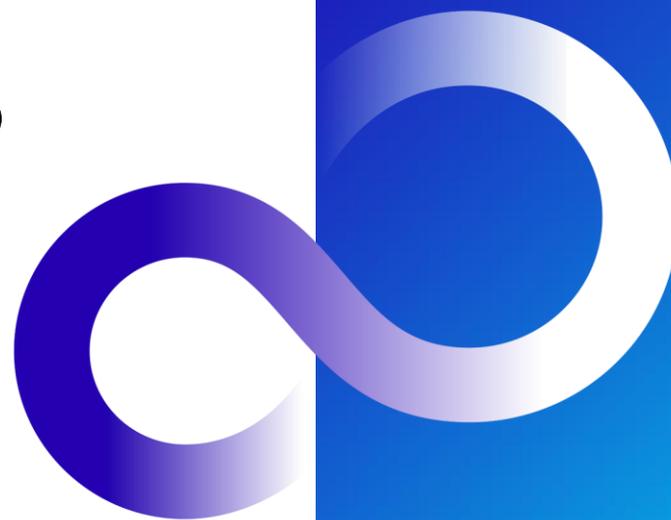


# 大規模CFD解析を加速する in-situ可視化

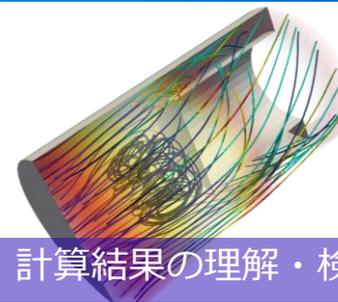
- 大日向大地(富士通株式会社)  
藤野敦志(宇宙航空研究開発機構)  
藤田直行(宇宙航空研究開発機構)



- **大規模CFD解析と可視化**
- in-situ可視化による効率化
- JAXAスーパーコンピュータの可視化環境
- まとめ

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0$$

```
1 #include <stdlib.h>
2 #include <mpi.h>
3 int main(int argc, char** argv)
4 {
5     MPI_Initialize(&argc, &argv);
6     for (int n = 0; n++ < argc) {
```

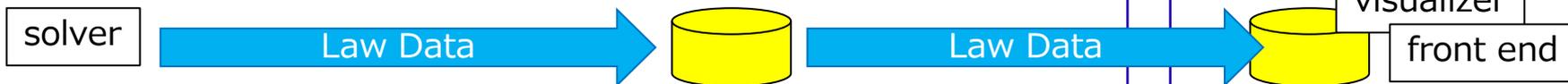


計算結果の理解・検証

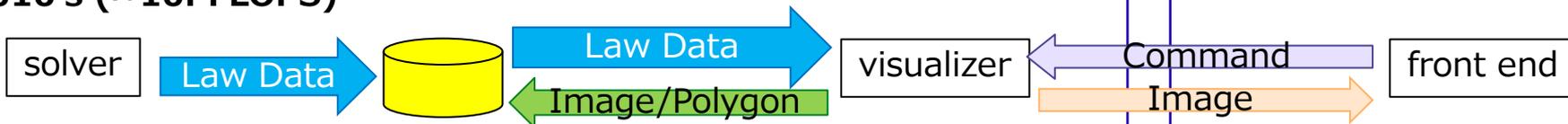
## Remote HPC System

## Local System

~2000's (~0.1PFLOPS)



2010's (~10PFLOPS)



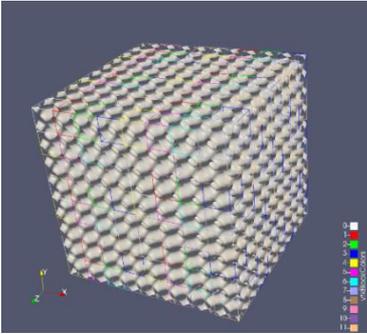
2020's ~ (10PFLOPS+)



- シミュレーションは解析を通じて人間が知見を得て意味がある
- 可視化はそのための「思考と理解の道具」
- 「待つ」「余計な操作」など思考を遮る要素は回避すべき



# Benchmark: Work load of Visualization



Software: ParaView v5.5.2

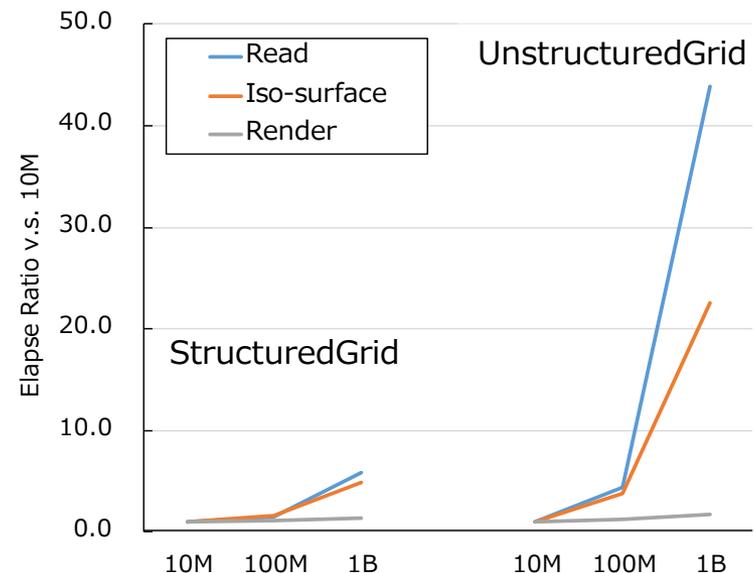
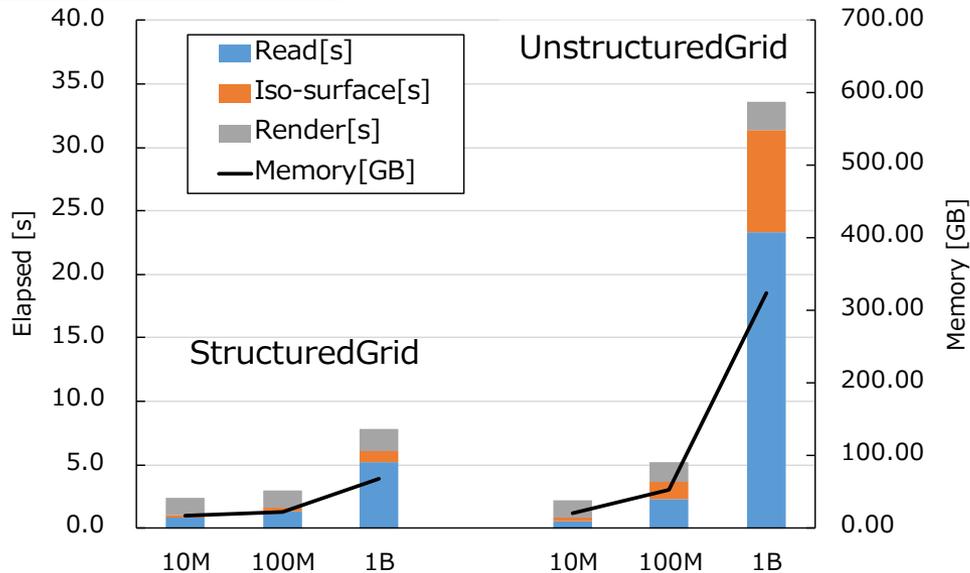
Computer: Intel Xeon E5-2643v2 x2, 64GB DDR3 1866, NVIDIA Quadro K2000

Parallelism: 8 nodes, 96 procs. (12procs/node)

Format: Multi-Block VTK 512 blocks, 8 values, Zlib compression

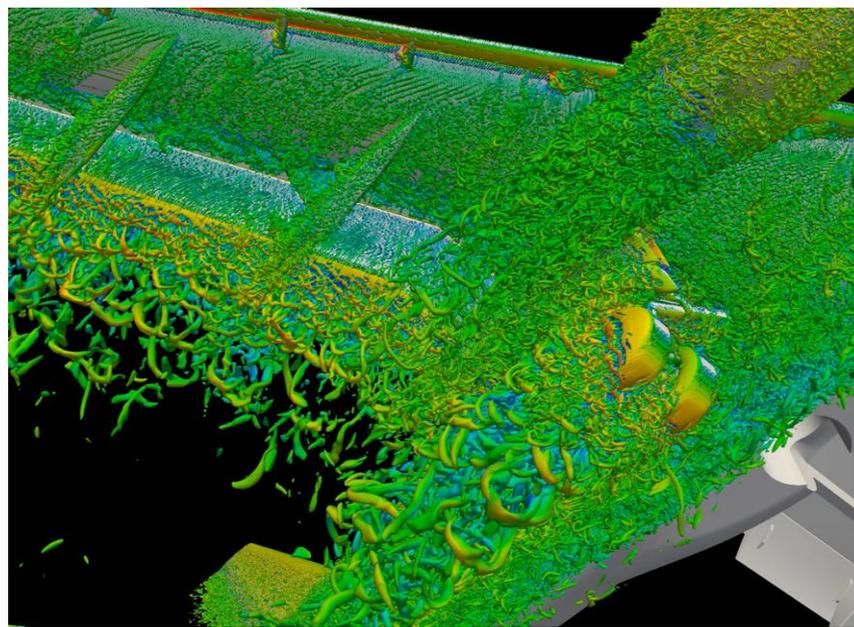
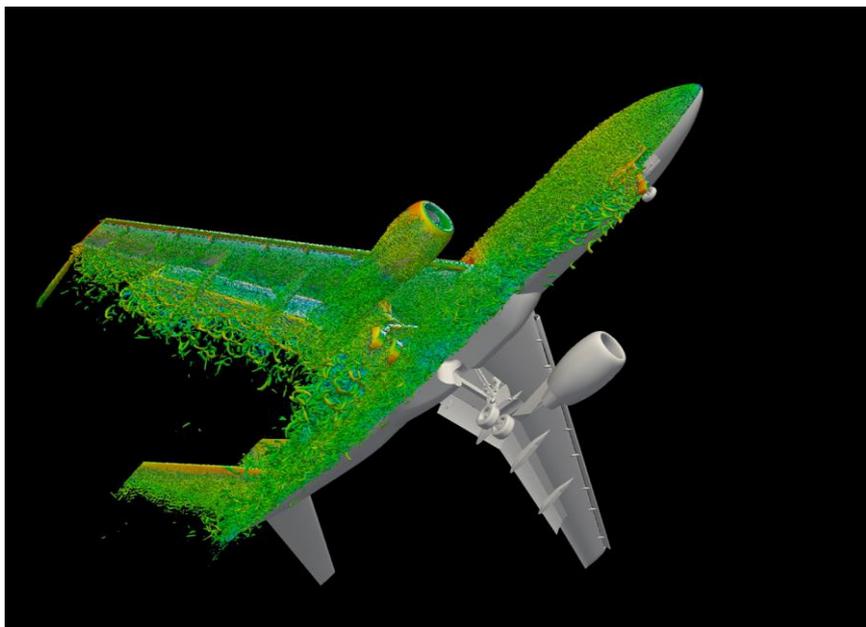
Field:  $P(x,y,z) = \sin(x) * \sin(y) * \sin(z)$

Visualization: Iso-surface of  $P=0.0$



格子サイズの増加に伴う“Read”(ストレージ律速)の増加は、“Iso-surface”, “Render”(CPU, GPU律速)の増加よりも高ペース。  
しかし技術トレンドでは、ストレージの性能向上は演算器の性能向上よりも遥かにゆっくり。

**既存の可視化システムではファイルIOの高負荷に耐えられない!!**



(データ提供: JAXA高木先生)

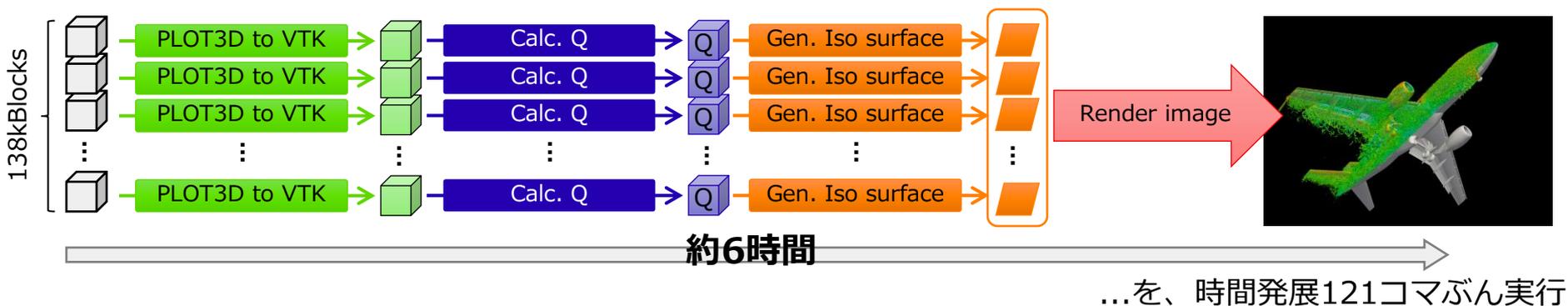
- FFVHC-ACE[1]を用いた実機詳細形状(JSM\_CRM\_LEGモデル)の解析。機体周囲の渦を可視化する。
- 直交格子13.8万ブロック, 45億点(100GB+), 時間発展121steps
- データサイズが大きく、可視化ソフトウェアでstraight forwardには可視化できなかった。

[1] 高木, 他. “階層型等間隔直交格子を用いた高速・高精度乱流解析プログラムの開発”, 第51回流体力学講演会/第37回航空宇宙数値シミュレーション技術シンポジウム講演論文集, pp.165-171. (2019.7)

## Preparation

- ブロック単位で、PLOT3DからVTK変換し、流れ場のQ値を算出し、等値面を生成する処理をプログラム実装する。
- 等値面データを統合して画像生成するプログラムを実装する。  
カラーコンター・画角などは同寸法の粗いデータで合わせておく。
- 再処理に備えて、各ステップの出力はブロック単位のファイルに書き出す。

## Execution



- 可視化条件変更時は、上流工程に戻って再処理する。
- スパコンシステムの混雑状況にもかなり影響を受ける。

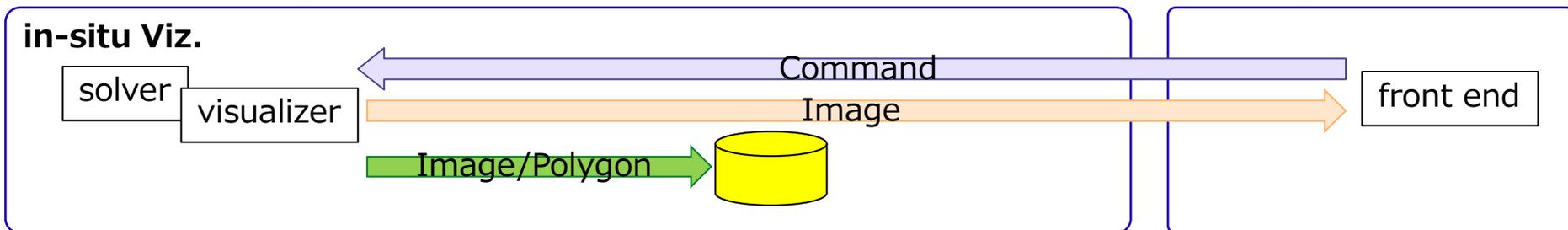
**“インタラクティブ”には程遠い状態**

- 大規模CFD解析と可視化
- **in-situ可視化による効率化**
- JAXAスーパーコンピュータの可視化環境
- まとめ

- “in-situ” = 「その場」 「あるべき場所」
  - その場：この場合はシミュレーション計算を実行している計算機
- シミュレーションをしながらそのまま可視化を行う
  - シミュレーション結果は数値データではなく可視化結果
  - 可視化結果：画像、動画、点群データ、ポリゴンデータ etc

## Remote HPC System

## Local System



- 演算性能とIO性能の乖離が進むエクサスケール時代において有力な可視化手段 [2]
- 数値計算と可視化の結合度で様々な方式

[2] S. Ahern, et al. “Scientific Discovery at the Exascale”, Report from the DOE ASCR 2011 Workshop on Exascale Data Management, Analysis, and Visualization (2011.2)

## Advantage

- 結果ファイルサイズが小さくなる → ファイルIO性能律速から解放
- シミュレーションステアリングの実現 [3]



## Disadvantage

- ソルバ側で作り込みが必要、ベースの作りによっては実装が困難
- ソルバと可視化で計算資源が競合する  
→ 対話的可視化は困難、バッチ可視化であれば十分実用的 [4][5]



- 計算資源の競合に対するアプローチ
  - 全てをin-situ処理で行わずにpost-hoc処理を併用 (→後述事例)
  - in-transit可視化  
ソルバと可視化で計算資源を物理的に分離し、通信でデータをステージングしながら(in-transit)可視化する [6]

[3] H. Yi, et al. "In-Situ of Turbulent Flows in Complex Geometries", 2014 Visualization and Computational Steering for Large-Scale Simulation IEEE International Conference on Big Data (2014.10)

[4] 堤, 他. "HPC環境におけるin-situ 可視化の利用と課題", 第31回数値流体力学シンポジウム (2017.12)

[5] 大西, 他. "大規模自動車空力解析におけるIn-situ 可視化", 計算工学講演会論文集, Vol.24 (2019.6)

[6] 堤, 他. "In Situ/In Transit アプローチを用いた大規模数値解析におけるポスト処理効率化", 第33回数値流体力学シンポジウム (2019.11)

- 航空エンジン燃焼器内で燃料が微粒化する過程の解析 [8]
- 微粒化した燃料を解像するために極めて高い空間分解能が必要  
→ 計算の大規模化が必然
- 解析規模：39億点と312億点 (比較のため2種類)
- 気液界面の情報を可視化したい、界面は等値面処理で抽出できる  
→ in-situ可視化と相性が良い
- 解析コード(JAXA南部さん開発)にin-situ可視化コードを実装して効果を検証した

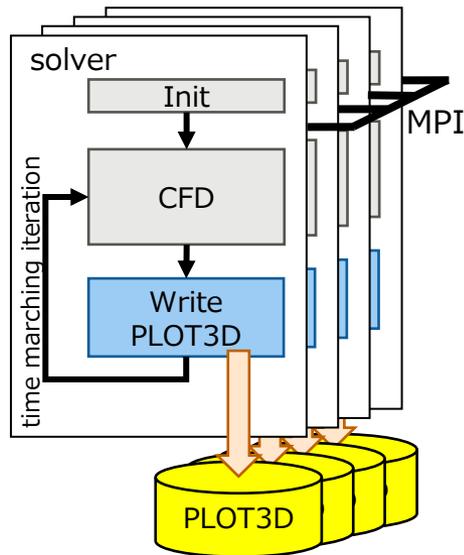


[7] 大日向, 他. “二相流解析コードへのin-situ可視化の適用と評価”, 第33回数値流体力学シンポジウム (2019.11)

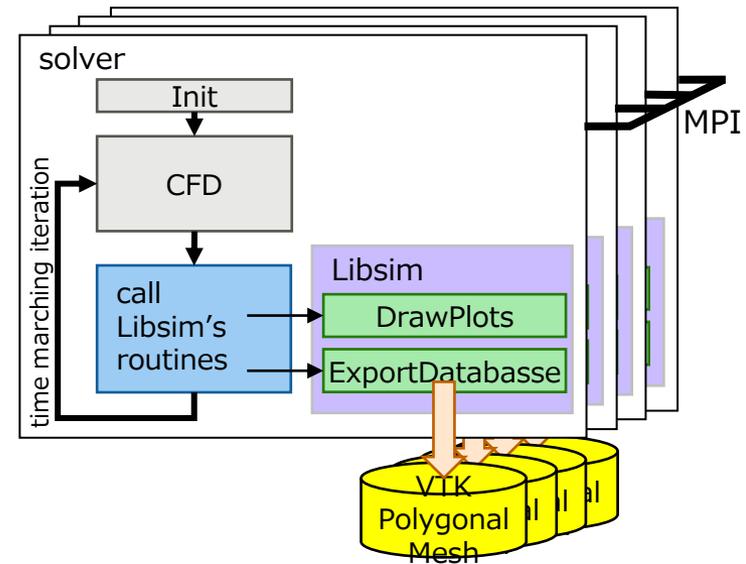
[8] 南部, 他. “ガスタービンエンジン燃焼器条件下におけるクロスフロー型燃料微粒化機構の詳細数値解析”, 第57回燃焼シンポジウム (2019.11)

- in-situ可視化ライブラリに VisIt/Libsim [9] を使用する
- ボリュームデータ(PLOT3D)を書き出す処理を、in-situ可視化処理に置き換える
- in-situ処理では、気液界面となる等値面の抽出処理とその結果データの出力を行う

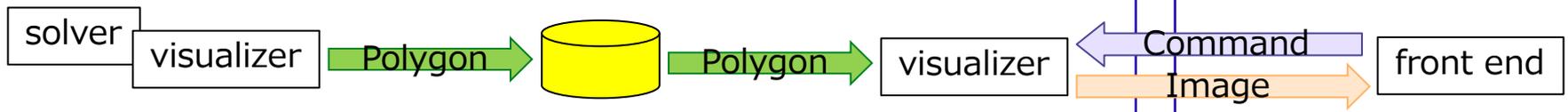
## Before



## After

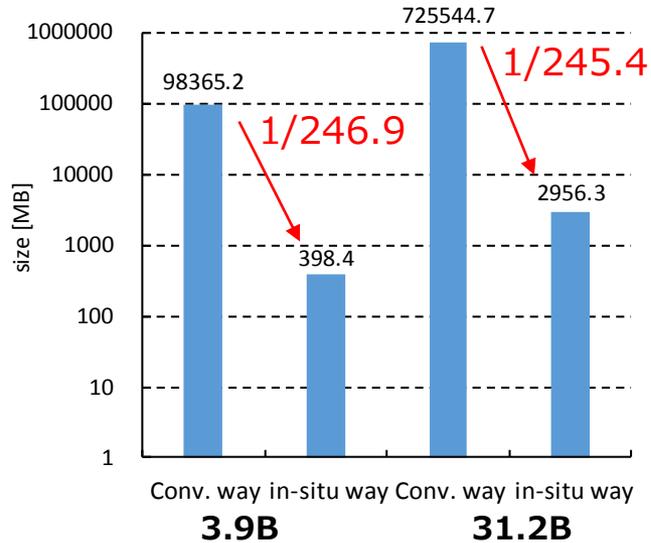


- 出力データはボリュームから面(VTK Polygonal Mesh)になる
- 面データの可視化はPost-hocで行い、視点変更などを対話的に行う

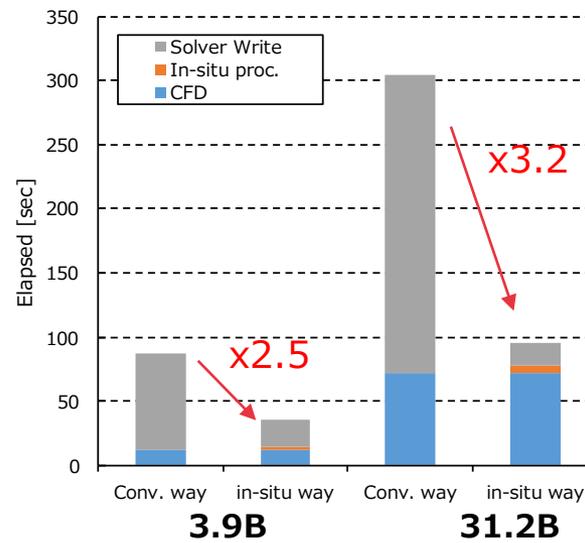


[9] "VisIt Users.org" <https://www.visitusers.org/>

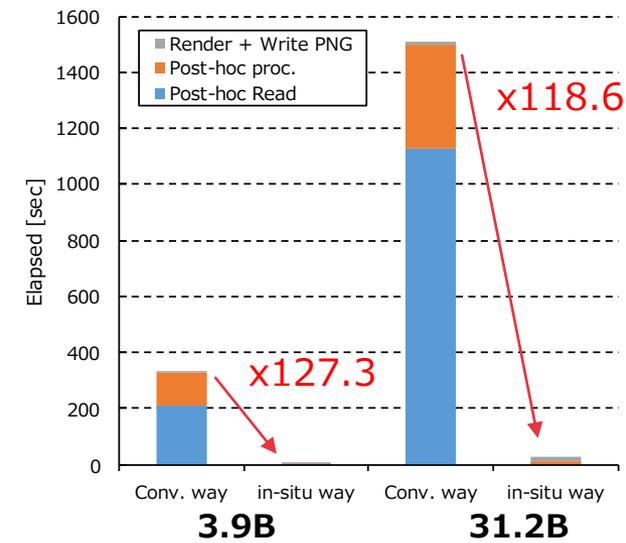
### File Size



### solver



### post-hoc viz.



- 出力データが約1/250にサイズダウン、post-hoc処理で計算資源をコンパクトに使用できるようになった
- データサイズ縮小によるIOコスト縮小、CPU負担が大きい等値面処理をソルバ側で高並列に実行  
→ データ書き出し時間の短縮効果がin-situ処理の時間を大きく上回る
- 反復的に行う post-hoc可視化のフェーズで100倍以上の速度改善、ストレス無い応答時間

**可視化のインタラクティブ性を確保、解析ワークフロー全体を加速**

- 大規模CFD解析と可視化
- in-situ可視化による効率化
- **JAXAスーパーコンピュータの可視化環境**
- まとめ

## JAXAスーパーコンピュータシステム JSS3

2020年12月1日

【コンピュータ基盤】 TOKI: TOKyo and ibaraKI



### 調布航空宇宙センター



#### TOKI-SORA: HPCシステム

PRIMEHPC FX1000  
 ノード数: 5,760 ノード (15ラック)  
 総理論演算性能: 19.4 PFLOPS  
 総主記憶容量: 180 TiB (32 GiB/ノード)



#### TOKI-RURI: 汎用システム

総理論演算性能: 1.24 PFLOPS  
 総主記憶容量: 104 TiB

**ST**: PRIMERGY RX2540 M5 x 375 ノード  
 (192 GiB/ノード, Quadro x1基)

**GP**: PRIMERGY CX2570 M5 x 32 ノード  
 (384 GiB/ノード, Tesla V100 x 4基)

**XM**: PRIMERGY RX2540 M5 x 2 ノード  
 (DCPMM 6.0 TiB/ノード, Quadro x1基)

**LM**: PRIMERGY RX2540 M5 x 7 ノード  
 (DCPMM 1.5 TiB/ノード, Quadro x1基)

可視化の利用はTOKI-RURIを使用することを基本とし、そのための環境を用意

- ・ リモートデスクトップ
  - ・ 多数の可視化ソフトウェア
  - ・ GPUによるグラフィックアクセラレーション
- など。

in-situ可視化は、TOKI-SORAの利用がメイン



### 筑波宇宙センター

#### TOKI-TFS: 筑波ファイルシステム

ファイルシステム: FEFS, 総実効容量: 0.4PB

#### TOKI-TLI: 筑波ログインシステム

PRIMERGY RX2540 M5 x 2 ノード  
 (384 GiB/ノード, Quadro x1基)

筑波運用管理制御システム

#### TOKI-TRURI: 筑波汎用システム

総理論演算性能: 145 TFLOPS  
 総主記憶容量: 10.8 TiB

**TST**: PRIMERGY RX2540 M5 x 46 ノード  
 (192 GiB/ノード, Quadro x1基)

**TGP**: PRIMERGY CX2570 M5 x 2 ノード  
 (384 GiB/ノード, Tesla V100 x 4基)

**TLM**: PRIMERGY RX2540 M5 x 1 ノード  
 (DCPMM 1.5 TiB/ノード, Quadro x1基)

### 【アーカイバ基盤】 J-SPACE



\* 調布航空宇宙センター内に設置

© スーパーコンピュータ活用課

TOKI-TRURIにも、TOKI-RURIと同じ環境を用意

from [https://www.jss.jaxa.jp/jss3\\_configuration/](https://www.jss.jaxa.jp/jss3_configuration/)

## 可視化アプリケーション

	TOKI-RURI,TRURI	TOKI-SORA	備考
FIELDVIEW	✓		
HPC FIELDVIEW	✓		
EnSight	✓		
Tecplot360	✓		
AVS/Express	✓		
ParaView	✓	✓	TOKI-SORAはバッチのみ
VisIt	✓	✓	TOKI-SORAはバッチのみ

## in-situ可視化関連

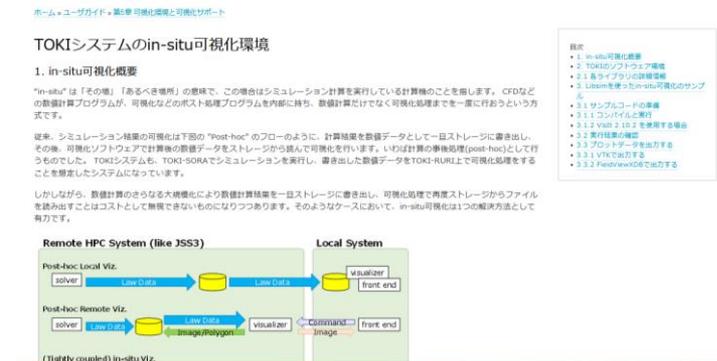
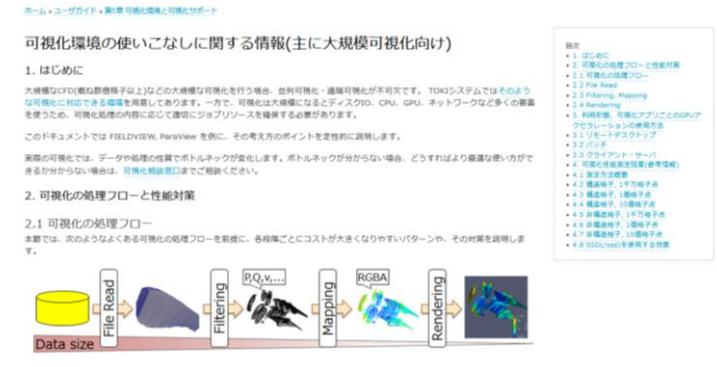
	TOKI-RURI,TRURI	TOKI-SORA	備考
Catalyst (ParaView)	✓	✓	
Libsim (VisIt)	✓	✓	
Libsim (HPC FIELDVIEW)	✓		VisItの可視化エンジンをFieldViewに移植
ADIOS2	✓	✓	in-transit可視化のデータステージング用途

## ●ポータルサイトで可視化利用についてガイド

- 環境・ツールの使い方
- システムの特徴に応じた使いこなし
- 可視化コードのサンプル実装
- 利用状況から随時アップデート

## ●可視化専門チームによる支援体制

- in-situ可視化や新しい可視化技術の相談にも対応



- 大規模CFD解析と可視化
- in-situ可視化による効率化
- JAXAスーパーコンピュータの可視化環境
- **まとめ**

- CFD解析の大規模化とその可視化について事例を交えて紹介した。
- 解析の大規模化の主要なボトルネックの1つはファイルIOであり、その技術的対策として「in-situ可視化」が有力である。
- 燃料微粒化解析にin-situ可視化を導入し、データサイズを著しく減少させ、可視化や解析ワークフロー全体の加速を実現した。
- JAXAスーパーコンピュータ「JSS3/TOKI」は、in-situ可視化のための環境・利用支援体制を備えている。

本講演では、in-situ可視化を大規模可視化の観点から用いているが、次のようなケースでも有用と考えられる。

- パラメータスタディのような小規模大量の解析の可視化
- 可視化以外のPost処理での利用  
例えば、機械学習などのデータサイエンス的な処理など
- 実世界との連携(デジタルツイン)におけるインタフェース

**Thank you**

