

# FOCUSスパコンにおける iconCFDの並列性能 ベンチマークテスト2

2017年11月10日

三菱電機株式会社 先端技術総合研究所  
熱流体システム技術部 流体・伝熱制御グループ  
永安 哲也

# 目次

1. 目的
2. FOCUSスパコンとは
3. ベンチマーク方法
4. 解析対象
5. 解析条件
6. 解析メッシュ
7. 解析ケース
8. 解析結果
9. ベンチマーク結果
10. まとめ

# 1. 目的

FOCUSスパコンのDシステム, Eシステム, Fシステム, Hシステムを使用し, オープンソースベースの汎用CFDソフトウェアiconCFDの並列性能ベンチマークテストを実施する.

# 2. FOCUSスパコンとは

スーパーコンピュータ「京」の産業利用の促進を図り産業界のスパコン利用企業層を拡大するための技術高度化支援を中心に供用を行うほか, 産学連携研究や実践的な企業技術者の育成を推進することを目的に整備された国内唯一の産業界専用の公的スーパーコンピュータ.

2011年4月に高度計算科学研究支援センター内で稼働を開始して以来, 利用者のニーズに応えるためシステムの増強を繰り返し実施し, 2017年8月現在、総理論演算性能は344テラフロップス(「京」の約1/30)に達している.

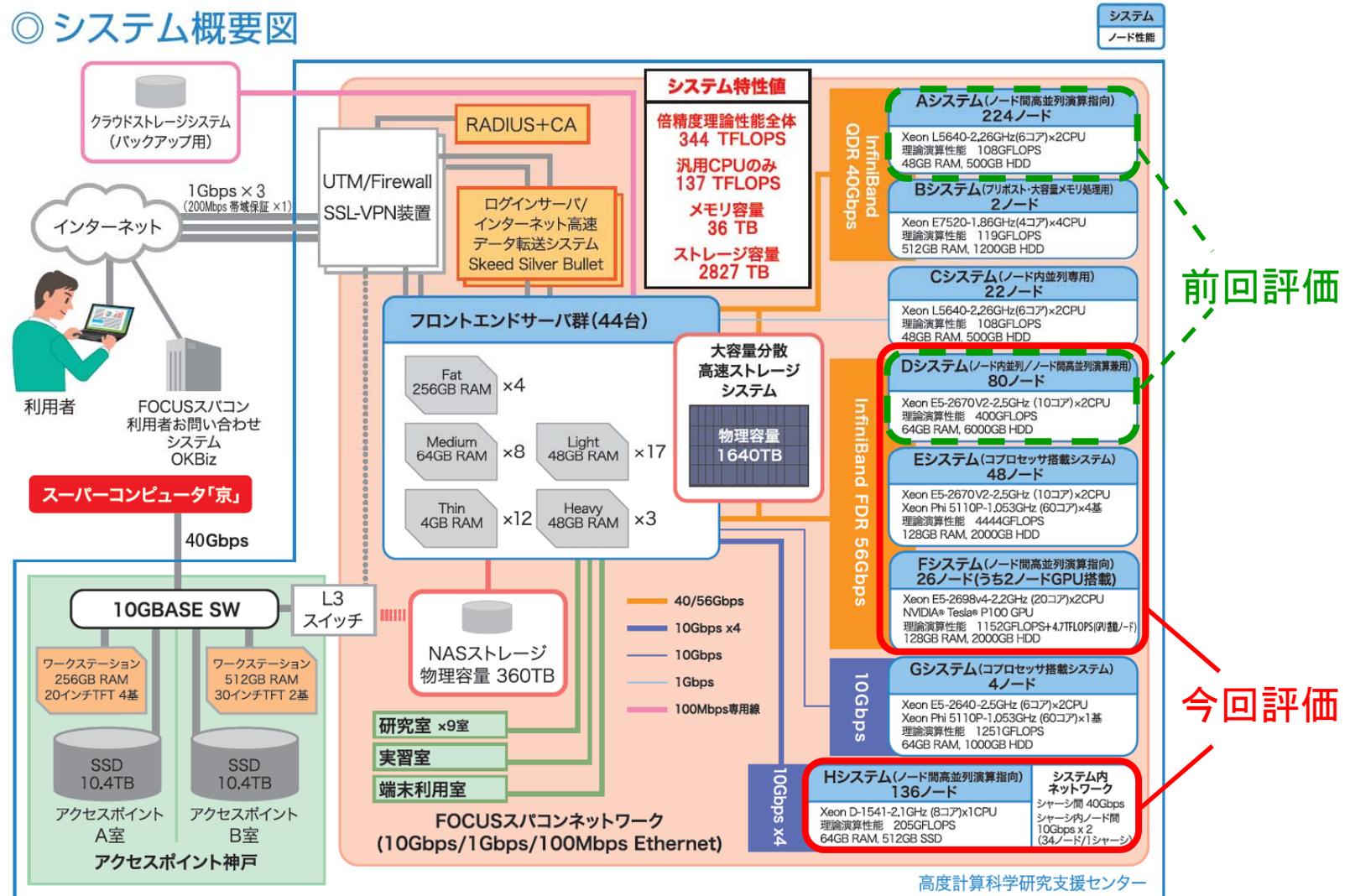


FOCUSスパコン

## 2. FOCUSスパコンとは

今回のベンチマークでは、ノード間の高並列演算を指向するDシステム、搭載メモリ量の多いEシステム、Fシステム、Hシステムを対象とした。

◎ システム概要図

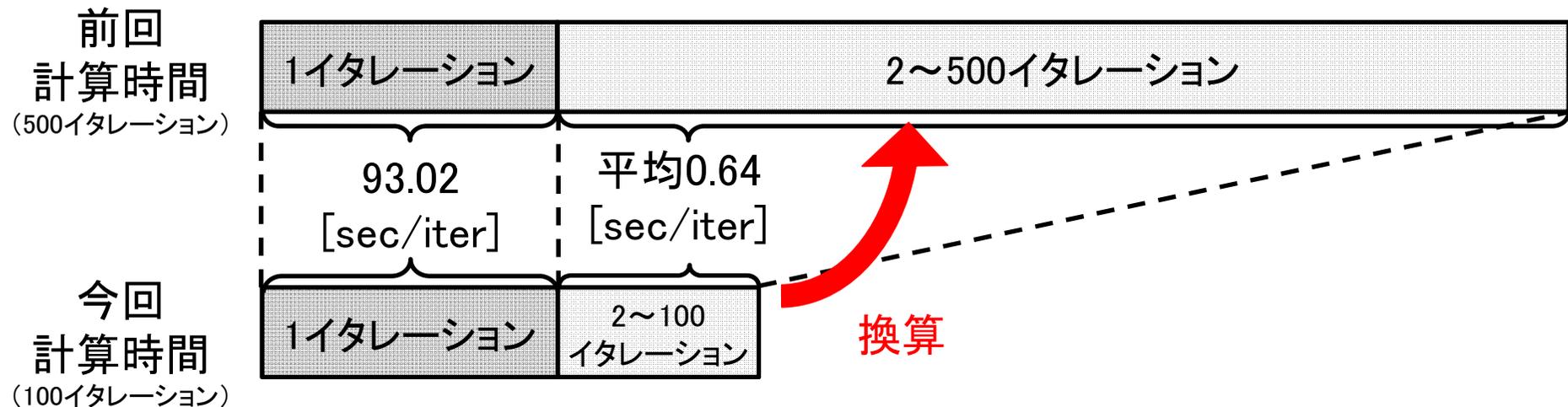


FOCUSスパコンのシステム概要図

### 3. ベンチマーク方法

同一形状の流体解析モデルをメッシュ数・並列数を変更し、それぞれ iconCFD による流体解析を実施する。今回は、500イタレーションまでの計算時間で評価したが、今回は、実行開始から100イタレーションまでで打ち切った。ただし、前回の結果との比較を可能にするため、今回の計算時間についても、500イタレーションまでの計算時間相当に換算して比較した。（※ 原則として、最終的な収束に達するまでの解析は行っていない。）

（例）Dシステムで、① Coarse Mesh(約2000万メッシュ)を960並列（フルコア使用）で計算した場合。（前回の結果より）



計算時間のイメージ

流体計算では、1イタレーション目の計算に非常に時間がかかる。また、100イタレーションと500イタレーションの計算時間を比較すると、単純に5倍ではない。

### 3. ベンチマーク方法

そのため、以下の式を使用し、100イタレーションまでの計算時間を500イタレーション相当の計算時間に換算して、比較を行った。

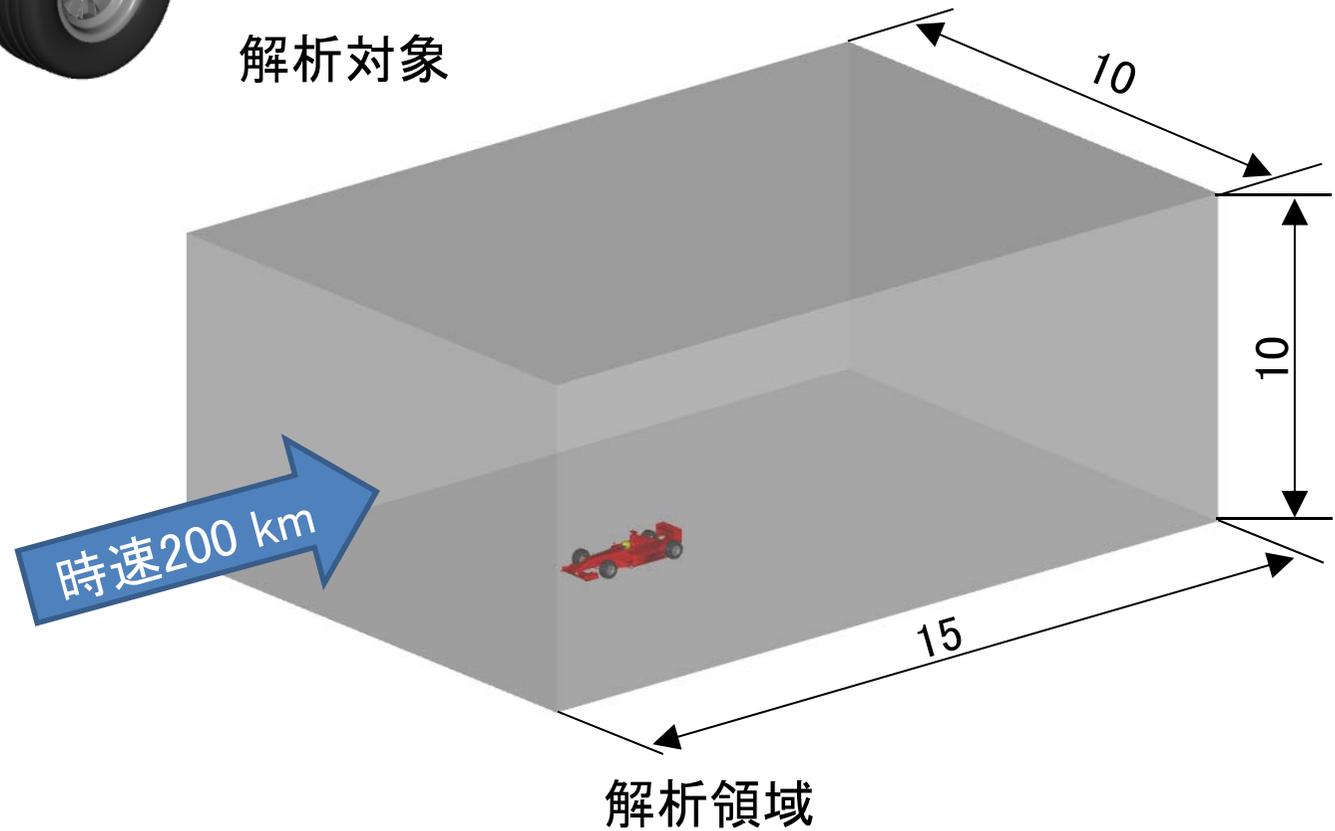
$$\text{計算時間} = (\text{1イタレーション目の計算時間}) + (\text{2~100イタレーションの平均計算時間}) \times 499$$

ここで、MPI(Message Passing Interface)は、iconCFDにバンドルされているintelMPIを使用した。環境は、通常運用中のFOCUSスパコン(Dシステム, Eシステム, Fシステム, Hシステム)とした。他のジョブの影響を小さくするため、原則3回の繰り返し測定を実施し、平均値を採用した。使用するiconCFDのバージョンは、前回との連続性から、v3.2.8(最新バージョンではない)とした。

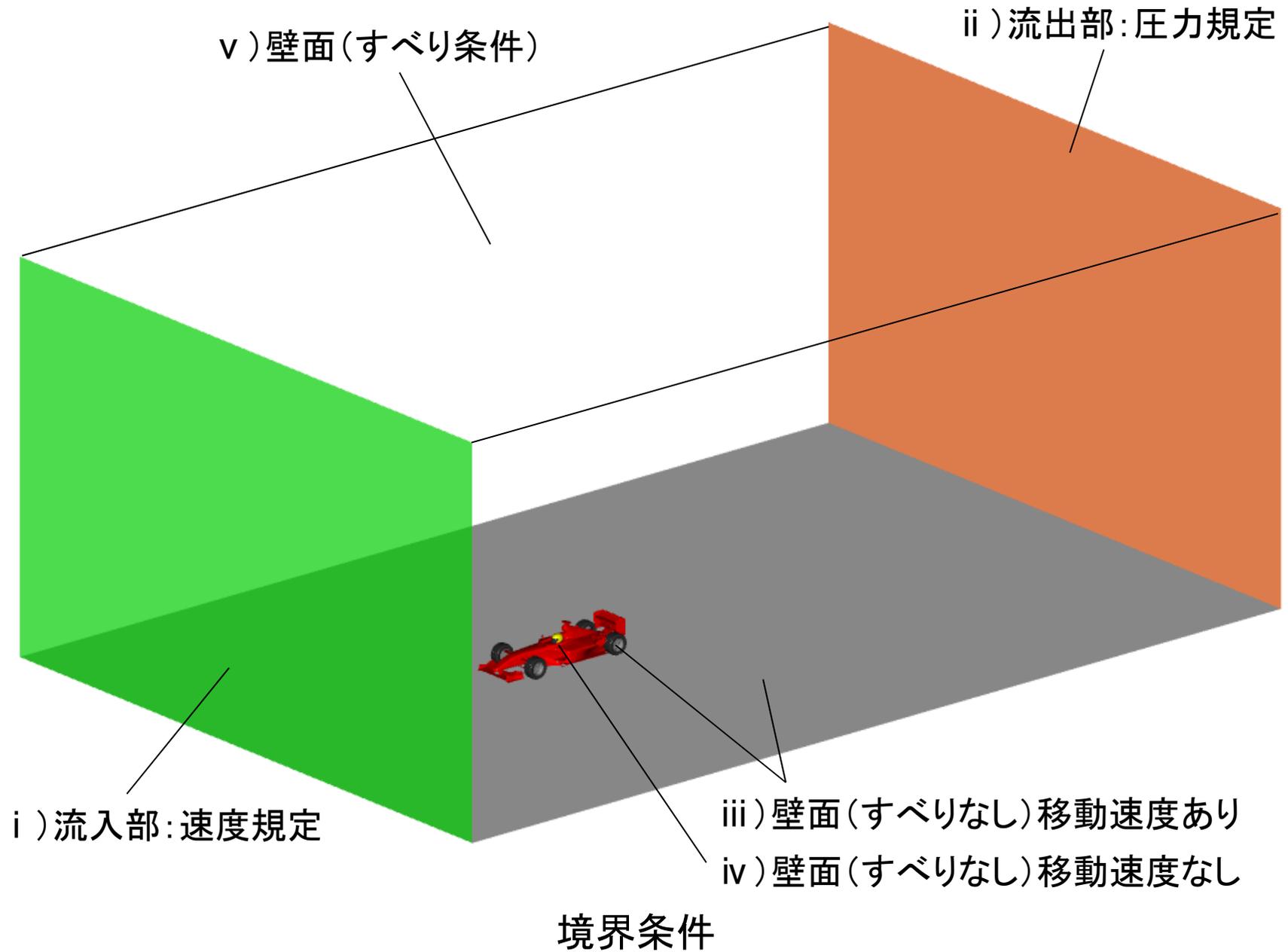
### 4. 解析対象

解析対象は、iconCFDの標準ベンチマークで用いられているレーシングカーモデルの外部流れとした。流れは、定常、非圧縮とし、乱流状態であるとした。200km/ hourの走行状態を仮定し、解析では、車体を固定、周囲流及び路面に相対的な速度を与えた。タイヤとホイールには、走行速度に合わせた回転速度を与えた。なお、流体の物性は、空気(密度: 1.205 [kg/m<sup>3</sup>], 動粘度: 1.511 × 10<sup>-5</sup> [m<sup>2</sup>/s])とし、乱流モデルは、k- $\omega$ SSTを使用した。

## 4. 解析対象

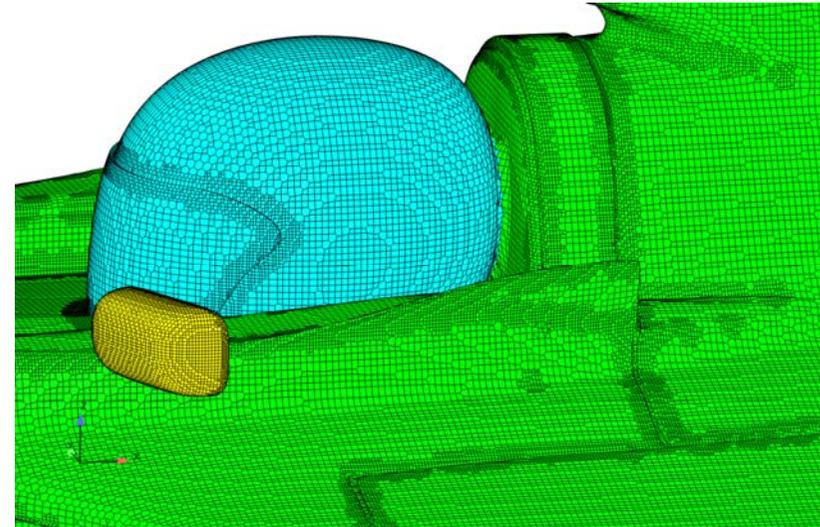
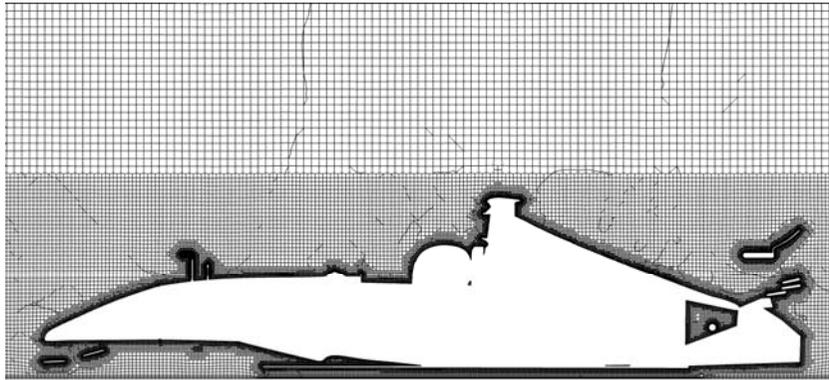


# 5. 解析条件

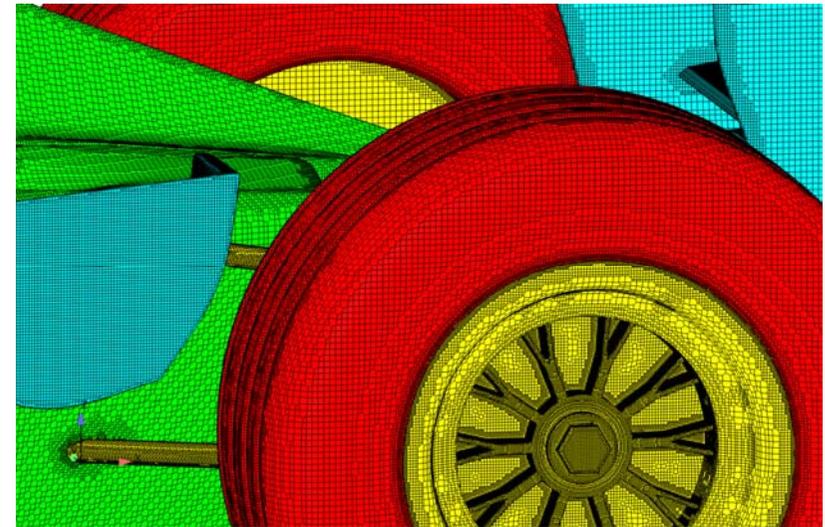
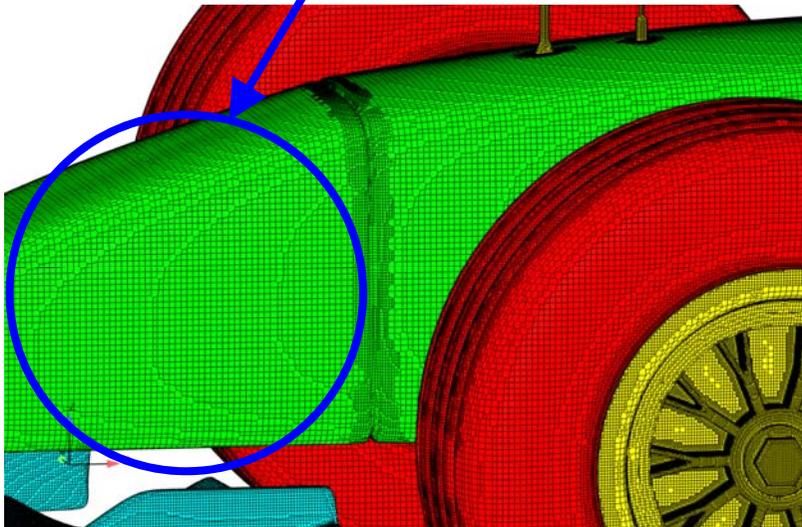


## 6. 解析メッシュ

① Coarse Mesh: 約2000万メッシュ(20,399,141メッシュ)

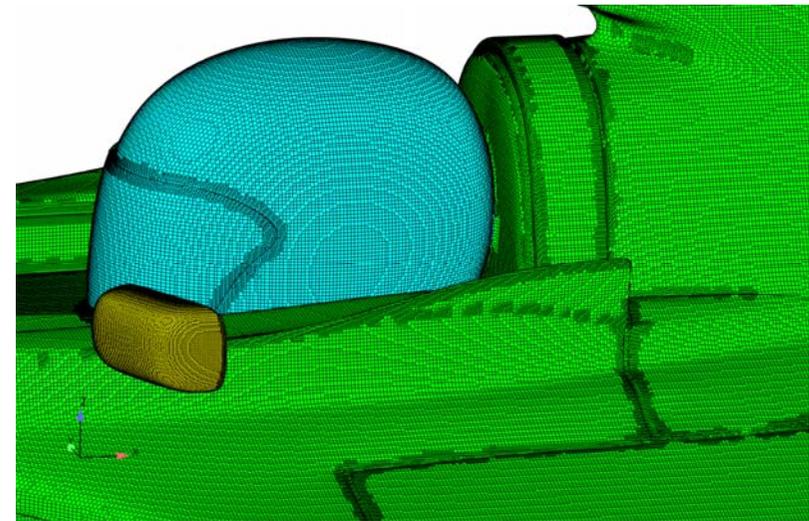
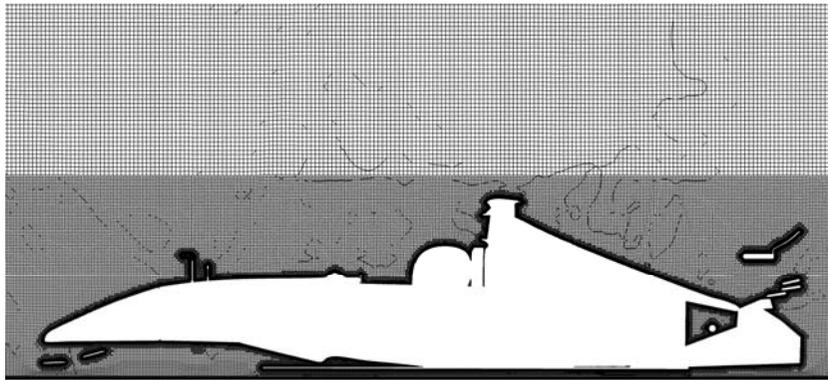


車体表面ベースサイズ:  
約2.8 [mm]

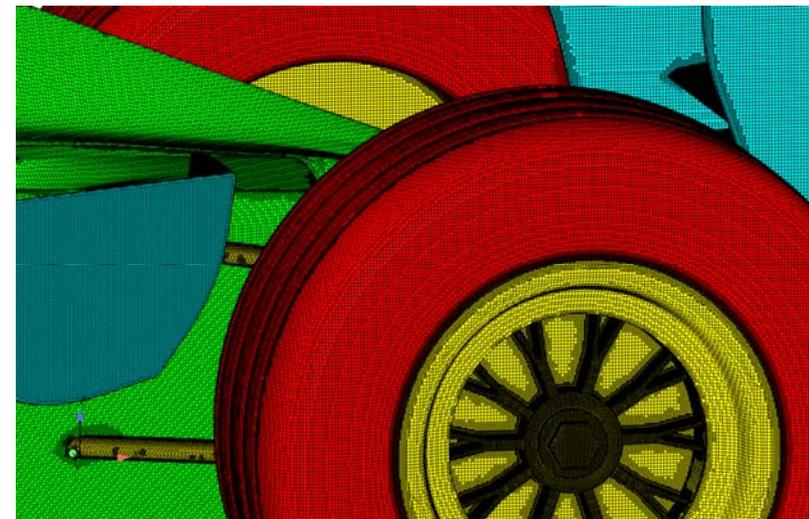
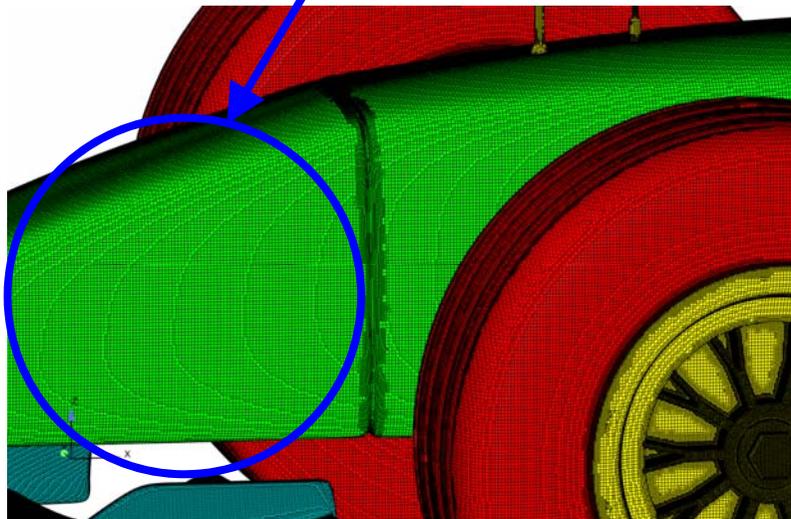


## 6. 解析メッシュ

② Standard Mesh: 約1億メッシュ (102,608,986メッシュ)

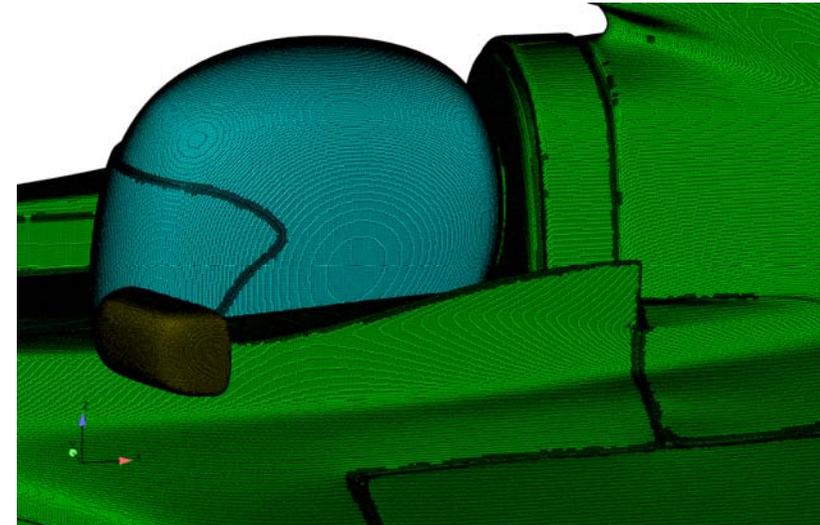
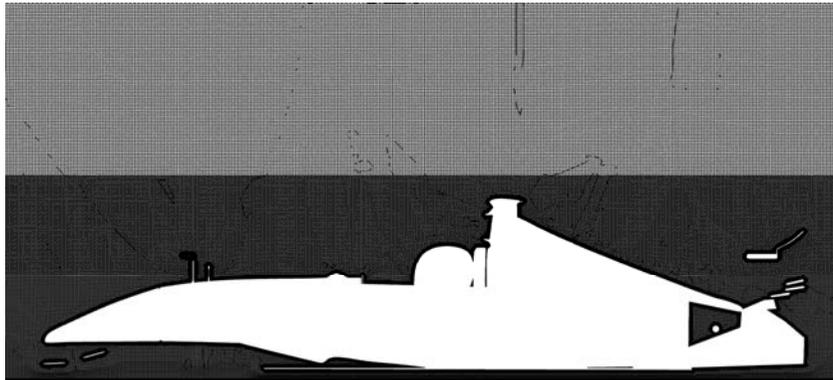


車体表面ベースサイズ:  
約1.6 [mm]

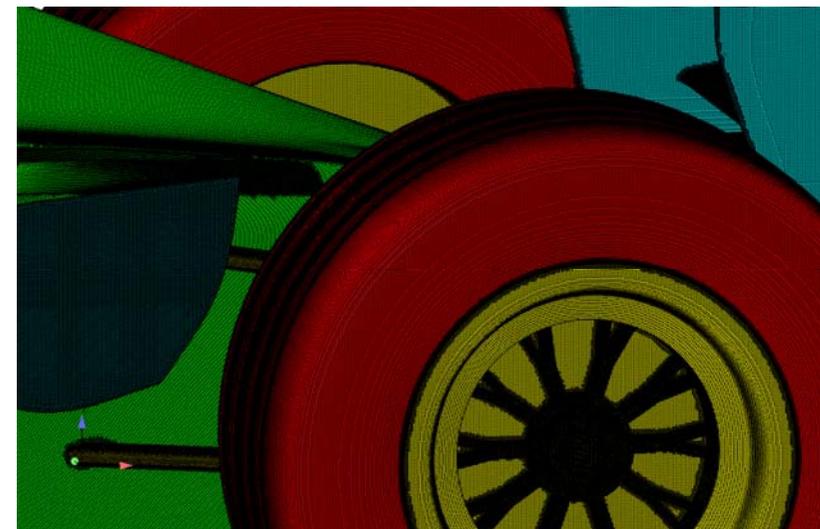
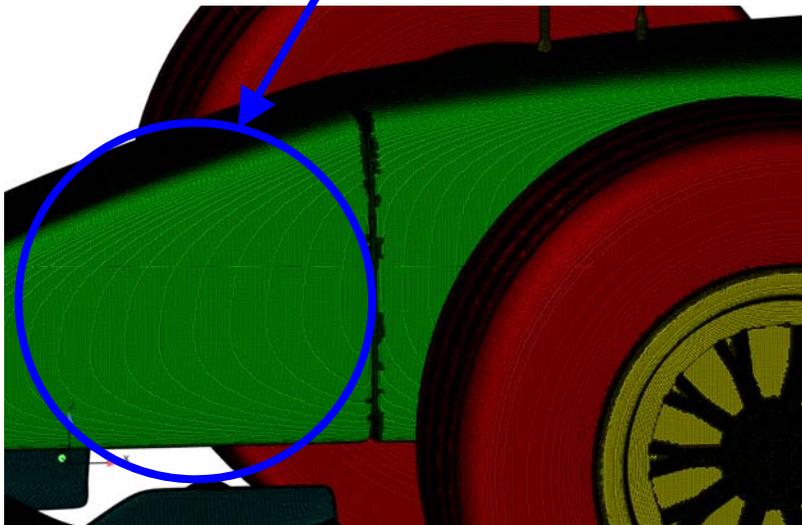


## 6. 解析メッシュ

③ Fine Mesh: 約5億メッシュ (501,384,138メッシュ)



車体表面ベースサイズ:  
約0.8 [mm]



# 7. 解析ケース

当初実施を計画した解析ケースの一覧を示す.

(i) システム比較 [E・F・Hシステム]

 メモリ不足の可能性大

Eシステム [20/20コア]	並列数	60	120	240	360	480	720	900	450
	ノード数	3	6	12	18	24	36	45	45
① Coarse Mesh		✓	✓	✓	✓	✓	✓	✓	✓
② Standard Mesh			✓	✓	✓	✓	✓	✓	✓
③ Fine Mesh					✓	✓	✓	✓	✓

Fシステム [40/40コア]	並列数	80	120	240	360	480	240
	ノード数	2	3	6	9	12	12
① Coarse Mesh		✓	✓	✓	✓	✓	✓
② Standard Mesh			✓	✓	✓	✓	✓
③ Fine Mesh							

← 半分のコアを使用(間引き)

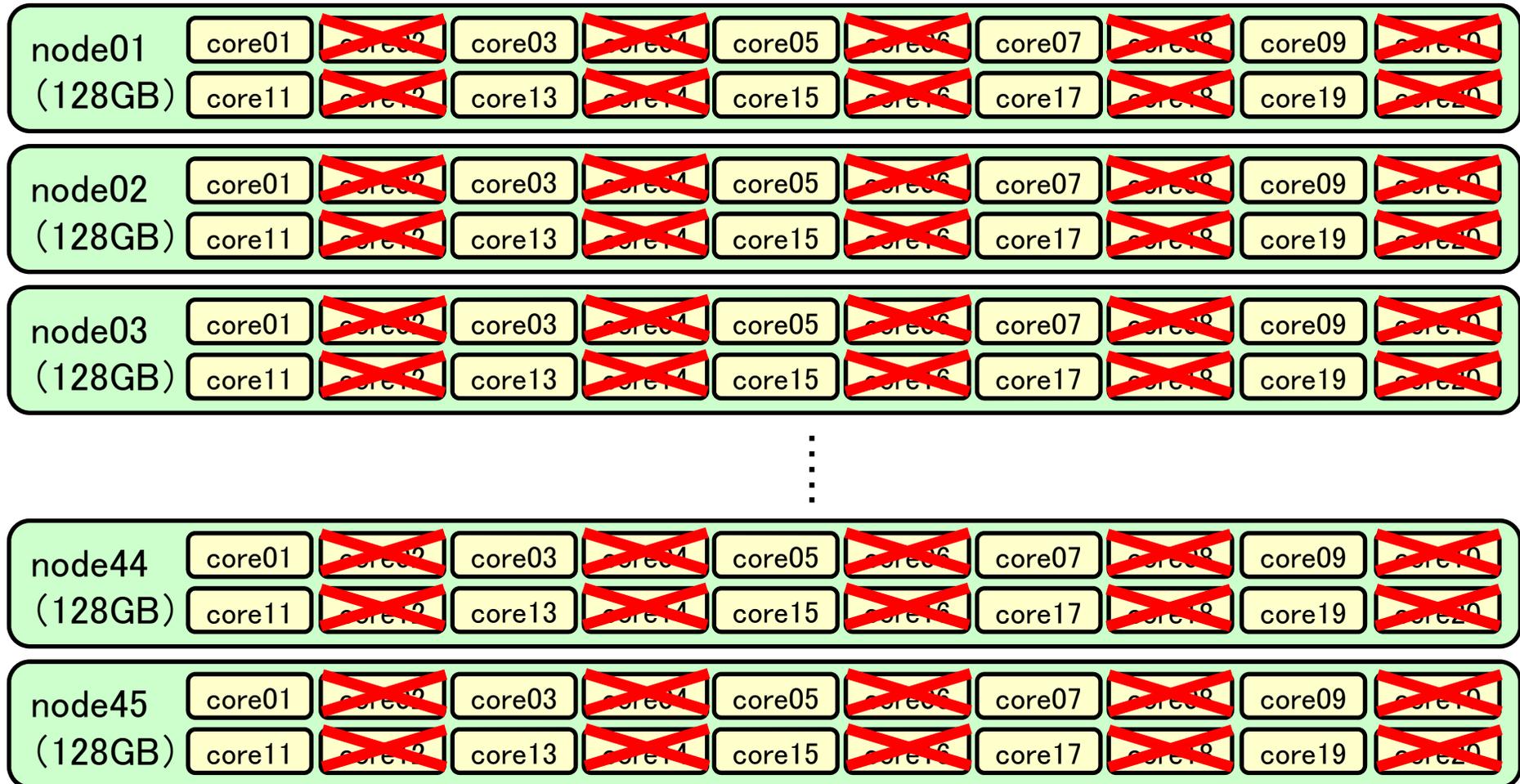
← 比較が困難なため計画なし

Hシステム [8/8コア]	並列数	64	128	256	384	512	768	1024	512
	ノード数	8	16	32	48	64	96	128	128
① Coarse Mesh		✓	✓	✓	✓	✓	✓	✓	✓
② Standard Mesh			✓	✓	✓	✓	✓	✓	✓
③ Fine Mesh				✓	✓	✓	✓	✓	✓

## 7. 解析ケース

(i) システム比較 [E・F・Hシステム(ここでは, Eシステムの例)]

最大並列数のケースのみ, 使用するノードの数は変えず, 並列数を半分に減らしたケースを追加で実施した. 1コア当たりのメモリ使用量を増加させる狙い.



450並列 (~~20~~ → 10コア/ノード × 45ノード)

## 7. 解析ケース

(ii) 半分間引きの影響検証 [Dシステムのみ]

 メモリ不足の可能性大

Dシステム [10/20コア]	並列数	60	120	180	240	360	480
	ノード数	6	12	18	24	36	48
① Coarse Mesh		✓	✓	✓	✓	✓	✓
② Standard Mesh			✓	✓	✓	✓	✓
③ Fine Mesh					✓	✓	✓

(iii) 間引きの最適化検証 [Dシステムのみ]

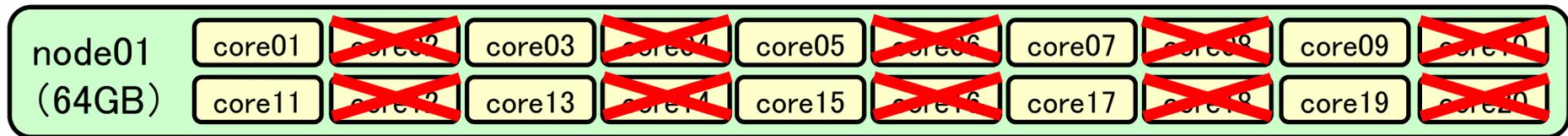
Dシステム (ノード数固定)	並列数	192	384	480	576	768	960
	ノード数	48 [4/20コア]	48 [8/20コア]	48 [10/20コア]	48 [12/20コア]	48 [16/20コア]	48 [20/20コア]
① Coarse Mesh		✓	✓	✓	✓	✓	✓
② Standard Mesh		✓	✓	✓	✓	✓	✓

Dシステム (並列数固定)	並列数	480	480	480	480	480
	ノード数	60 [8/20コア]	48 [10/20コア]	40 [12/20コア]	30 [16/20コア]	24 [20/20コア]
① Coarse Mesh		✓	✓	✓	✓	✓
② Standard Mesh		✓	✓	✓	✓	✓

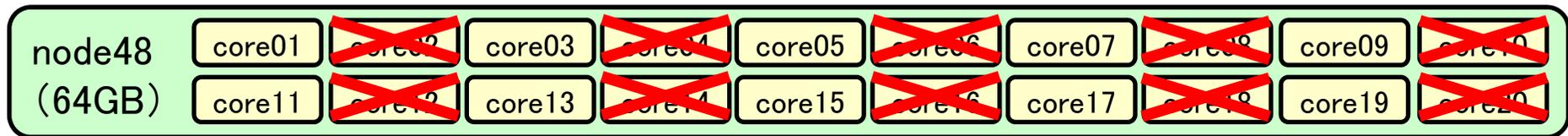
# 7. 解析ケース

## (ii) 半分間引きの影響検証 [Dシステム]

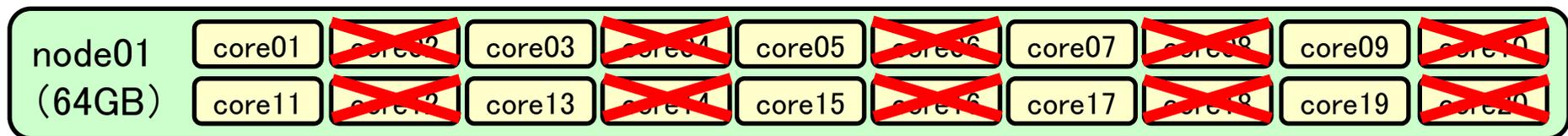
前回のベンチマークでは、使用するノード数(48)を変更せず、コア数を半分にすると計算速度が向上したが、他の(少ない)ノード数(6~36)でも向上するか？



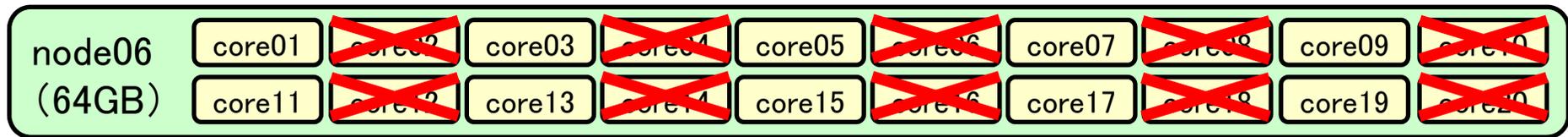
⋮



~~960~~ → 480 並列 (~~20~~ → 10コア/ノード × 48ノード) → 計算速度向上



⋮

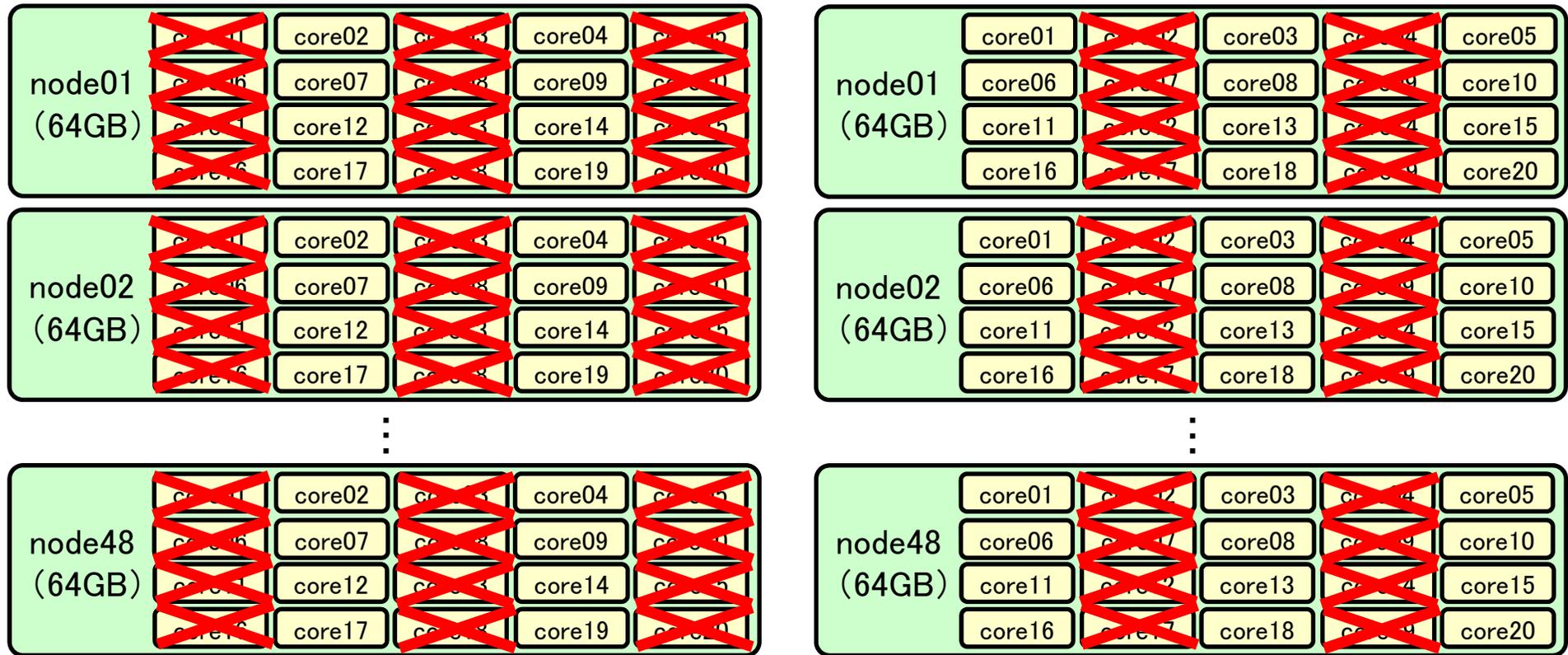


~~120~~ → 60 並列 (~~20~~ → 10コア/ノード × 6ノード) → 計算速度向上？

# 7. 解析ケース

(iii) 間引きの最適化検証 [Dシステム]

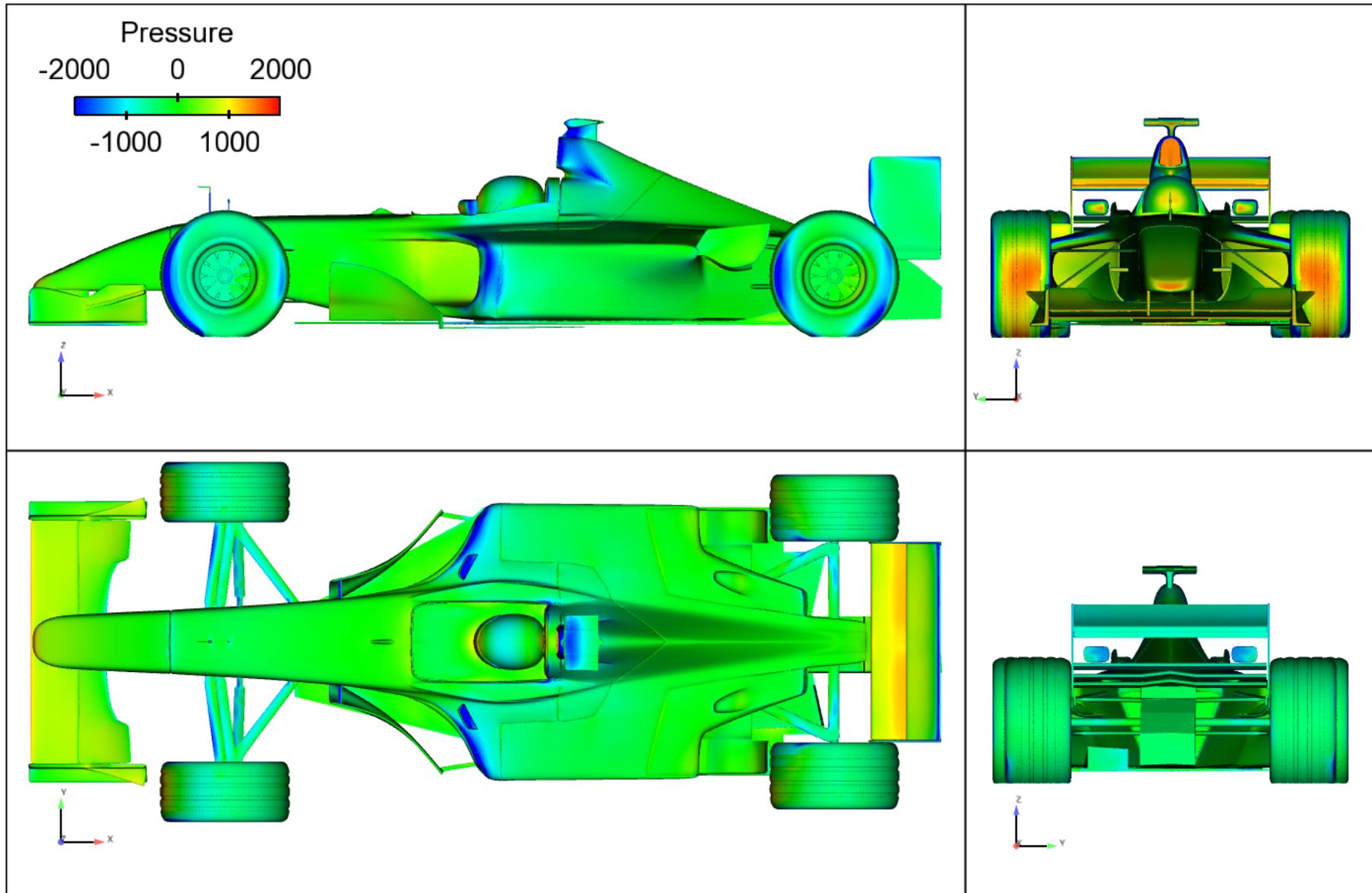
同じノード数で間引きを変えると計算時間はどの程度変化するか？  
同じ並列数でノード数を変えると計算時間はどの程度変化するか？



~~960~~ → 384 並列 (20 → 8 コア/ノード × 48 ノード)      ⇔      ~~960~~ → 576 並列 (20 → 12 コア/ノード × 48 ノード)  
どちらが速い？

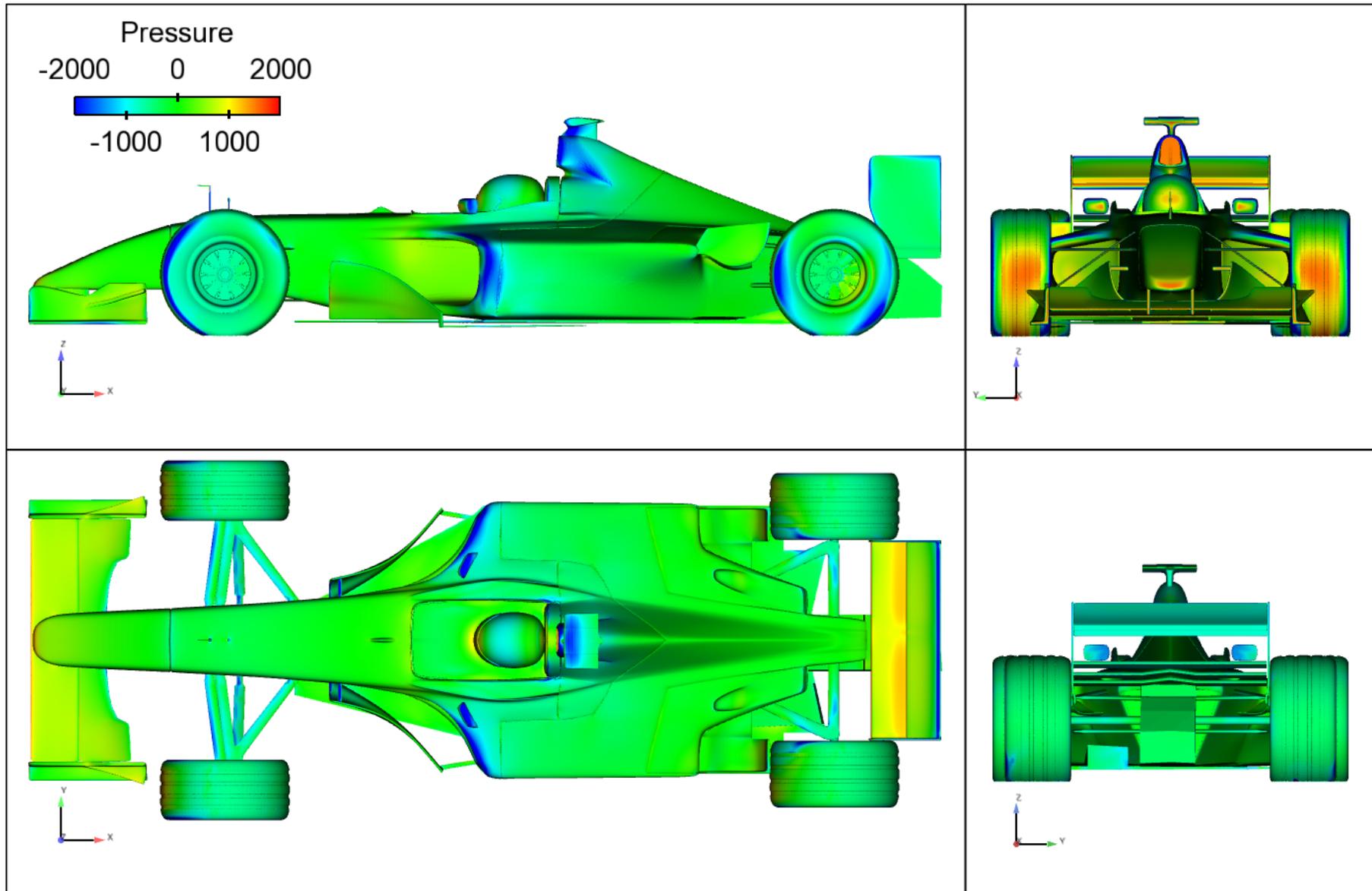
## 8. 解析結果(圧力分布)

① Coarse Mesh: 約2000万メッシュ(20,399,141メッシュ)



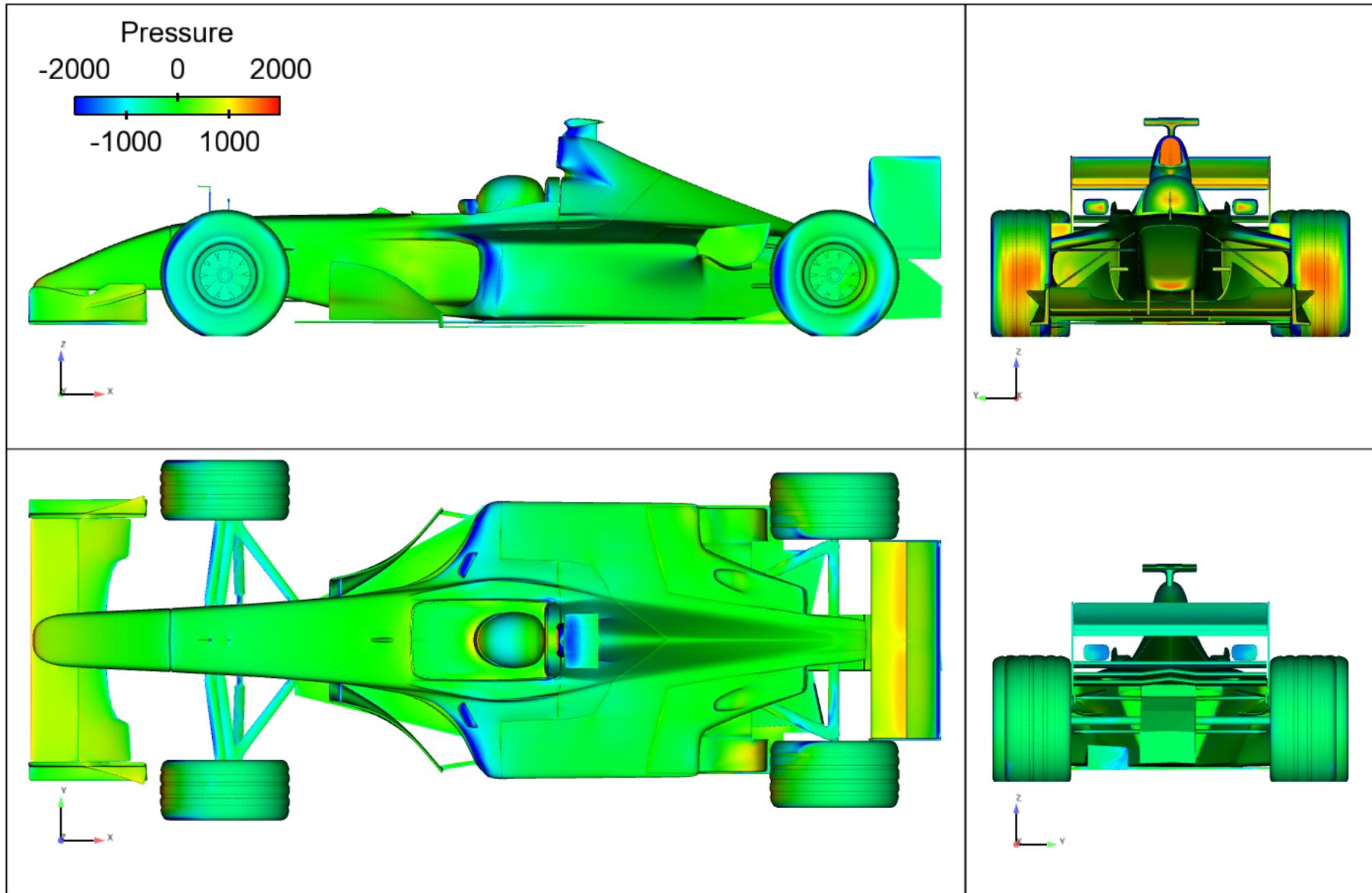
## 8. 解析結果(圧力分布)

② Standard Mesh: 約1億メッシュ(102,608,986メッシュ)



## 8. 解析結果(圧力分布)

③ Fine Mesh: 約5億メッシュ(501,384,138メッシュ)



# 9. ベンチマーク結果

(カッコ)内は、各基準に対する計算速度を示す  
(スケーラビリティ)

(i) システム比較 [E・Fシステム]

計算時間(実行開始から500イタレーション相当として換算[単位:sec])

Eシステム [20/20コア]	並列数	60	120	240	360	480	720	900	450
	ノード数	3	6	12	18	24	36	45	45
① Coarse Mesh (約2000万メッシュ)		2699 (0.5)	1422 (1.0)	741.2 (1.9)	521.7 (2.7)	473.4 (3.0)	426.5 (3.3)	423.4 (3.4)	244.7 (5.8)
② Standard Mesh (約1億メッシュ)			7386 (1.0)	3763 (2.0)	2852 (2.6)	2312 (3.2)	1743 (4.2)	1476 (5.0)	1170 (6.3)
③ Fine Mesh (約5億メッシュ)					15637 (1.0)	11093 (1.4)	9491 (1.6)	8248 (1.9)	6251 (2.5)

メモリ不足で実行不可 スケーラビリティ算出の基準

Fシステム [40/40コア]	並列数	80	120	240	360	480	240
	ノード数	2	3	6	9	12	12
① Coarse Mesh (約2000万メッシュ)		2442 (0.7)	1693 (1.0)	874.8 (1.9)	587.0 (2.9)	499.3 (3.4)	435.2 (3.9)
② Standard Mesh (約1億メッシュ)			9132 (1.0)	4563 (2.0)	3322 (2.7)	2661 (3.4)	2324 (3.9)

半分のコアを使用(間引き)

# 9. ベンチマーク結果

(カッコ)内は、各基準に対する計算速度を示す  
(スケーラビリティ)

(i) システム比較 [Hシステム]

計算時間(実行開始から500イタレーション相当として換算[単位:sec])

原則3回の繰り返し測定を行う予定であったが、1回目の計算で時間がかかりすぎたため、2回目の計算以降を中止

Hシステム [8/8コア]	並列数	64	128	256	384	512	768	1024	512
	ノード数	8	16	32	48	64	96	128	128
① Coarse Mesh (約2000万メッシュ)	4197 (0.7)	3086 (1.0)	2339 (1.3)	7459 (0.4)	39838 (0.1)	414186 (0.01)			
② Standard Mesh (約1億メッシュ)		15467 (1.0)	16204 (1.0)	18116 (0.9)					
③ Fine Mesh (約5億メッシュ)									

メモリ不足で実行不可

スケーラビリティ算出の基準

Hシステムでは、並列数が384を超えると、計算時間が大幅に増加する傾向が見られた。スケーラビリティも大幅に低下したため、以降の計算(③ Fine Meshなど)を中止した。

# 9. ベンチマーク結果

(ii) 半分間引きの影響検証 [Dシステム]

スケーラビリティ  
算出の基準

(カッコ)内は、各基準に  
対する計算速度を示す  
(スケーラビリティ)

計算時間(実行開始から500イタレーション相当として換算[単位:sec])

メモリ不足で  
実行不可

(参考)  
前回実施

Dシステム	並列数	60	120	180	240	360	480
[10/20コア]	ノード数	6	12	18	24	36	48
① Coarse Mesh (約2000万メッシュ)		1460 (0.5)	767.5 (1.0)	499.9 (1.5)	390.0 (2.0)	274.1 (2.8)	241.9 (3.2)
② Standard Mesh (約1億メッシュ)		メモリ不足	3948 (1.0)	2625 (1.5)	1986 (2.0)	1478 (2.7)	1180 (3.3)
③ Fine Mesh (約5億メッシュ)		メモリ不足	メモリ不足	メモリ不足	11059 (1.0)	8300 (1.3)	5871 (1.9)

Dシステム	並列数	60	120	240	360	480	720	960
[20/20コア]	ノード数	3	6	12	18	24	36	48
① Coarse Mesh (約2000万メッシュ)		2740 (0.3)	1413 (0.5)	720.3 (1.0)	526.5 (1.4)	428.3 (1.7)	413.6 (1.7)	392.9 (1.8)
② Standard Mesh (約1億メッシュ)		18801 (0.2)	7442 (0.5)	3770 (1.0)	2747 (1.4)	2130 (1.8)	1645 (2.3)	1538 (2.5)
③ Fine Mesh (約5億メッシュ)		メモリ不足	メモリ不足	メモリ不足	12532 (0.8)	9745 (1.0)	メモリ不足	5944 (1.6)

# 9. ベンチマーク結果

(カッコ)内は、各基準に対する計算速度を示す  
(スケーラビリティ)

## (iii) 間引きの最適化検証 [Dシステム]

計算時間(実行開始から500イタレーション相当として換算[単位:sec])

### 1 ノード数固定(料金一定)

Dシステム	並列数	192	384	480	576	768	960
	ノード数	48 [4/20コア]	48 [8/20コア]	48 [10/20コア]	48 [12/20コア]	48 [16/20コア]	48 [20/20コア]
① Coarse Mesh (約2000万メッシュ)	337.5 (1.2)	266.8 (1.5)	241.9 (1.6)	250.8 (1.6)	271.3 (1.4)	392.9 (1.0)	
② Standard Mesh (約1億メッシュ)	1800 (0.9)	1358 (1.1)	1180 (1.3)	1162 (1.3)	1270 (1.2)	1538 (1.0)	

### 2 並列数固定

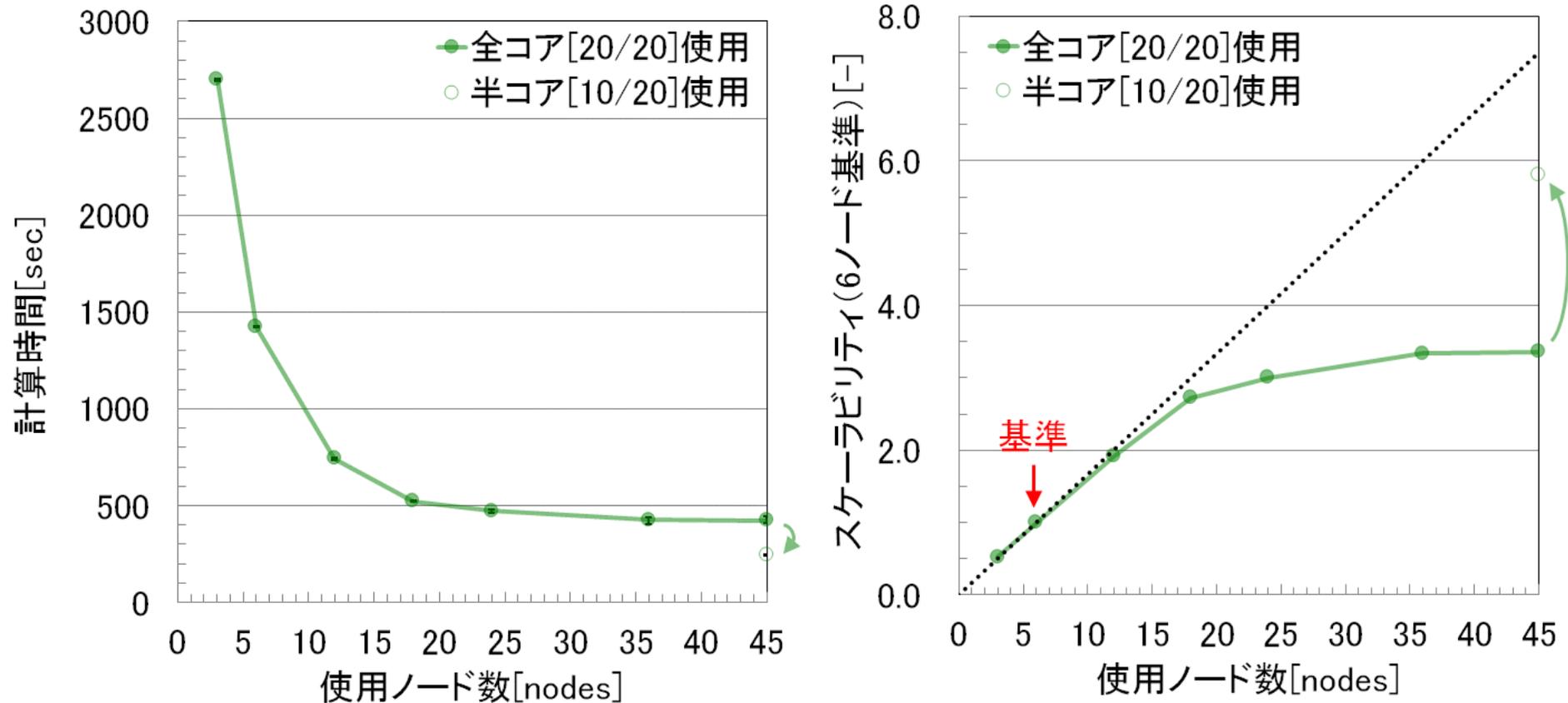
計算時間が最も短い

Dシステム	並列数	480	480	480	480	480
	ノード数	60 [8/20コア]	48 [10/20コア]	40 [12/20コア]	30 [16/20コア]	24 [20/20コア]
① Coarse Mesh (約2000万メッシュ)	231.2 (1.9)	241.9 (1.8)	261.4 (1.6)	313.3 (1.4)	428.3 (1.0)	
② Standard Mesh (約1億メッシュ)	1154 (1.8)	1180 (1.8)	1325 (1.6)	1656 (1.3)	2130 (1.0)	

スケーラビリティ  
算出の基準  
(全コア使用)

## 9. ベンチマーク結果

(i) システム比較 [Eシステム] ① Coarse Mesh: 約2000万メッシュ



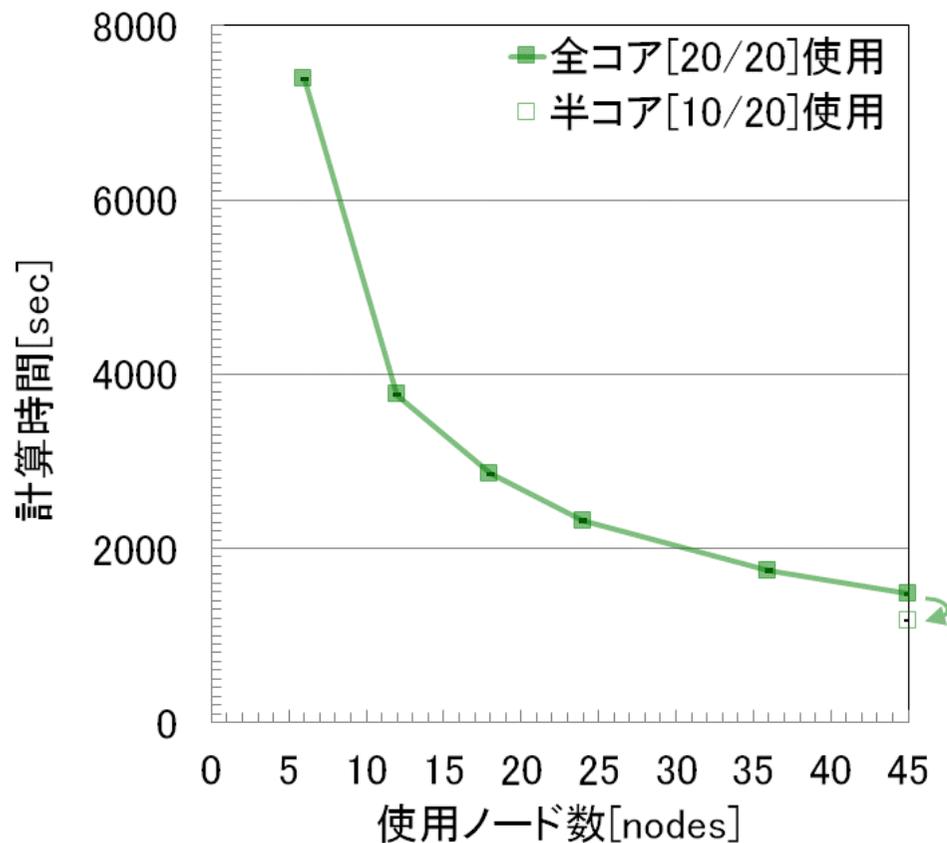
使用ノード数と計算時間の関係

使用ノード数とスケーラビリティの関係

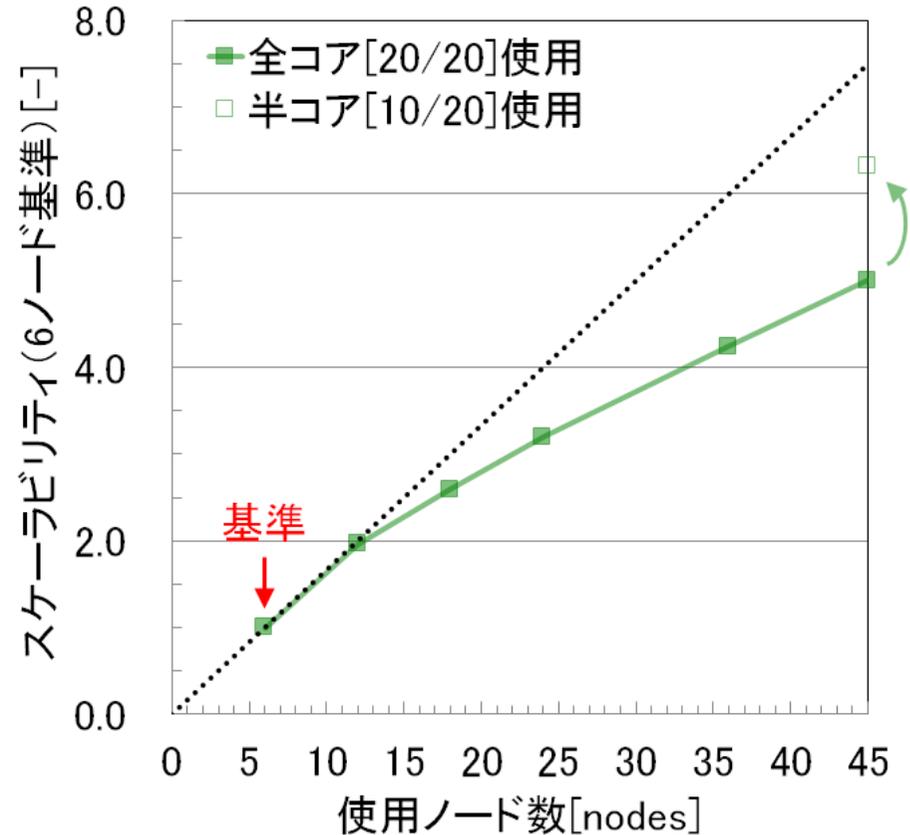
- ・Eシステムの① Coarse Meshでは、480並列(24ノード使用)以上で計算時間はほぼ一定となった。各ケースとも計算時間のばらつきは、非常に少ない。
- ・45ノード使用時に、使用するコア数を半分に間引くことで、並列効率は大幅に向上した(6ノード使用時を基準として、45→77%)。

## 9. ベンチマーク結果

(i) システム比較 [Eシステム] ② Standard Mesh: 約1億メッシュ



使用ノード数と計算時間の関係

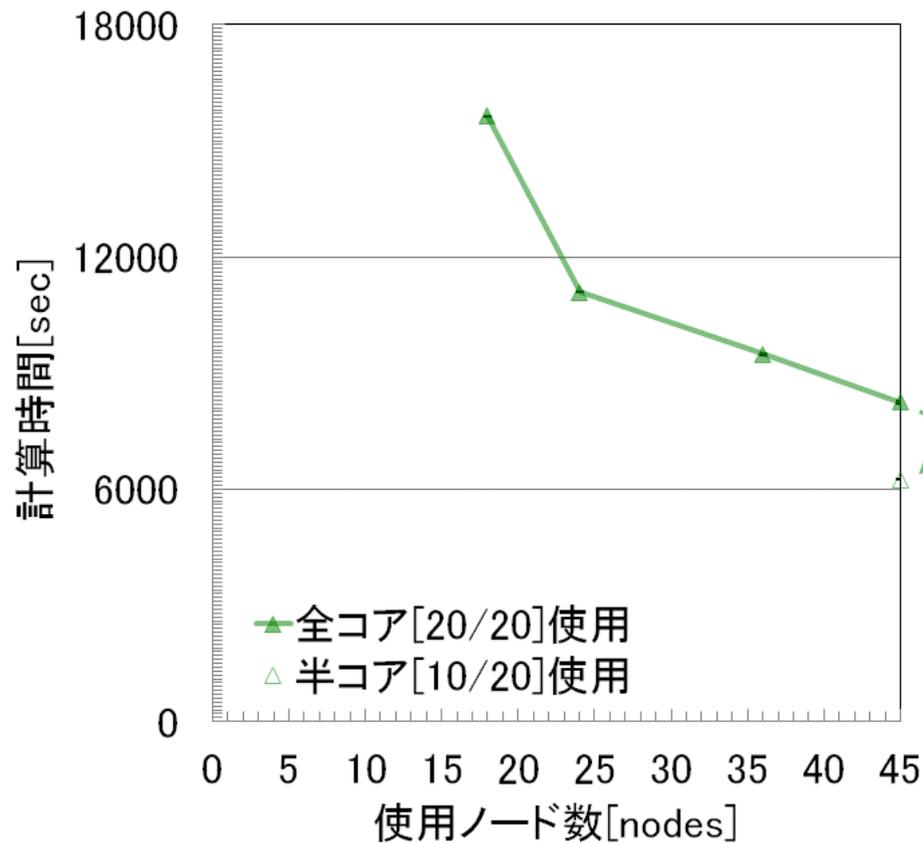


使用ノード数とスケーラビリティの関係

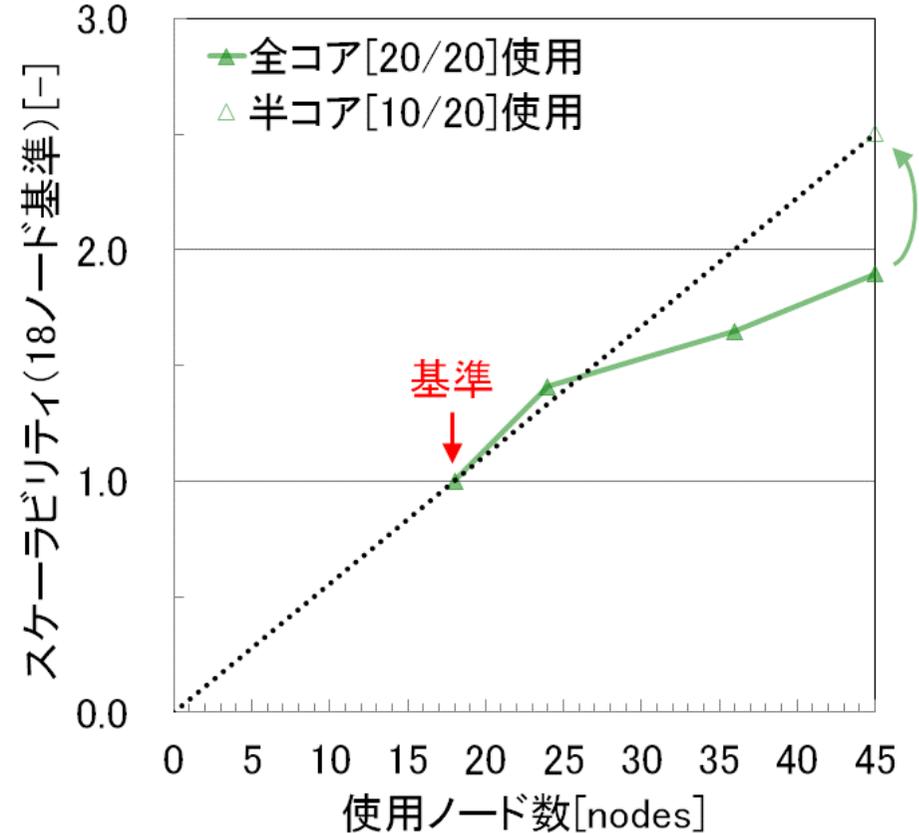
- ・Eシステムの② Standard Meshでも、並列数の増加に伴い、計算時間は短くなった。
- ・45ノード使用時に、使用するコア数を半分に間引くことで、並列効率は向上した (6ノード使用時を基準として、67→84%)。

## 9. ベンチマーク結果

(i) システム比較 [Eシステム] ③ Fine Mesh: 約5億メッシュ



使用ノード数と計算時間の関係

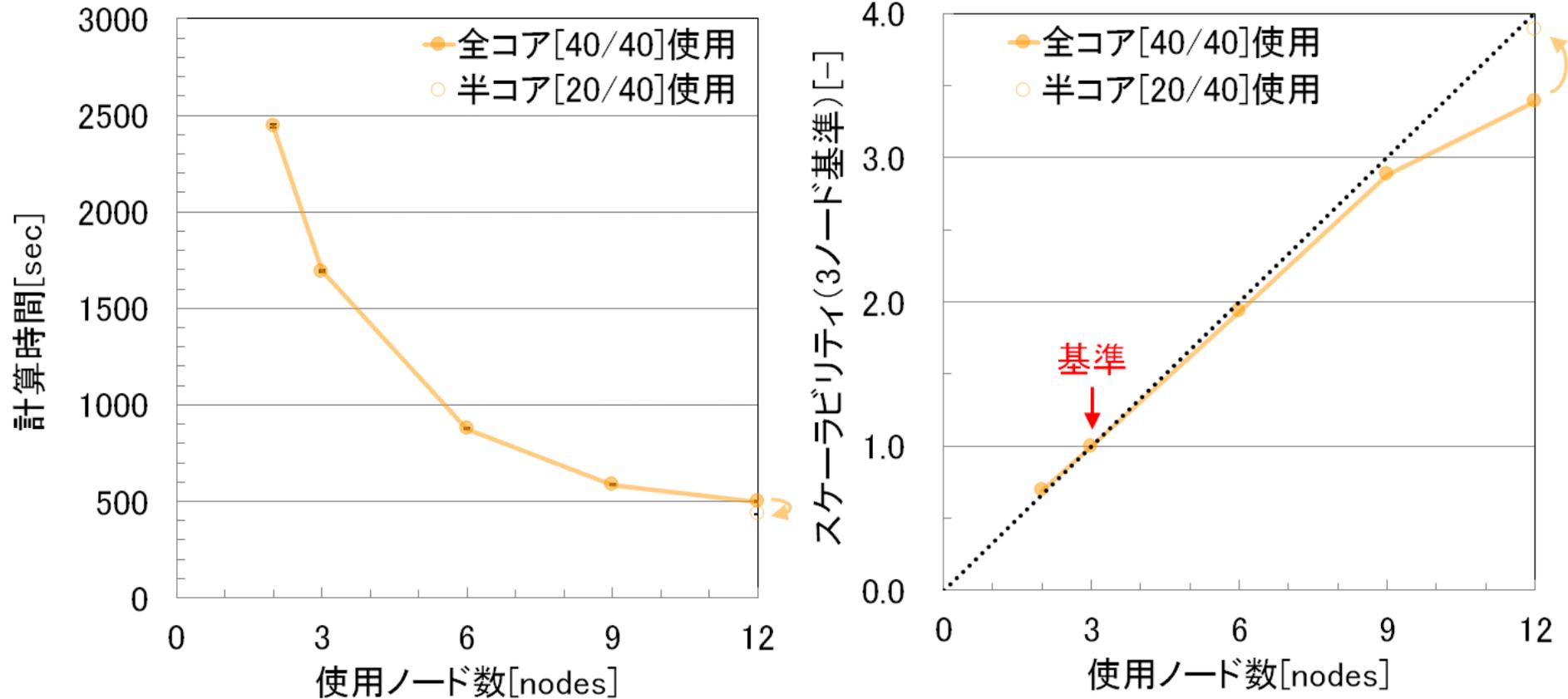


使用ノード数とスケーラビリティの関係

- ・Eシステムの③ Fine Meshでも、並列数の増加に伴い、計算時間は短くなった。
- ・45ノード使用時に、使用するコア数を半分に間引くことで、並列効率は大幅に向上した(18ノード使用時を基準として、76→100%)。

## 9. ベンチマーク結果

(i) システム比較 [Fシステム] ① Coarse Mesh: 約2000万メッシュ



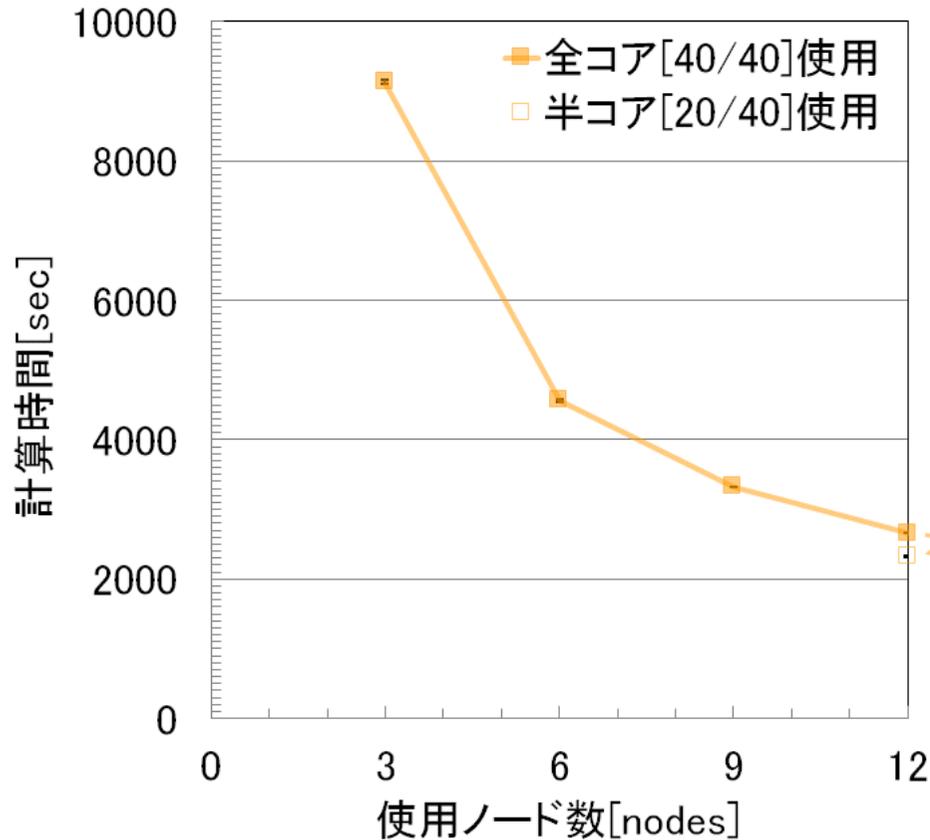
使用ノード数と計算時間の関係

使用ノード数とスケーラビリティの関係

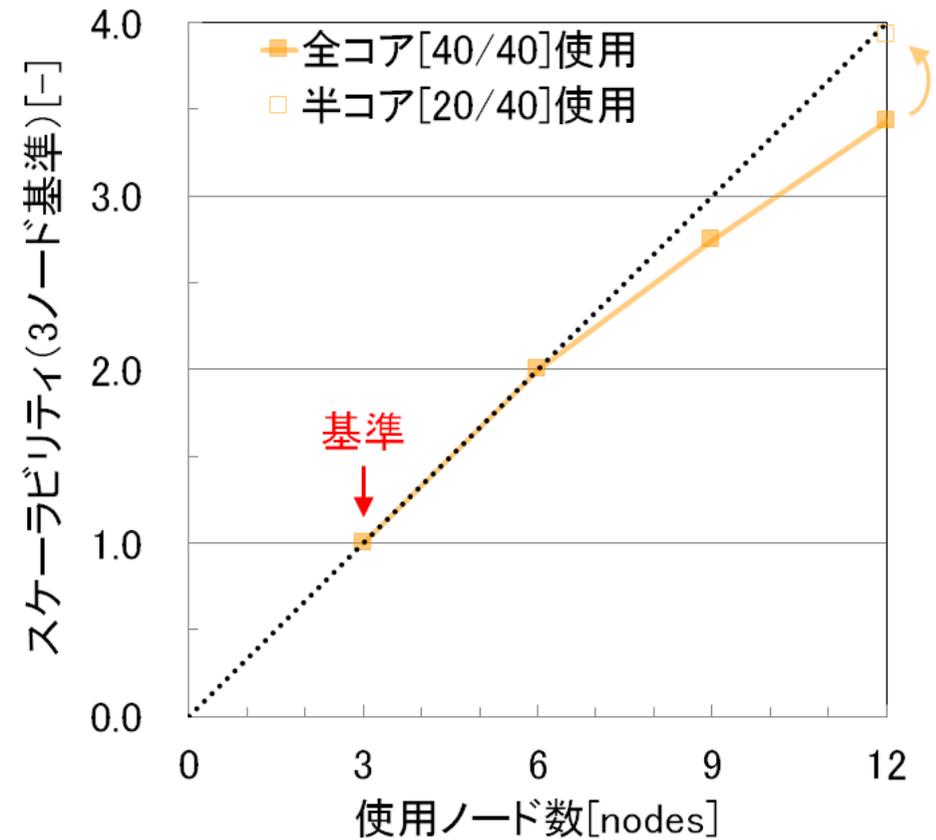
- ・Fシステムの① Coarse Meshでも、並列数の増加に伴い、計算時間は短くなった。計算時間のばらつきも少ない。
- ・12ノード使用時に、使用するコア数を半分に間引くことで、並列効率はやや改善した(3ノード使用時を基準として、85→97%)。

## 9. ベンチマーク結果

(i) システム比較 [Fシステム] ② Standard Mesh: 約1億メッシュ



使用ノード数と計算時間の関係

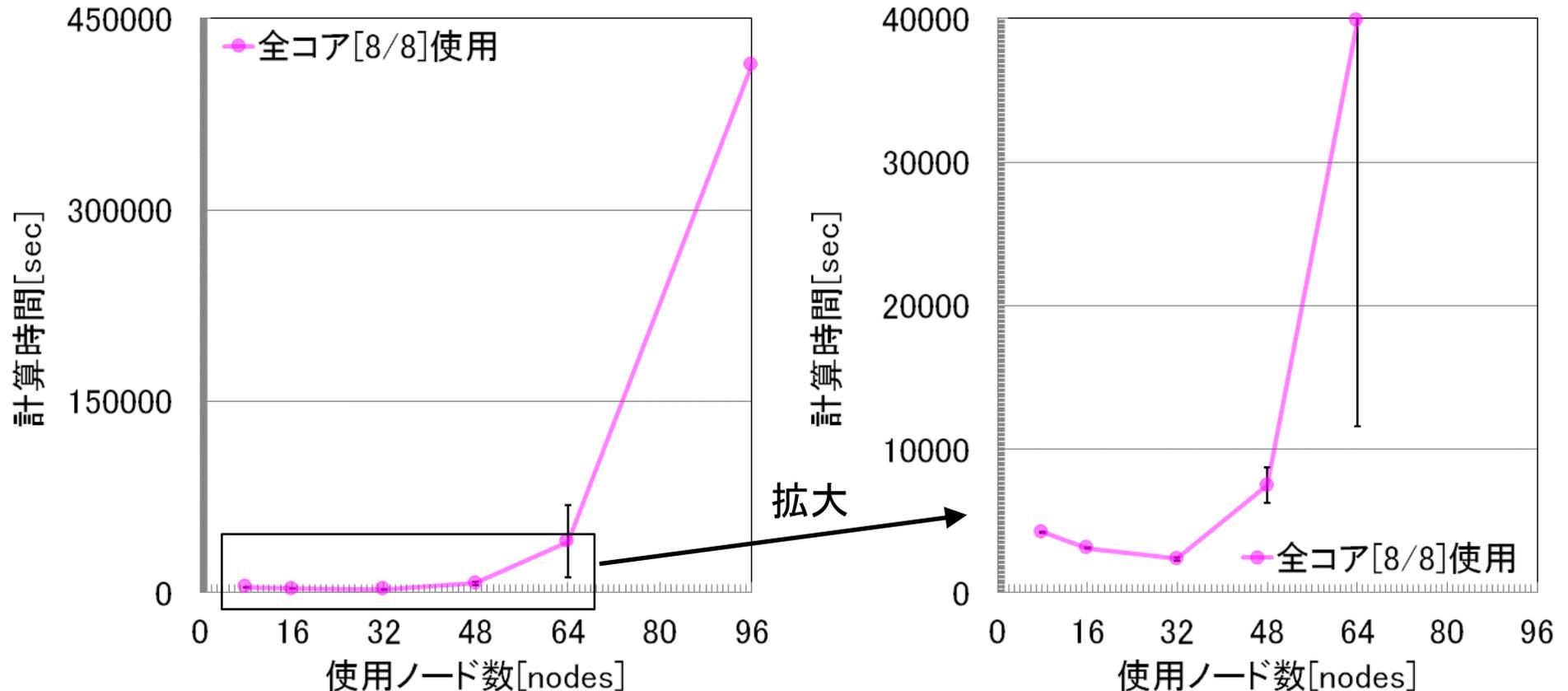


使用ノード数とスケーラビリティの関係

- ・Fシステムの② Standard Meshでも, 並列数の増加に伴い, 計算時間は安定的に短くなった.
- ・12ノード使用時に, 使用するコア数を半分に間引くことで, 同じく並列効率はやや改善した(3ノード使用時を基準として, 86→98%).

## 9. ベンチマーク結果

(i) システム比較 [Hシステム] ① Coarse Mesh: 約2000万メッシュ



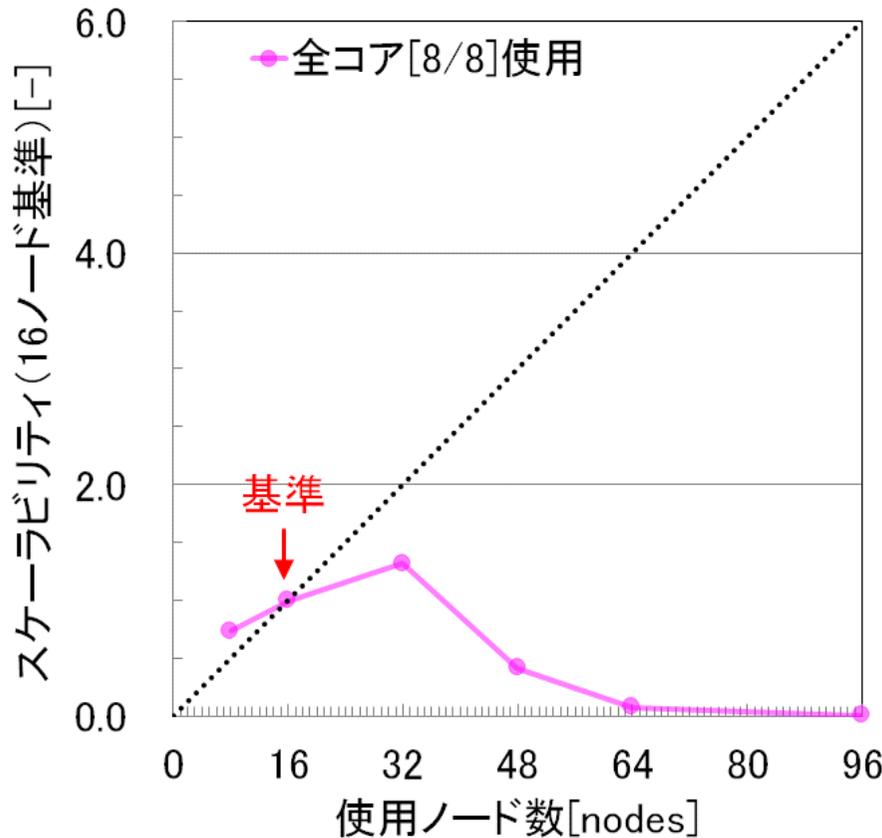
使用ノード数と計算時間の関係

使用ノード数と計算時間の関係(拡大)

- ・Hシステムの① Coarse Meshでは, 256並列(32ノード使用)までは計算時間が短くなるが, 384並列(48ノード使用)以上では計算時間は長くなった.
- ・512並列(64ノード使用)では, 計算時間のばらつきも大きい. 768並列(96ノード使用)の計算は1回のみ(2回目以降は中止).

# 9. ベンチマーク結果

(i) システム比較 [Hシステム] ① Coarse Mesh: 約2000万メッシュ



使用ノード数とスケールラビリティの関係

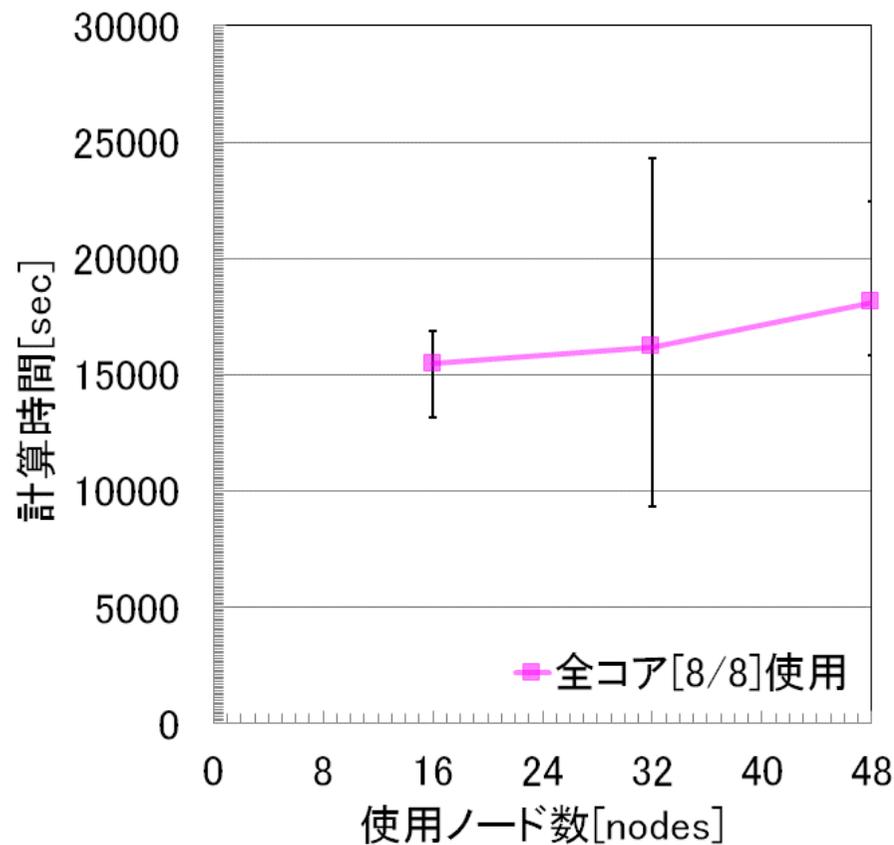
特性値	システム構成	性能
性能全体 LOPS のみ LOPS 容量 TB	<b>Aシステム</b> (ノード間高並列演算指向) 224ノード InfiniBand QDR 40Gbps	Xeon L5640-2.26GHz(6コア)×2CPU 理論演算性能 108GFLOPS 48GB RAM, 500GB HDD
	<b>Bシステム</b> (フリホスト-大容量メモリ処理用) 2ノード	Xeon E7520-1.86GHz(4コア)×4CPU 理論演算性能 119GFLOPS 512GB RAM, 1200GB HDD
	<b>Cシステム</b> (ノード内並列専用) 22ノード	Xeon L5640-2.26GHz(6コア)×2CPU 理論演算性能 108GFLOPS 48GB RAM, 500GB HDD
量分散 ストレージ システム 容量 40TB	<b>Dシステム</b> (ノード内並列/ノード間高並列演算兼用) 80ノード InfiniBand FDR 56Gbps	Xeon E5-2670V2-2.5GHz (10コア)×2CPU 理論演算性能 400GFLOPS 64GB RAM, 6000GB HDD
	<b>Eシステム</b> (コプロセッサ搭載システム) 48ノード	Xeon E5-2670V2-2.5GHz (10コア)×2CPU Xeon Phi 5110P-1.053GHz (60コア)×4基 理論演算性能 4444GFLOPS 128GB RAM, 2000GB HDD
	<b>Fシステム</b> (ノード間高並列演算指向) 26ノード (うち2ノードGPU搭載) 10Gbps	Xeon E5-2698v4-2.2GHz (20コア)×2CPU NVIDIA® Tesla® P100 GPU 理論演算性能 1152GFLOPS+4.7TFLOPS (別機ノード) 128GB RAM, 2000GB HDD
用線	<b>Gシステム</b> (コプロセッサ搭載システム) 4ノード 10Gbps	Xeon E5-2640-2.5GHz (6コア)×2CPU Xeon Phi 5110P-1.053GHz (60コア)×1基 理論演算性能 1251GFLOPS 64GB RAM, 1000GB HDD
	<b>Hシステム</b> (ノード間高並列演算指向) 136ノード 10Gbps x4	Xeon D-1541-2.1GHz (8コア)×1CPU 理論演算性能 205GFLOPS 64GB RAM, 512GB SSD

システム内ネットワーク  
シャード間 40Gbps  
シャード内ノード間 10Gbps x 2 (34ノード/1シャード)

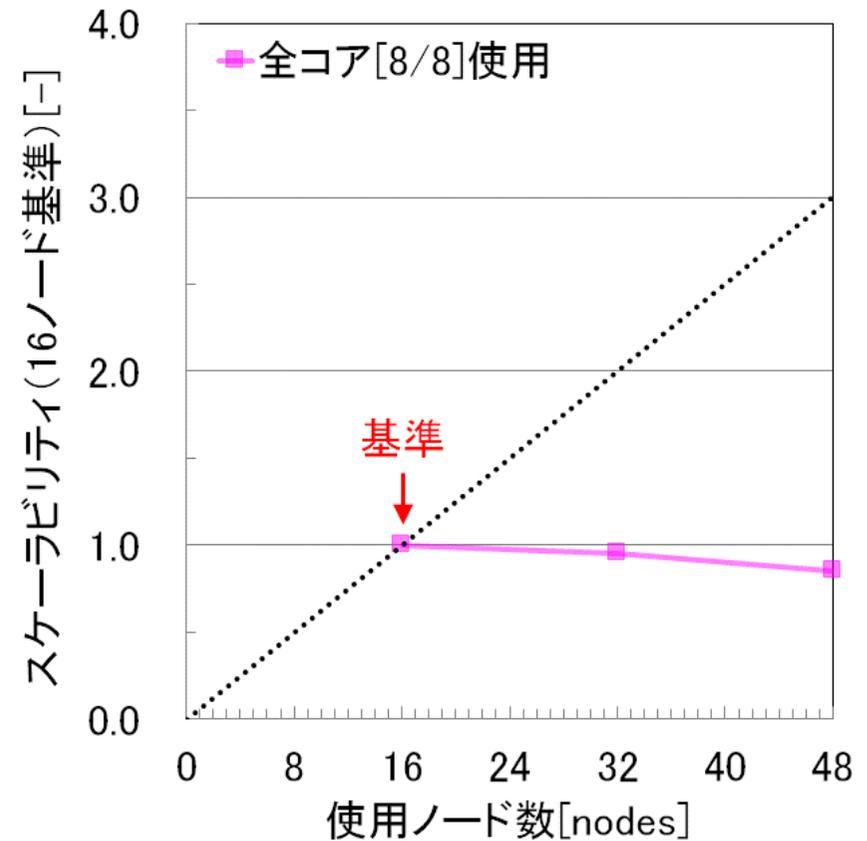
- ・Hシステムの① Coarse Meshでは、256並列(32ノード使用)を超えるとスケールラビリティが大幅に低下した。これは、Hシステムのノード間通信が10Gbpsであり、通信負荷が増大したためと考えられる。

## 9. ベンチマーク結果

( i ) システム比較 [Hシステム] ② Standard Mesh: 約1億メッシュ



使用ノード数と計算時間の関係

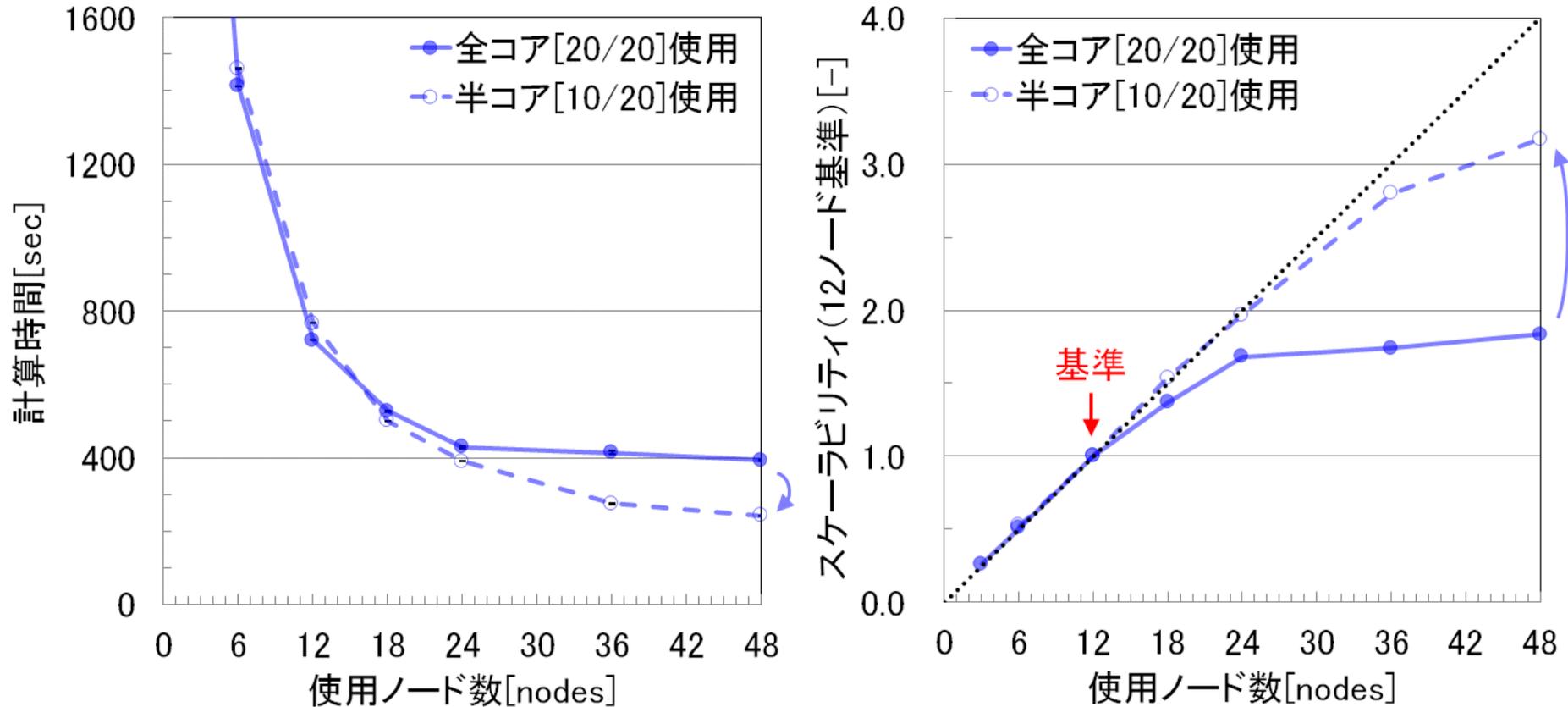


使用ノード数とスケーラビリティの関係

- ・Hシステムの② Standard Meshでは、並列数を増やすほど計算時間が長くなる傾向が見られた。計算時間のばらつきも大きい。
- ・前述の通り、通信負荷の増大が原因として考えられる。

## 9. ベンチマーク結果

(ii) 半分間引きの影響検証 [Dシステム] ① Coarse Mesh: 約2000万メッシュ



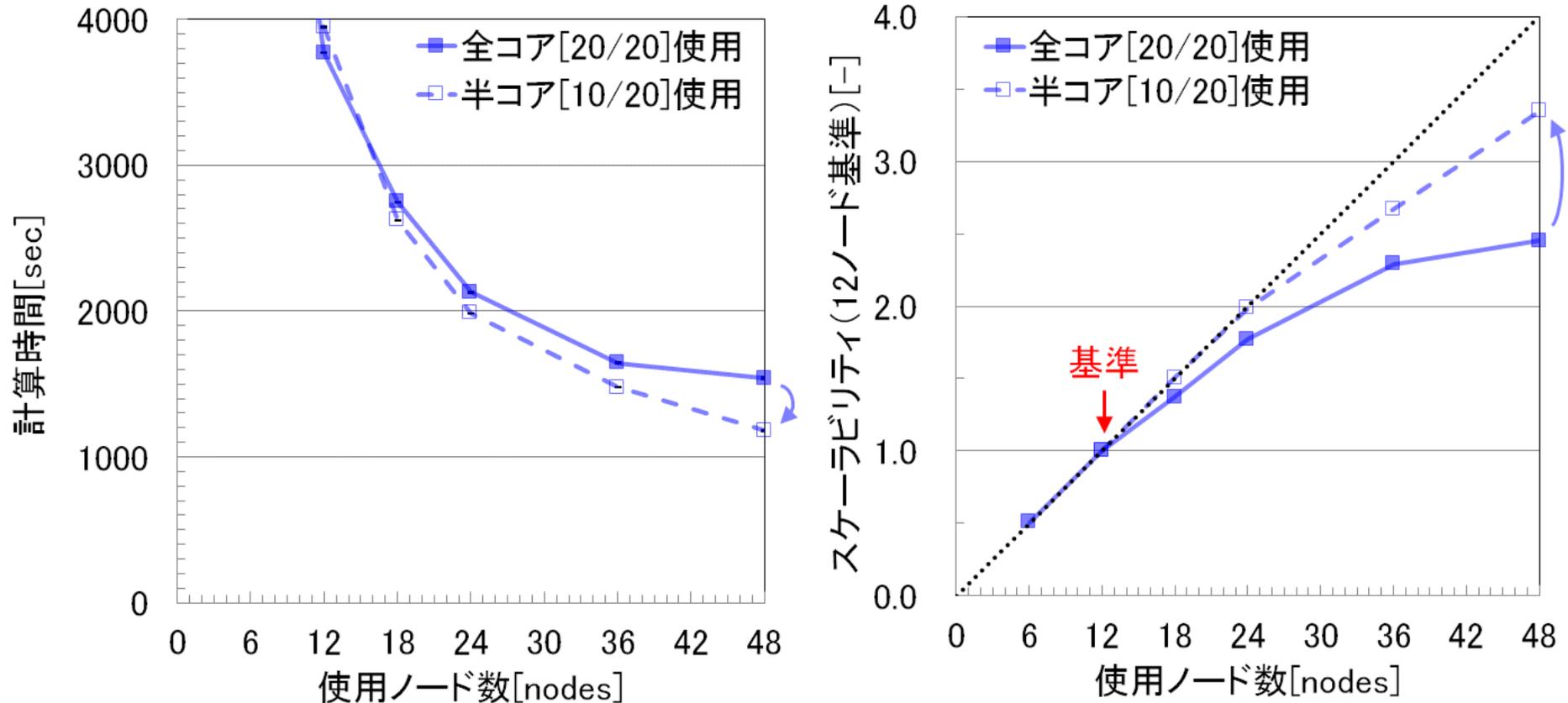
使用ノード数と計算時間の関係

使用ノード数とスケールラビリティの関係

- ・Dシステムの① Coarse Meshでは、18ノード以上の並列で、使用するコアを半分に間引いた方が計算時間が短くなった。
- ・最大の48ノード使用時には、使用するコア数を半分に間引くことで、大幅に並列効率が改善した(12ノード使用時を基準として、45→79%)。

## 9. ベンチマーク結果

(ii) 半分間引きの影響検証 [Dシステム] ② Standard Mesh: 約1億メッシュ



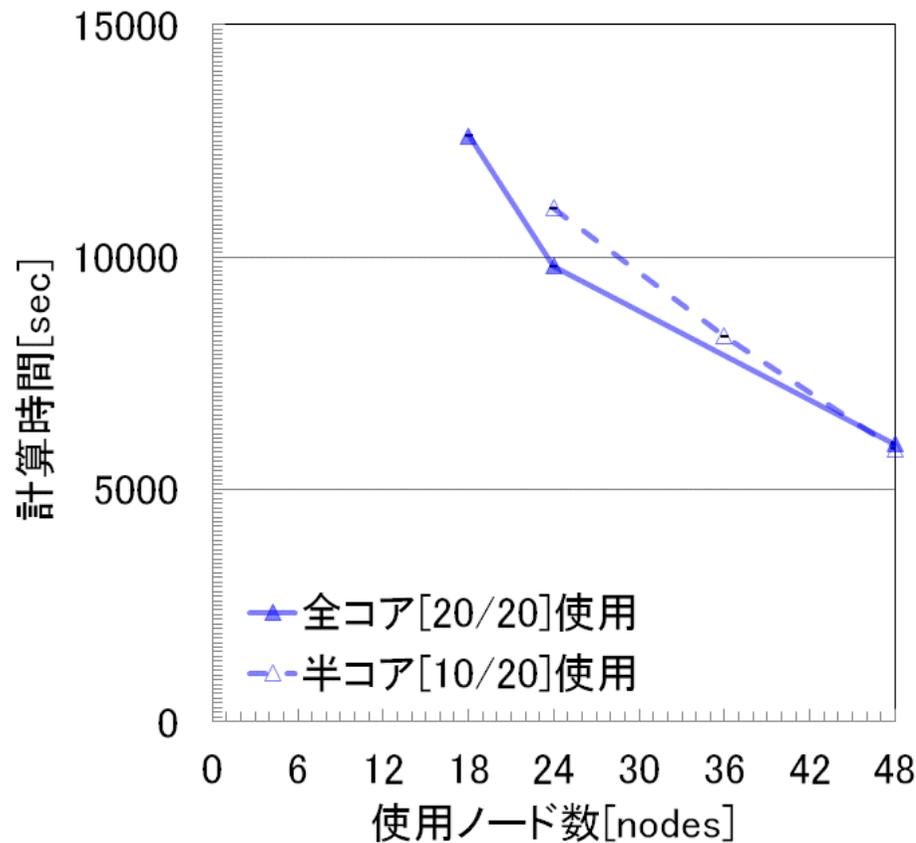
使用ノード数と計算時間の関係

使用ノード数とスケーラビリティの関係

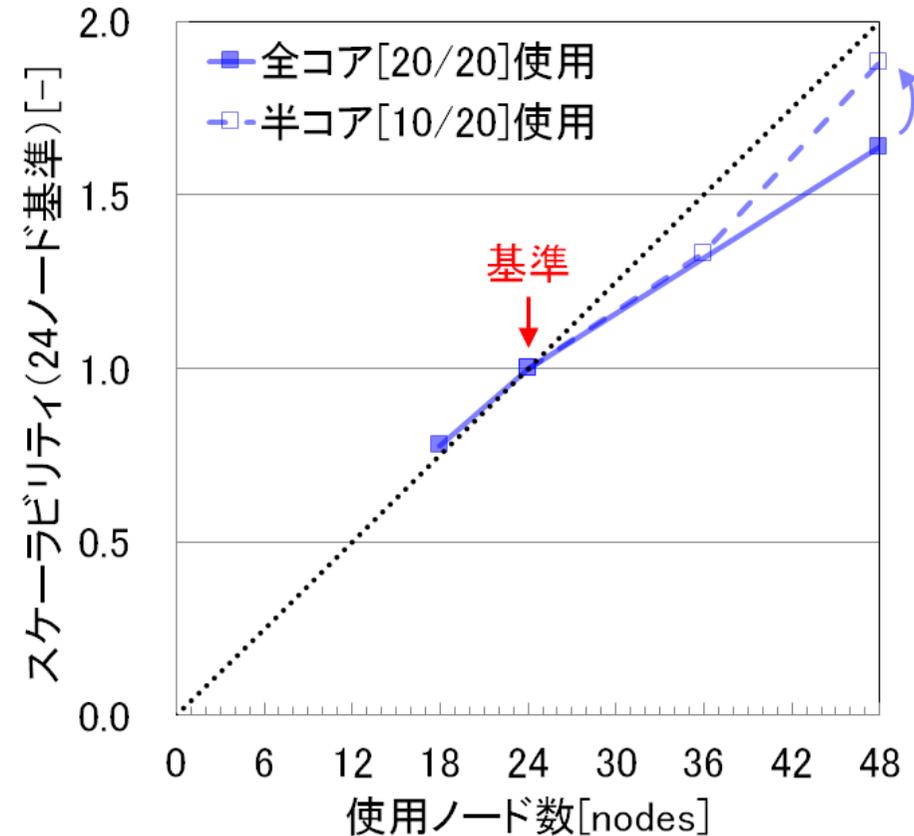
- ・Dシステムの② Standard Meshでも, 18ノード以上の並列で, 使用するコアを半分  
に間引いた方が計算時間が短くなった.
- ・最大の48ノード使用時には, 使用するコア数を半分に間引くことで, 大幅に並列  
効率が改善した(12ノード使用時を基準として, 63→84%).

## 9. ベンチマーク結果

(ii) 半分間引きの影響検証 [Dシステム] ③ Fine Mesh: 約5億メッシュ



使用ノード数と計算時間の関係



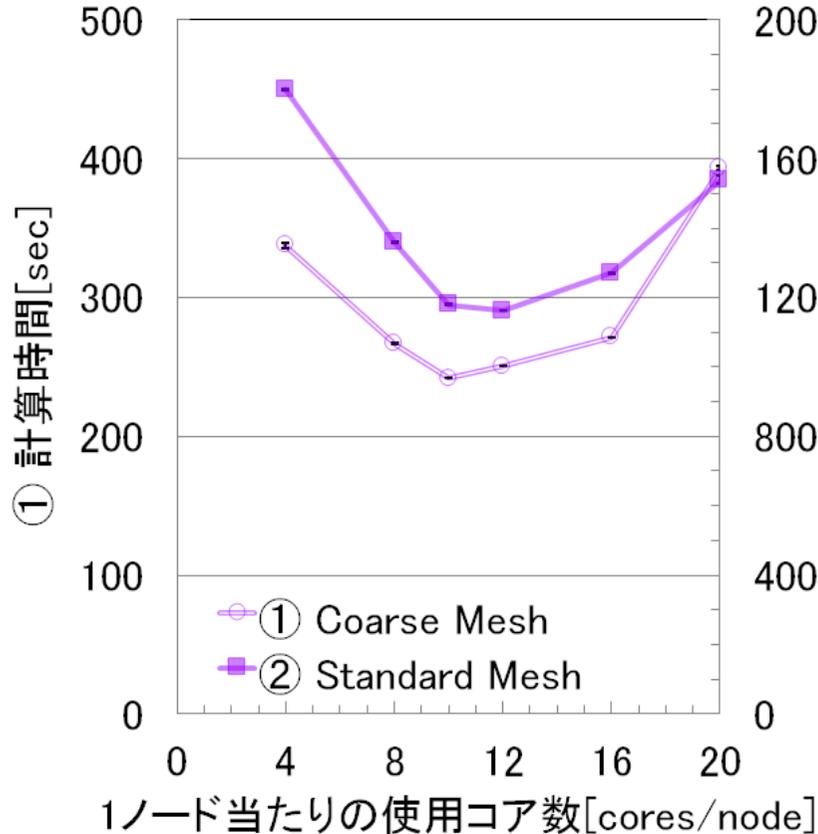
使用ノード数とスケーラビリティの関係

- ・Dシステムの③ Fine Meshでも、並列数の増加に伴い、計算時間が短くなった。ここでは、48ノードを使用した場合を除き、全部のコアを使用した方が計算時間が短くなった。ただし、測定点数が少なく明確な傾向は得られなかった。

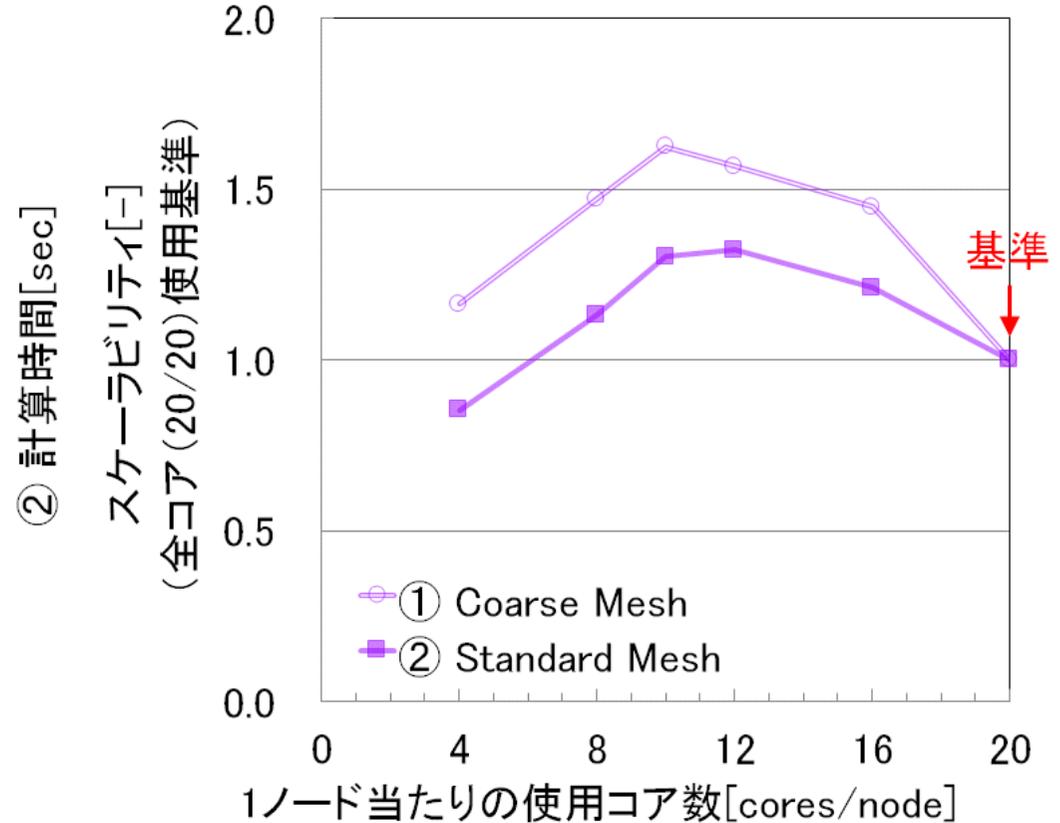
# 9. ベンチマーク結果

(iii) 間引きの最適化検証(48ノード固定)[Dシステム]

① Coarse Mesh: 約2000万メッシュ, ② Standard Mesh: 約1億メッシュ



使用コア数と計算時間の関係



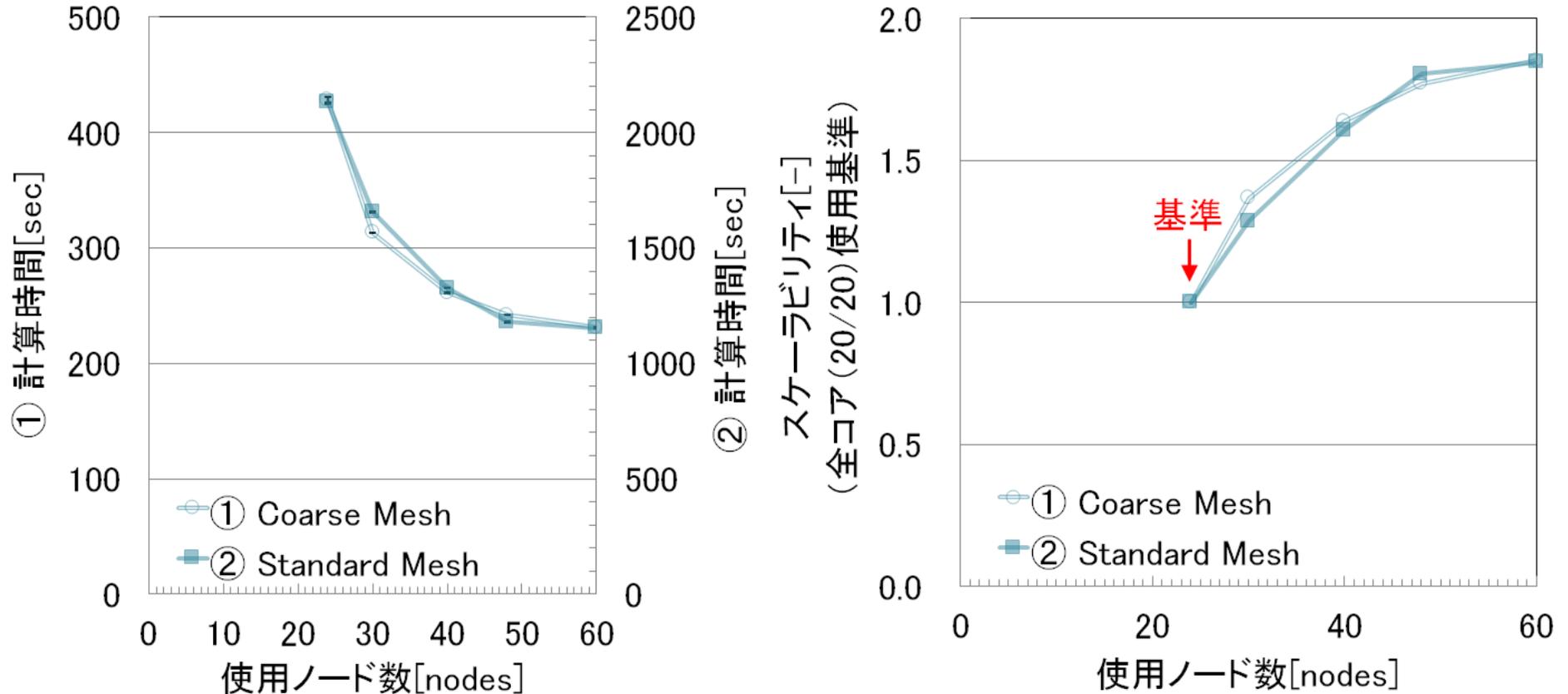
使用コア数とスケーラビリティの関係

- ・Dシステムで使用するノード数を48で固定し, 1ノード当たりの使用コア数を変化させたところ, ① Coarse meshでは, 10/20コア使用で, ② Standard Meshでは, 12/20コアの使用で計算時間が最も短くなった.

## 9. ベンチマーク結果

(iii) 間引きの最適化検証(480コア固定)[Dシステム]

① Coarse Mesh: 約2000万メッシュ, ② Standard Mesh: 約1億メッシュ



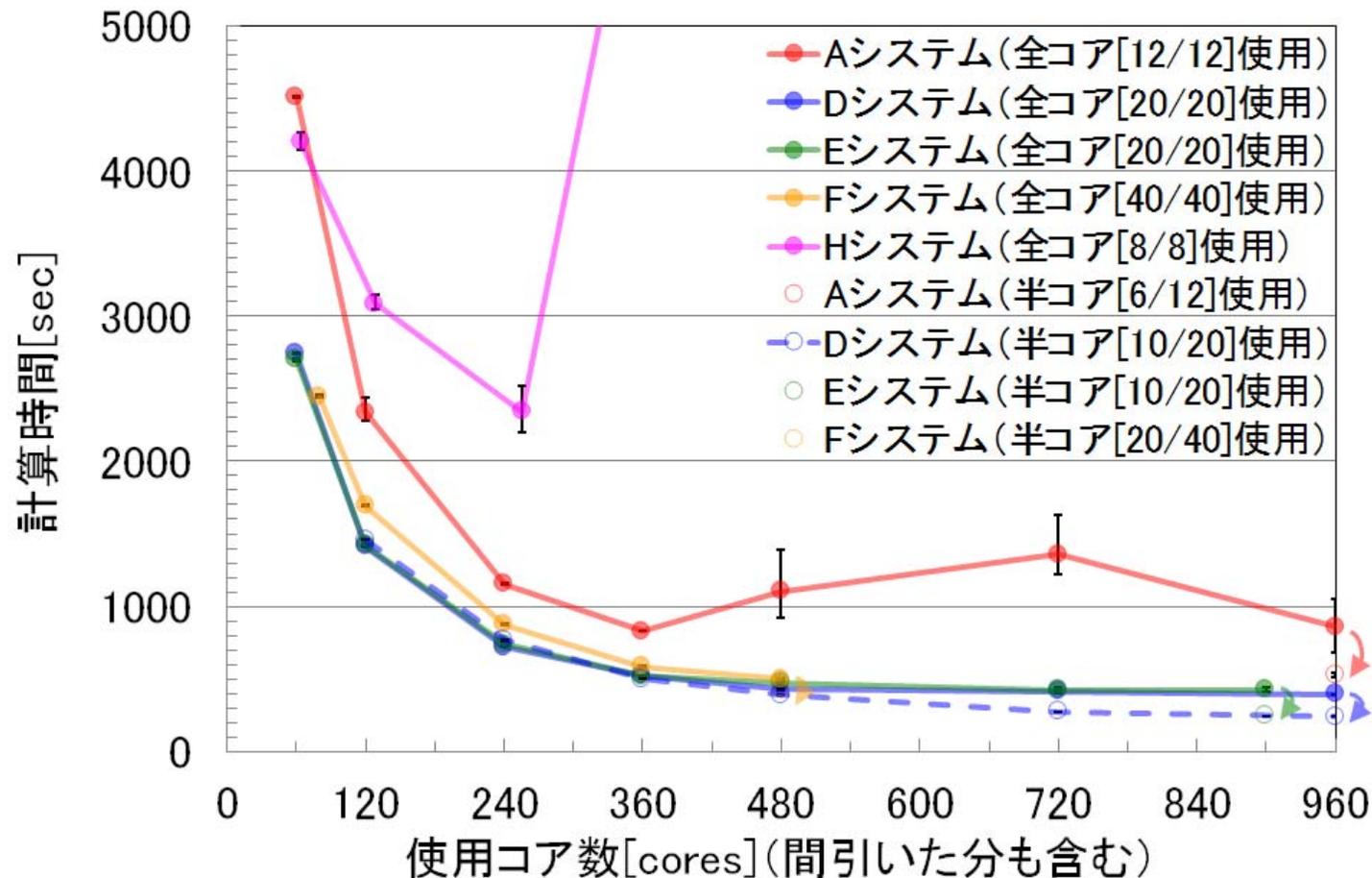
使用ノード数と計算時間の関係

使用ノード数とスケーラビリティの関係

- ・Dシステムで並列数を480で固定し, 使用するノード数を変化させたところ, いずれのメッシュでも, 使用ノード数が48以上で, 計算時間はほぼ一定となった. 全コア(24ノード)使用時を基準としたスケーラビリティは, 約1.8に達した.

## 9. ベンチマーク結果(まとめ)

① Coarse Mesh: 約2000万メッシュ(20,399,141メッシュ)

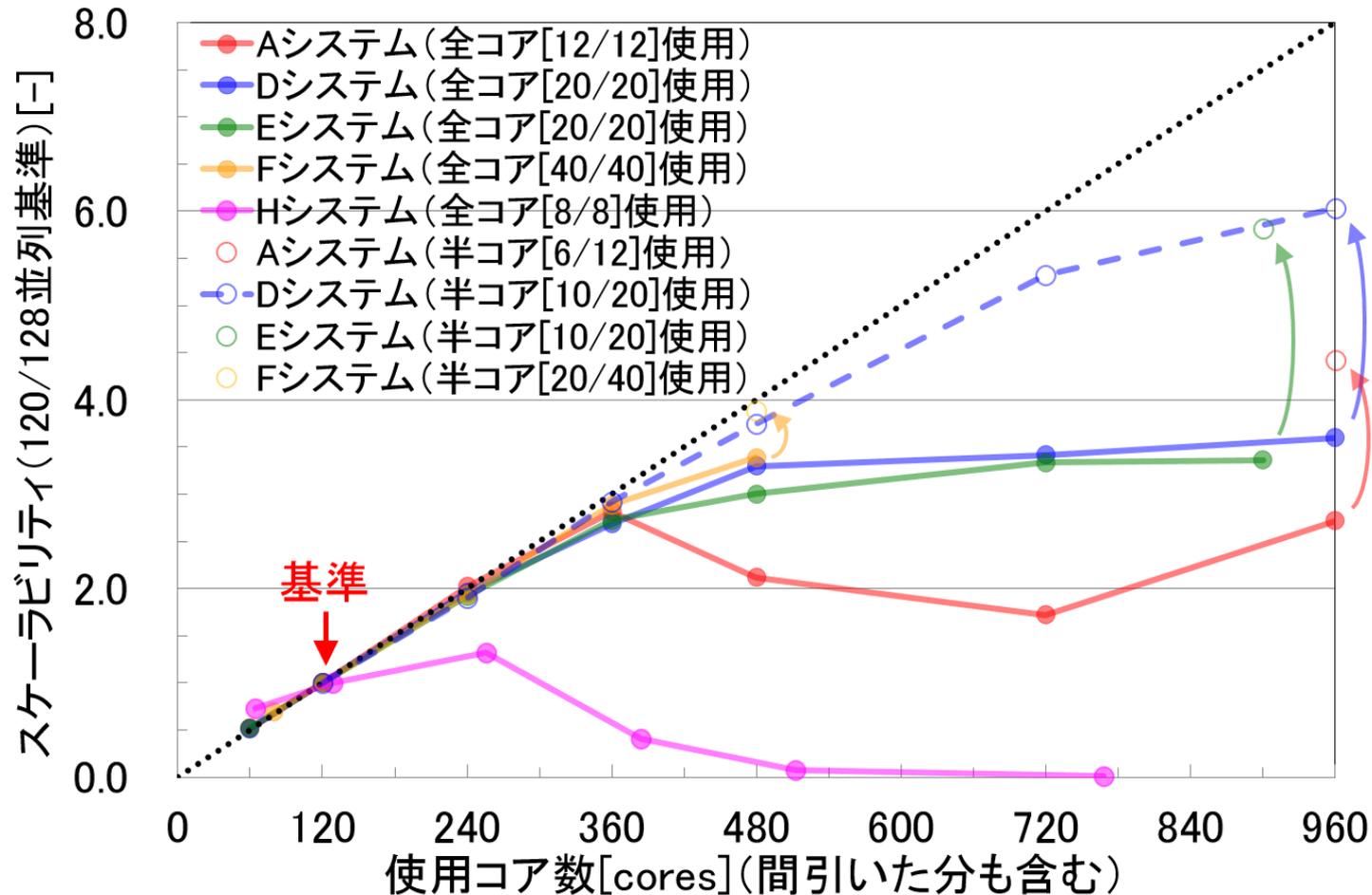


使用コア数と計算時間の関係

- ① Coarse Meshでは, Dシステム, Eシステムで計算時間が最も短く, 使用するコアを間引くことでさらに短くなる. Fシステムは, CPUの動作周波数の影響と考えられるが, Dシステム, Eシステムよりはやや計算時間が長い.

## 9. ベンチマーク結果(まとめ)

### ① Coarse Mesh: 約2000万メッシュ(20,399,141メッシュ)

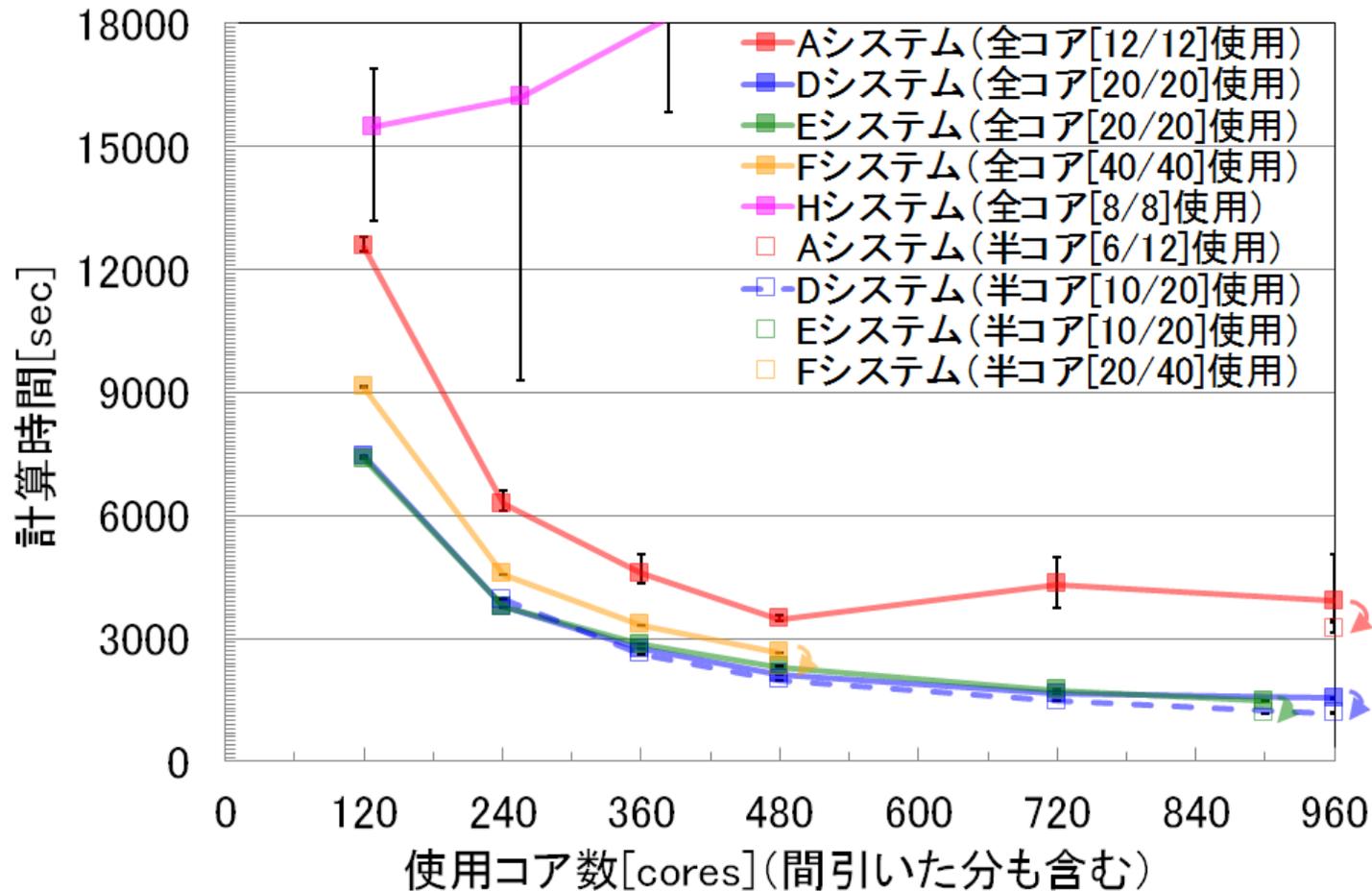


使用コア数とスケーラビリティの関係

- ① Coarse Meshでは, Dシステム, Eシステムで480並列(全コア使用)以上になると, スケーラビリティはほぼ一定に近づく. ただし, 使用するコア数を半分にすることで, 高並列時でもスケーラビリティは向上する.

## 9. ベンチマーク結果(まとめ)

### ② Standard Mesh: 約1億メッシュ(102,608,986メッシュ)

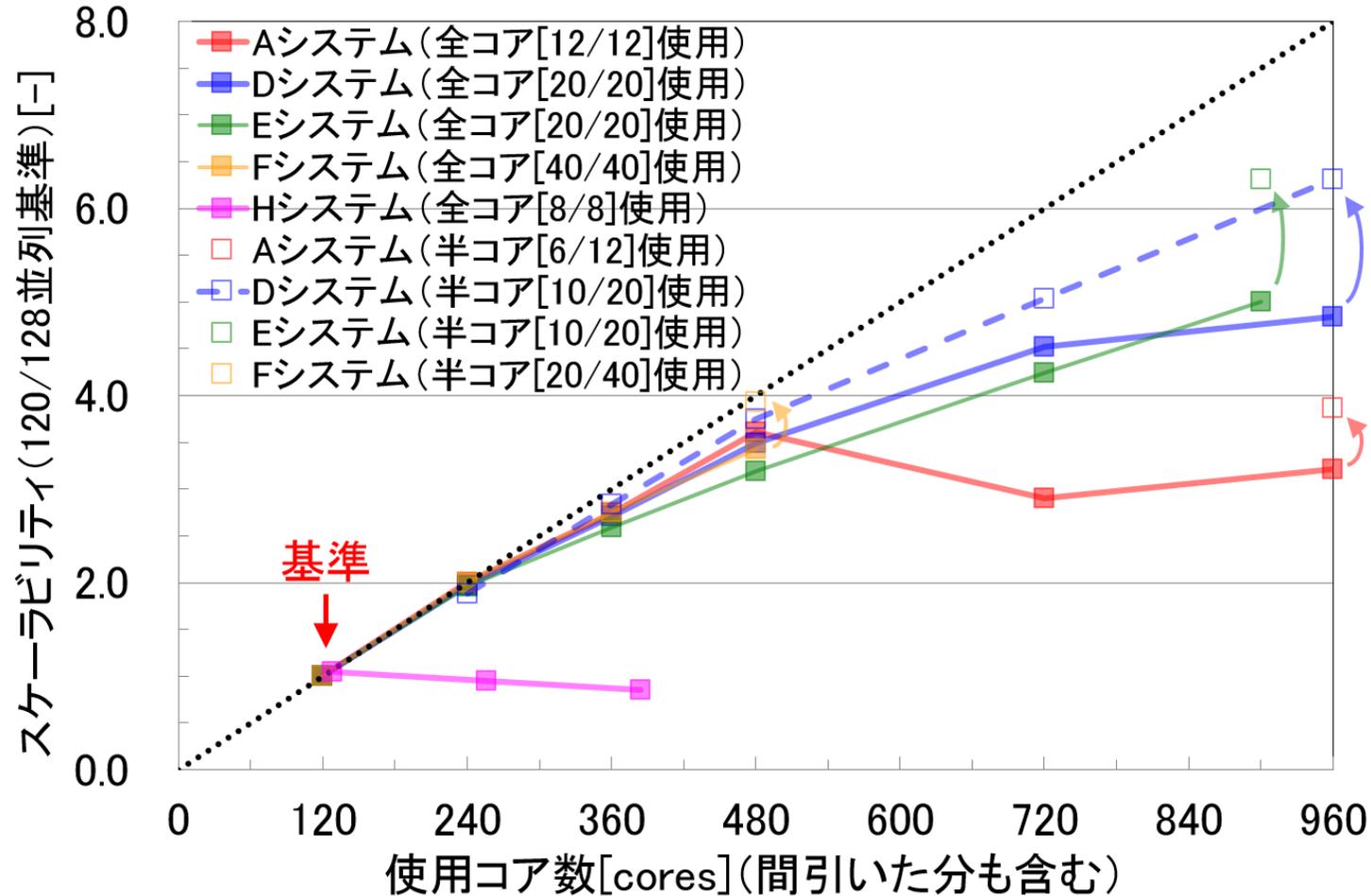


使用コア数と計算時間の関係

- ② Standard Meshでも, Dシステム, Eシステムで計算時間が最も短く, 使用するコアを間引くことでさらに短くなった. Fシステムは, CPUの動作周波数の影響と考えられるが, Dシステム, Eシステムより, 計算時間がやや長い.

## 9. ベンチマーク結果(まとめ)

### ② Standard Mesh: 約1億メッシュ(102,608,986メッシュ)



使用コア数とスケーラビリティの関係

- ② Standard Meshでは, Dシステム, Eシステム, Fシステム)とも使用するコア数の増加に伴い, スケーラビリティは向上した. また, 使用するコア数を半分にしても, スケーラビリティは向上した.

## 10. まとめ

FOCUSスパコンのDシステム, Eシステム, Fシステム, Hシステムを使用し, オープンソースベースの汎用CFDソフトウェアiconCFDの並列性能ベンチマークテストを実施した.

- FOCUSスパコンのEシステム, Fシステムでは, 並列数を増やすことで, 計算時間は短くなった. 高並列時に使用するコアを間引くことで, さらに並列効率は向上した. バラつきもほぼ見られなかった.
- FOCUSスパコンのHシステムでは, 約2000万メッシュの解析で, 256並列(32ノード使用)を超えると, 並列効率は著しく低下した. バラつきも大きい.
- FOCUSスパコンのDシステムでは, 18ノード以上の並列で, 使用するコアを半分に間引いた方が計算時間が短くなった.
- FOCUSスパコンのDシステムで使用するノード数を48(一定)としたところ, 1ノード当たりの使用コア数を半分[10/20コア], もしくは, 半分からやや多め[12/20コア]とすることで, 計算速度は最大となった.
- 前回のベンチマーク結果も含めると, FOCUSスパコンではDシステム, Eシステムの計算時間が最も短く, 次いでFシステムの計算時間が短い. Hシステムは, 大規模な計算には向かない場合がある. コストも考慮して考えると, バラつきはあるが, Aシステムの選択が良い場合もある.

# 謝辞

本ベンチマークは、公益財団法人 計算科学振興財団 (FOCUS) 様の協力を得て実施した。