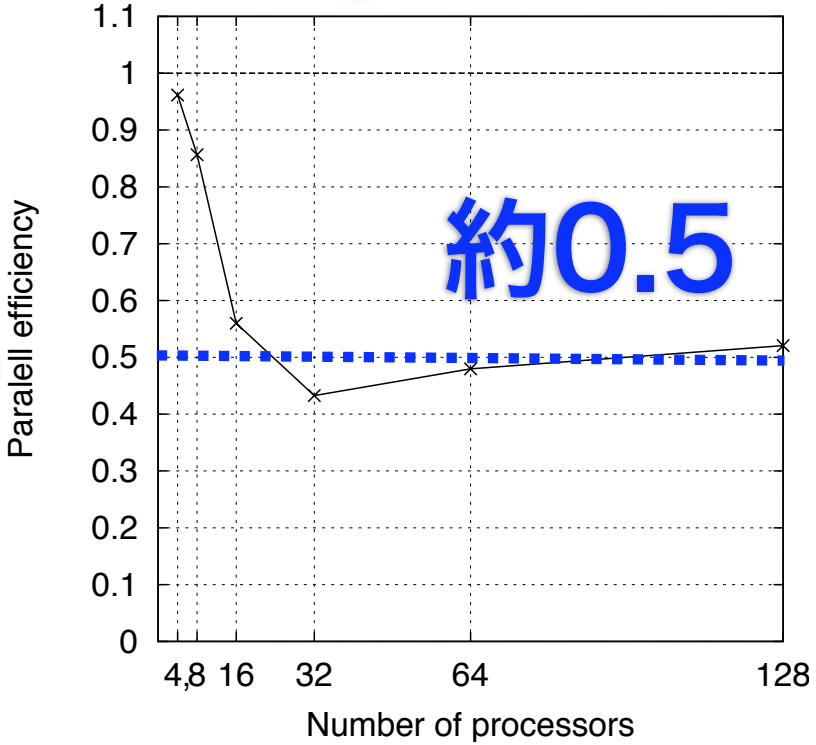
並列化効率

並列化効率

Case-L(格子数2,300万)

1プロセッサ当り
格子数: 180,000
共有面数: 20,000(1/9)

スピードアップ比



約0.5

並列化効率

NUMA最適化の結果

128プロセッサ

ケース	NUMA最適化によるスピードアップ
Case-S	1.6
Case-M	1.7
Case-L	1.1

コア当りのメモリ負荷が増えるとNUMA最適化の効果も大きいはずだが???

まとめ

- 東京大学版T2Kオープンスーパーコンピュータに、 オープンソースCFDツールボックOpenFOAMをイントールした。
- 39万、290万、2,300万格子の3レベルでの非圧縮性・非等温LES解析を行なった。
- 8ノード、128プロセッサでの、コンパイルオプションの検討や並列化効率を調査した。

まとめ

- HA8000においてOpenFOAMは問題なく動いた
- オプションは-O3 -xO -no-prec-divが概ね最速
- 8プロッセサ迄の並列化効率は0.8以上と良い
- 16プロッセサ以上では並列化効率は減少
- 290万、2300万格子数の規模では概ね0.5程度
- 効果の差はあるが、NUMA最適化は必須

今後の課題

- HA8000用のチューニング
- MPIだけではなく、OpenMPとのハイブリッド
- より多くのノードを使用した場合の検討
- より大きな問題での検討
- HA8000用OpenFOAMコンパイルキット公開