

---

---

2018年12月14日

第十八回PCクラスタシンポジウム@秋葉原コンベンションホール  
PCクラスタコンソーシアム実用アプリケーション部会報告

**OpenFOAMによる流体解析ベンチマークテスト**  
**スパコン・クラウド・FOCUSでのチャンネル流れ解析**

今野 雅

PCクラスタコンソーシアム実用アプリケーション部会 委員

オープンCAE学会V&V小委員会 委員長

東京大学情報基盤センター 客員研究員

株式会社OCAEL

# オープンCAE学会共通OpenFOAMベンチマーク

Open CAE Laboratory

- 大学のスーパーコンピュータ

- ✓ 近年は産業利用など教育・公共機関以外でも利用可能なシステム有り
- ✓ 通常、課題審査が必要。通常1ヶ月～1年単位での課金

- 産業界専用のスーパーコンピュータFOCUS

- ✓ 法人の場合、課題の審査無く利用可能
- ✓ 1円単位で使った分だけ課金。ただし1年毎にアカウント発行料も必要

- クラウドサービス

- ✓ 通常、誰でも課題の審査無く利用可能で、使った分だけ課金(分または時間単位)

- CPU, メモリ, インターコネクต์に違いがあり, 性能比較が難しい.



- オープンCAE学会V&V小委員会で, チャンネル流れによる共通OpenFOAMベンチマークを作成し, 解析速度と並列化効率を比較した.

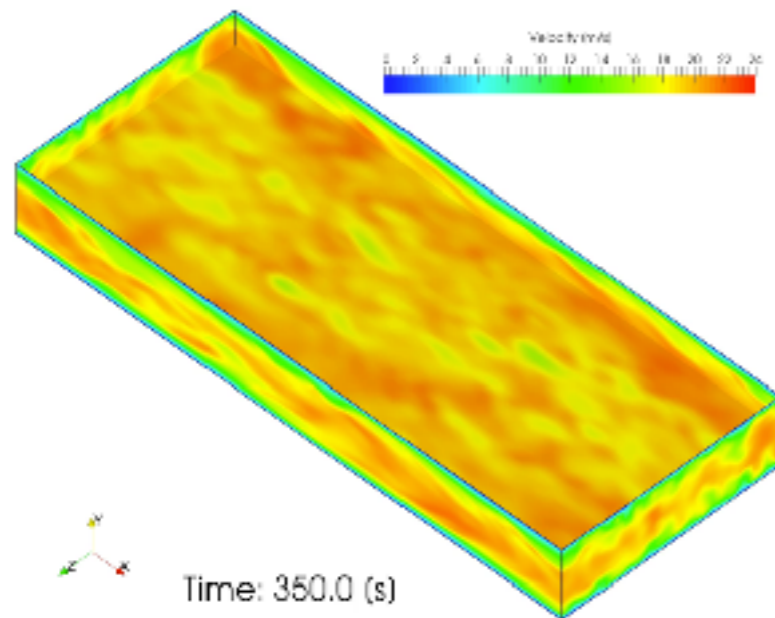
- 測定結果は学会のリポジトリで公開している.

- なお, 昨年までは費用比較も行なっていたが, 価格体系が複雑であったり, 定額制であるシステムも含まれる事から, 今回は費用比較を行わない.

# オープンCAE学会チャンネル流ベンチマーク

## チャンネル流れ ( $Re_\tau = 110$ )

## 解析条件



格子数約3M  
(0.37, 24M  
等他の格子  
数の結果は  
レポジットリ  
参照)

- ✓ メッシュ生成に時間を要しない
- ✓ 構造格子のため、格子数変更が容易
- ✓ 圧力と速度のみ解くので、「圧力線形ソルバの解析時間が支配的」という非圧縮性流体解析の特性を素直に示す

$L_x \times L_y \times L_z = 5\pi \times 2 \times 2\pi$   
 $Re_\tau = u_\tau \delta / \mu = 110 [-]$   
ここで  
 $L_x, L_y, L_z$ : 各方向のチャンネル幅 [m]  
 $u_\tau$ : 壁面摩擦速度 [m/s]  
 $\delta$ : チャンネル半幅 [m] ( $=L_y/2$ )  
 $\mu$ : 動粘性係数 [ $m^2/s^2$ ]  
主流方向(x): 一定の圧力勾配  
主流方向(x), スパン方向(z): 周期境界  
ソルバ: OpenFOAM pimpleFoam  
乱流モデル: 無し(laminar)  
速度線型ソルバ: BiCG (前処理DILU)  
圧力線型ソルバ: PCG (前処理DIC)  
領域分割手法: scotch(周期境界面は同領域)

• 2~51ステップのCPU時間(Execution time)から1時間あたりのステップ数を算出

# 計測システム(黄色が昨年発表からの追加分)

機関	システム (略称)	CPU [Intel Xeon] (周波数[GHz])	CPU数 (コア)	倍精度性能 [GFlops]	メモリ[GiB] (帯域幅[GB/s])	インターコネク (帯域幅[Gbps])
<a href="#">JCAHPC</a>	Oakforest- PACS(OFP)	Phi 7250 (1.4)	1(68)	3046	96(115.2), MCDRAM 16(490)	Intel Omni-Path (100)
<a href="#">東京大学</a>	Reedbush-U(RBU)	E5-2695 v4(2.1-3.3)	2(36)	1210	256(76.8×2)	Infiniband EDR(100)
<a href="#">FOCUS</a>	A(FA)	L5640(2.26)	2(12)	108	48(25.6×2)	Infiniband QDR(40)
	D(FD)	E5-2670 v2(2.5)	2(20)	400	64(51.2×2)	Infiniband FDR(56)
	F(FF)	E5-2698 v4(2.2)	2(40)	1152	128(76.8×2)	
	H(FH)	D-154(2.1)	1(8)	205	64 (34.1)	10GbE(10)×2 or 4
<a href="#">Amazon</a>	c4.8xlarge(c48x)	E5-2666 v3(2.9)	2(18)	310	60(不明)	10GbE(10)
<a href="#">Microsoft</a>	Azure A9(A9)	E5-2670(2.6)	2(16)	333	112(不明)	Infiniband QDR(40)
	Azure H16r(H16r)	E5-2667 v3(3.2)	2(16)	691	112(不明)	Infiniband FDR(56)
<a href="#">大阪大学</a>	OCTOPUS(OCT)	6126 (2.6)	2(24)	1997	192(255.9)	Infiniband EDR(100)
<a href="#">九州大学</a>	ITO A(ITO)	6154 (3.0-3.7)	2(36)	3456	192(255.9)	Infiniband EDR(100)
<a href="#">Oracle</a>	BM HPC 2.36(ORA)	6154 (3.0-3.7)	2(36)	3456	384(255.9)	RoCE v2 (100)

注) 倍精度性能は、機関のWEBページに明記された値、または、AVXの動作周波数を考慮して算出したWEBページから引用

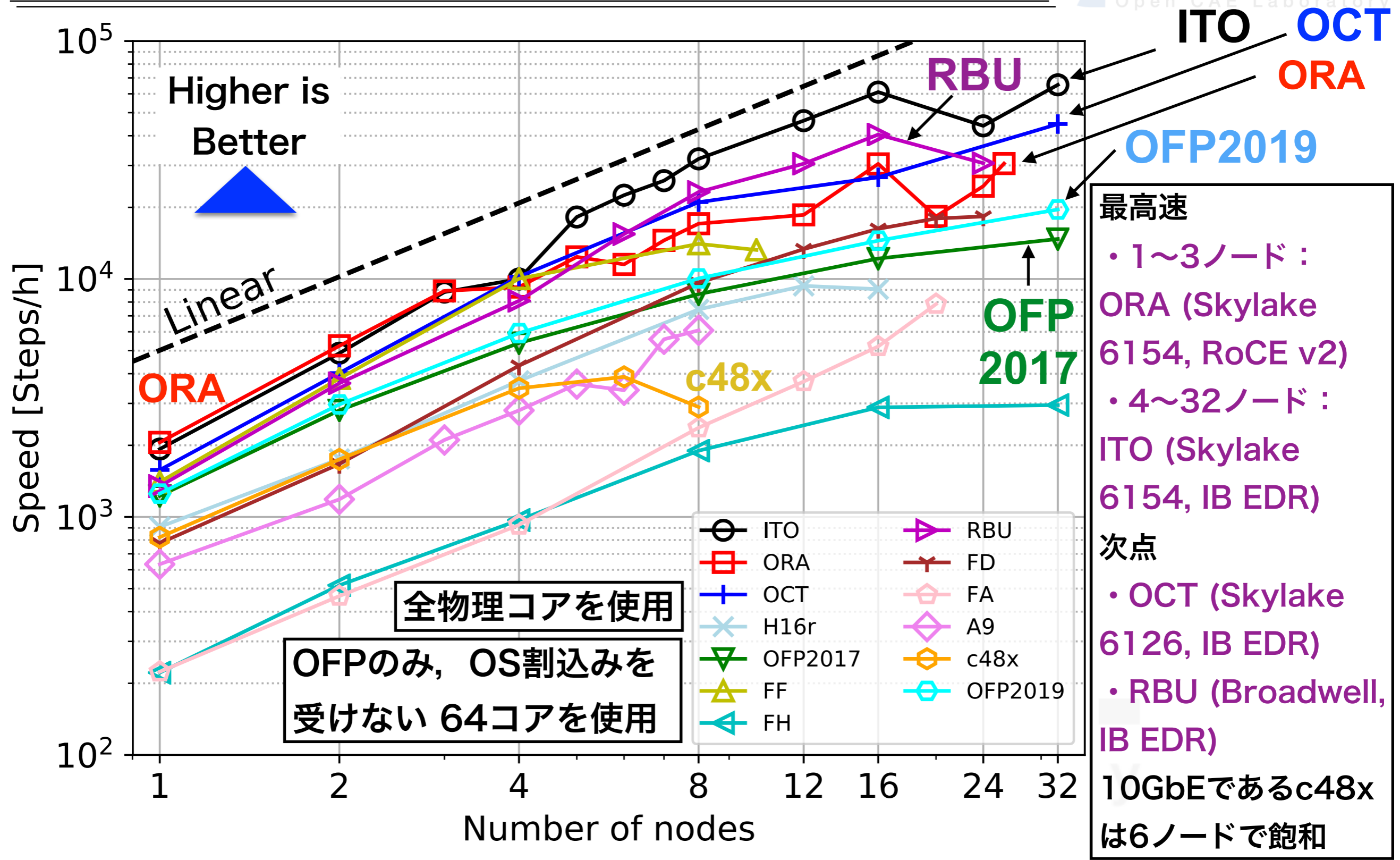
# 使用OpenFOAMバージョン・コンパイラ・MPI

Open CAE Laboratory

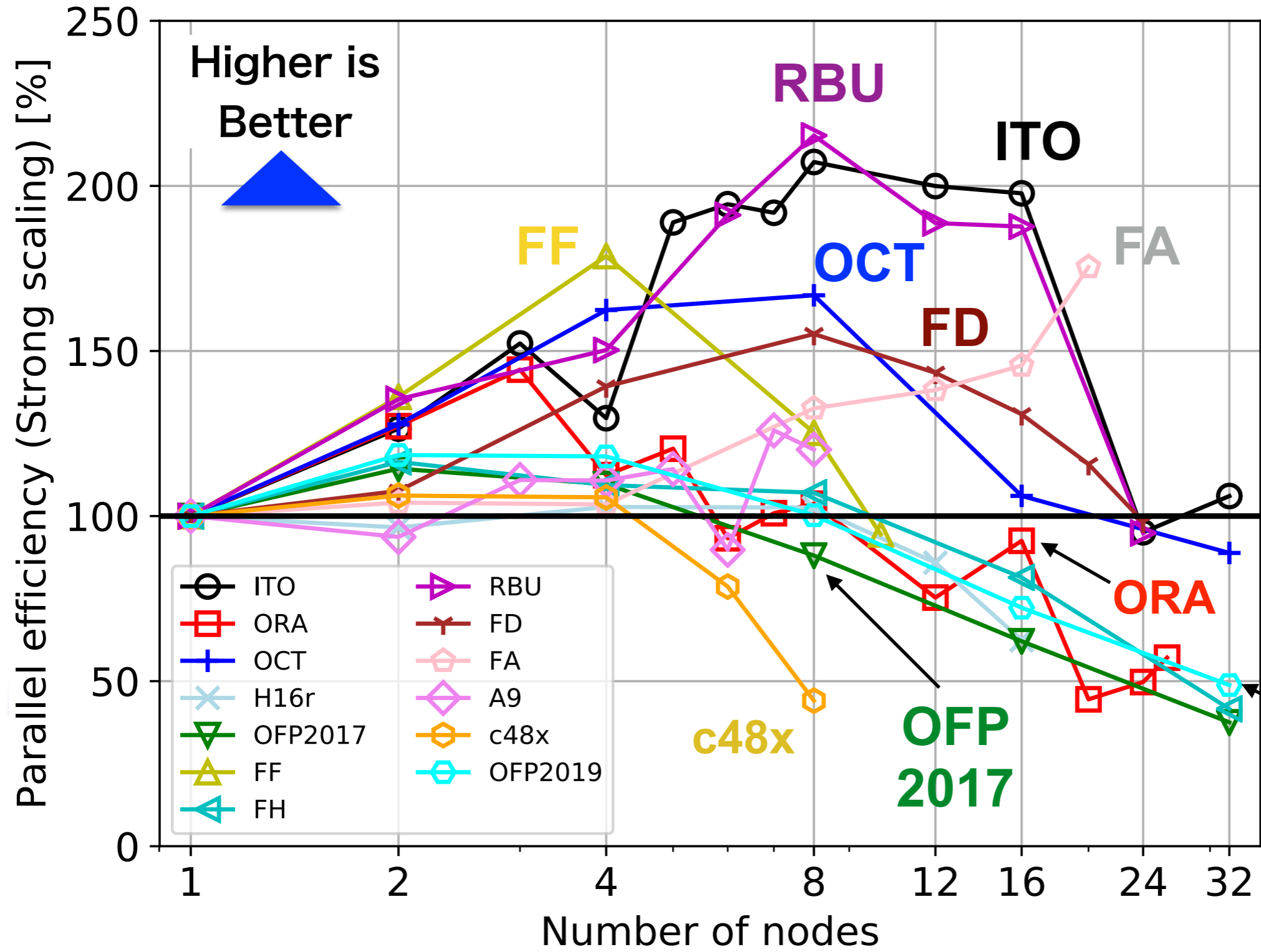
システム	Ver.	コンパイラ	MPI	実行時の主な設定(注)
OCT	OpenF OAM v1612+	icc 2018.2	Intel MPI 2018.2	I_MPI_DYNAMIC_CONNECTION=0, I_MPI_DAPL_TRANSLATION_CACHE=0, I_MPI_DAPL_UD_TRANSLATION_CACHE=0
OFP2019		icc 2019.1	Intel MPI 2019.1	unset KMP_AFFINITY ;KMP_HW_SUBSET=1T, I_MPI_PIN_PROCESSOR_EXCLUDE_LIST=0,1,68,69,136,137 ,204,205, I_MPI_PIN_DOMAIN =4, MPI_BUFFER_SIZE=8388608(8MiB), HBM_THRESHOLD=4, HBM_SIZE=256, LD_PRELOAD=libhbm.so. IccKNLでビルド
OFP2017 (Flat)		icc 2017.4	Intel MPI 2017.3	
RBU	OpenF OAM 2.3.0	Gcc-4.8.5	SGI MPI 2.14	
ITO			Intel MPI 2018.1	I_MPI_DAPL_TRANSLATION_CACHE=0, I_MPI_DAPL_UD_TRANSLATION_CACHE=0
ORA			Intel MPI 2018.1	I_MPI_DAPL_TRANSLATION_CACHE=0, I_MPI_DAPL_UD_TRANSLATION_CACHE=0
H16r			Intel MPI 5.0.3	I_MPI_FABRICS=shm:dapl, I_MPI_DAPL_PROVIDER=ofa- v2-ib0, I_MPI_DYNAMIC_CONNECTION=0, I_MPI_DAPL_DIRECT_COPY_THRESHOLD=655360
A9			Intel MPI 5.1.1	I_MPI_FABRICS=shm:dapl, I_MPI_DAPL_PROVIDER=ofa- v2-ib0, I_MPI_DYNAMIC_CONNECTION=0
c48x			OpenMPI 1.8.5	mpirun -bind-to core
FOCUS			OpenMPI 1.6.5	mpirun -bind-to-core -mca btl openib,sm,self

注) OFP2017では, MPI\_BUFFER\_SIZE=1000000, HBM\_THRESHOLD=16,HBM\_SIZE=100. その他は同様

# 各システムの解析速度比較



# 各システムのStrong scaling並列化効率比較



- InfinibandのRBU, ITO, OCT, FD, FAは16ノードまでスーパーニア
- RoCE v2のORAは16ノードまで概ねスケール
- OFP2019は12ノード以上で悪化
- 10GbEのc48xは6ノードから劣化

OFP 2019

# まとめ

- FOCUSや大学のスパコンおよびクラウドにおいて、オープンCAE学会V&V小委員会作成のチャンネル流れによるOpenFOAMベンチマークテストを実行し、各システムでの解析速度や並列化効率を比較した。
- 理論演算性能やメモリバンド幅が高い最新のシステムは高い性能が得られた。
- インターコネク트가Infinibandのシステムでは、16ノードまでスーパーリニアとなるが、10GbEのシステムでは6ノード以上でスケールしなかった。一方、RoCE v2のシステムは16ノードまで概ねスケールした。
- インターコネク트의トポロジによってはMPI通信のレイテンシが大きくばらつき、多ノード並列での性能が劣化する。今後はホスト間MPI通信の帯域幅・レイテンシの統計値等も併せて取得していきたい。

謝辞：電通国際情報サービス(計測当時)の住友様からAmazon EC2, 日本Microsoft(計測当時)の佐々木様からMicrosoft Azure A9, 日本Microsoftの五十木様からMicrosoft Azure H16rでのベンチマークの結果をご提供頂いた。ここに深く感謝する。

**本ベンチマーク測定結果のコミットをお待ちしております!**



# 附録：OpenFOAM自動ビルドスクリプト

- 実用アプリケーション部会でOpenFOAM講習会を実施したシステムでは、原則として本OpenFOAMベンチマークを実施し、公開している(例：大阪大学 OCTOPUS).
- OpenFOAMをビルドするには、各種依存ソフトウェアが必要であり、依存ソフトウェアがシステムに無い時、ソースからビルドする必要が生じる。
- 種々の依存ソフトウェアのバージョン制限を考え、適切なバージョンのソースをダウンロードし、適切な順序でビルドしていくのは面倒。
- ビルド作業を自動化する `installOpenFOAM` を作成。
  - bashスクリプトなので、大抵のLinuxシステムでそのまま実行可能。
  - 依存ソフトウェアのバージョンを指定可能。
  - 必要なソースを自動でダウンロード。
  - ビルドした依存ソフトウェアの再利用。
  - 複数のコンパイラバージョン、MPIバージョンの組み合わせを自動的にビルド。
- FOCUS, Reedbush-U, Oakforest-PACS, ITO, OCTOPUS等のシステムについて、環境設定済み、並列ビルド用のジョブスクリプトも有り。

**オープンCAE学会のレポジトリで公開中!**