

# 弱連成のための汎用カップリングライ ブラリh3-Open-UTIL/MPの紹介

PCCC-OSSSワークショップ  
2023/03/30 東大柏Ⅱキャンパス 情報基盤センター

荒川隆

東京大学情報基盤センター/CliMTech

# 講演の内容

---

- 連成計算とは
- h3-Open-UTIL/MPの紹介
  - 概要
  - 連成方法
  - 特徴
- 性能評価
  - テストモデルを用いた評価
- 異機種間連成
  - シミュレーションモデル + AI

# 連成計算とは？

そして「現実」とは、大小無数の、振幅も展開速度もちがう「現象系」が、相互に影響しあいながら、もやもやしたパターンを描いている巨大複雑な「複合系」であり、まだ今のところ、コンピューターにすべての「現実」がはいっているわけでもない。

小松左京「日本沈没」より

- 連成計算
  - 振幅も展開速度もちがう複数の「現象系」を「複合」させて「現実」を再現する試み
- 強連成と弱連成
  - 強連成：異なる現象系をひとつの方程式系にまとめて計算する
  - 弱連成：個々の現象を個別に計算する
- 弱連成
  - 振幅も展開速度もちがう「現象系」どうしを橋渡しする必要

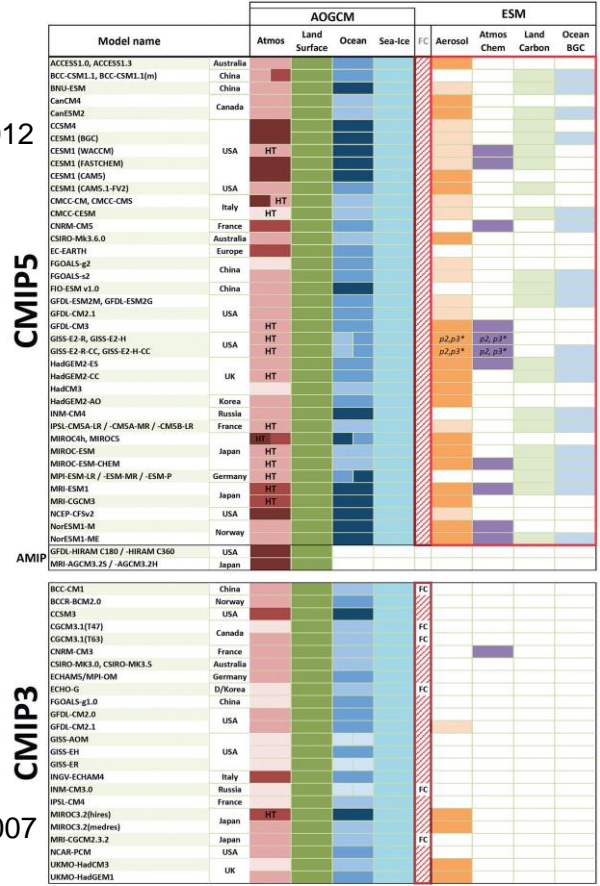
空間構造    時間構造

そのためのソフトウェア = カプラ

# 連成計算の事例

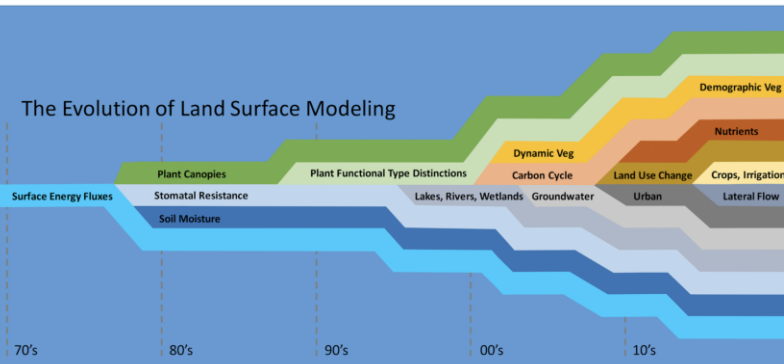
- 真鍋らによる大気海洋結合
  - 1980年代から本格的な結合計算
- 現代ではより多くのモデルコンポーネントが組み合わさるようになっている
  - 陸面や海氷

IPCC5次 2012

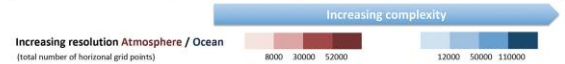


Portrait of Prof. Sukuro Manabe from Wikipedia

IPCC4次 2007



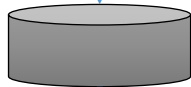
Fisher and Koven (2020)



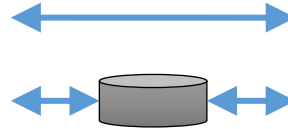
Increasing of resolution and complexity from IPCC AR5 WG1 report

# より多様な連成計算に向けて

- 現代のHPCは heterogeneous なアーキテクチャが主流
  - Simulation → Simulation + Data + Learning
  - File Sharing System → One way and Sequential



- 異機種間連成を可能にするソフトウェアがあれば
  - 双方向通信、並列実行が可能になる
  - S + D + Lの新しい可能性を拓く!



# h3-Open-UTIL/MPの概要

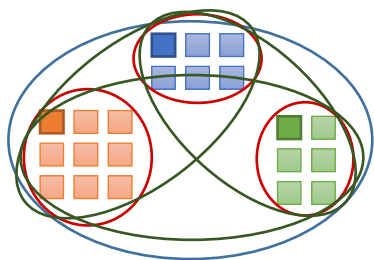
- 連成計算のための**汎用**ライブラリ
  - 弱連成計算を対象
  - 異なる時空間スケールを持つ複数のシミュレーションモデル間で情報（データ）を交換しながら計算を実行
- 特徴
  - 格子形状に依存しない→任意の格子のモデルに適用可能
  - 空間補間方法を利用者が規定できる
  - アンサンブル連成可能
  - h3-Open-SYS/WaitIOと連携し異機種間で連成計算できる
  - Pythonインターフェースを持つ

基本機能の特徴

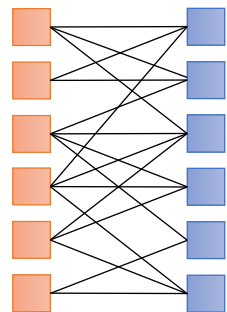
拡張機能の特徴

# h3-Open-UTIL/MPの一般論

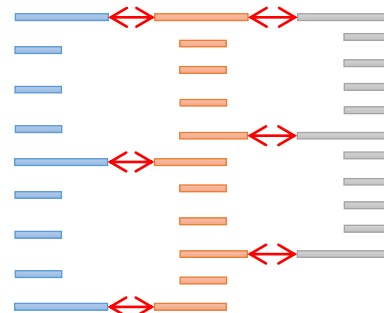
- カプラの一般的な機能
  - 複数のモデルコンポーネントのプロセス管理
  - データ交換
  - 格子変換



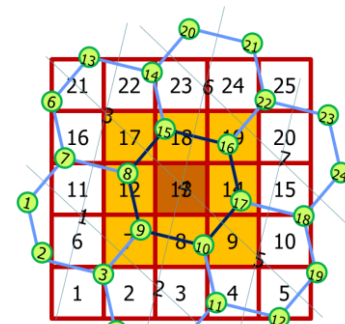
MPI communicators generated by the coupler



Data exchange pattern



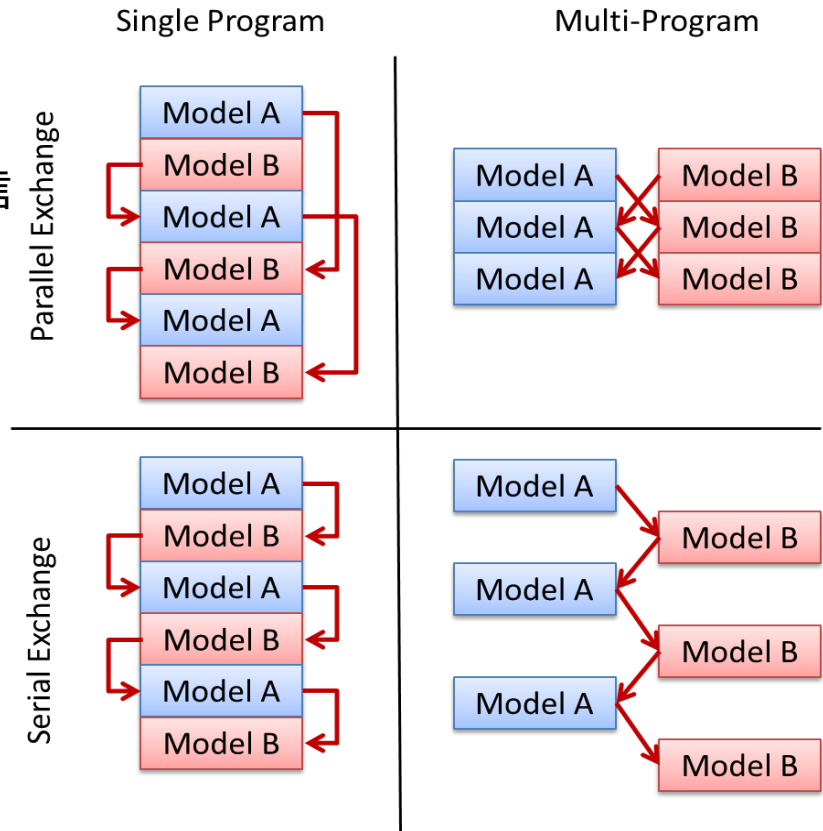
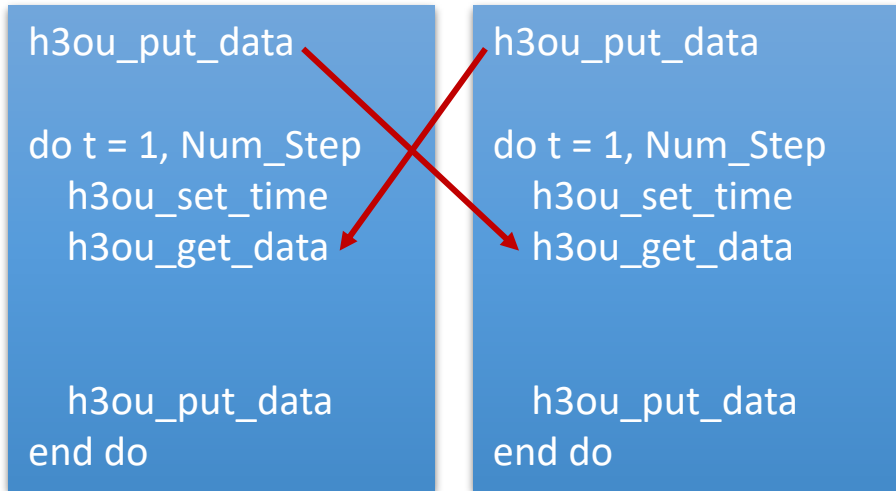
Data exchange timing



Grid remapping

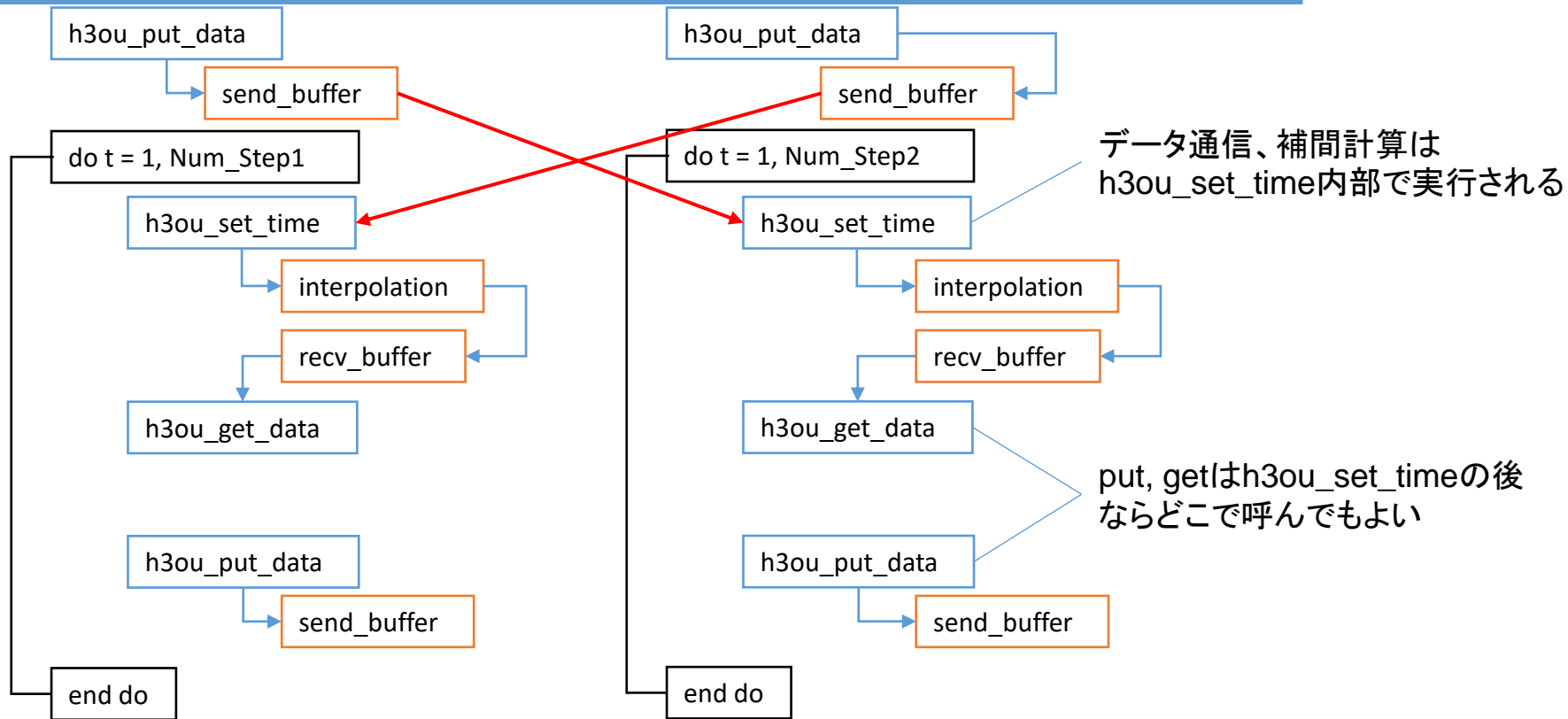
# h3-Open-UTIL/MPの連成方式

- プログラムの実行とデータ交換
  - マルチプログラム：2モデルが並列に実行される
  - 並列データ交換：相互に1ステップ前のデータを受信





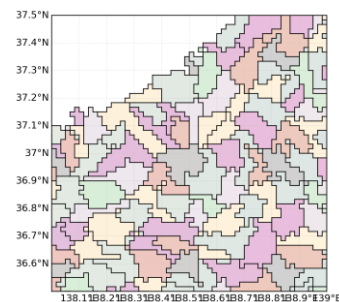
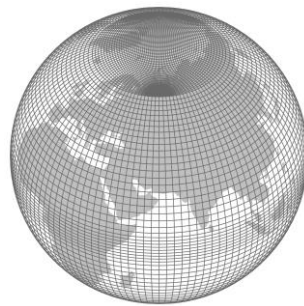
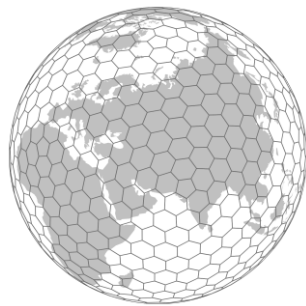
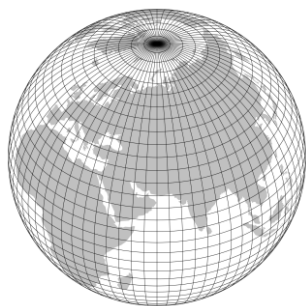
# もう少し詳しいデータの流れ



# h3-Open-UTIL/MPの特徴 (基本機能)

## • 汎用性

- 幅広い格子系に対応
  - 格子形状を入力情報としない
  - 格子点番号と補間テーブルの2つが入力情報
- 補間計算を利用者が規定できる
  - 補間に関わる格子点番号と補間係数をAPIを通じて与える
  - 補間ルーチンは外部に公開されており利用者が自由にコーディング可能



# 一般的な補間アルゴリズム

- 補間計算

- 受信側の格子点値は送信側の複数の格子点値と係数から計算される

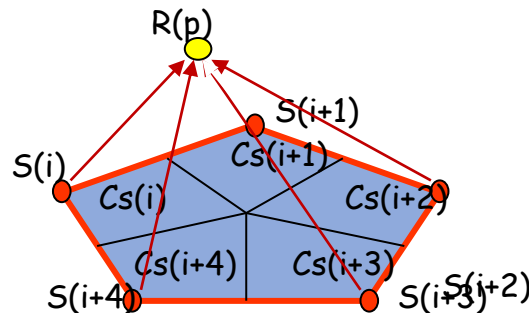
- 補間アルゴリズム

- 距離重み法
- 面積重み法
- バイリニア
- トリリニア

補間元(送信側)格子点の選択方法も係数の計算方法も異なる



必要と思われるすべてのアルゴリズムをカプラ内に実装するのは現実的ではない



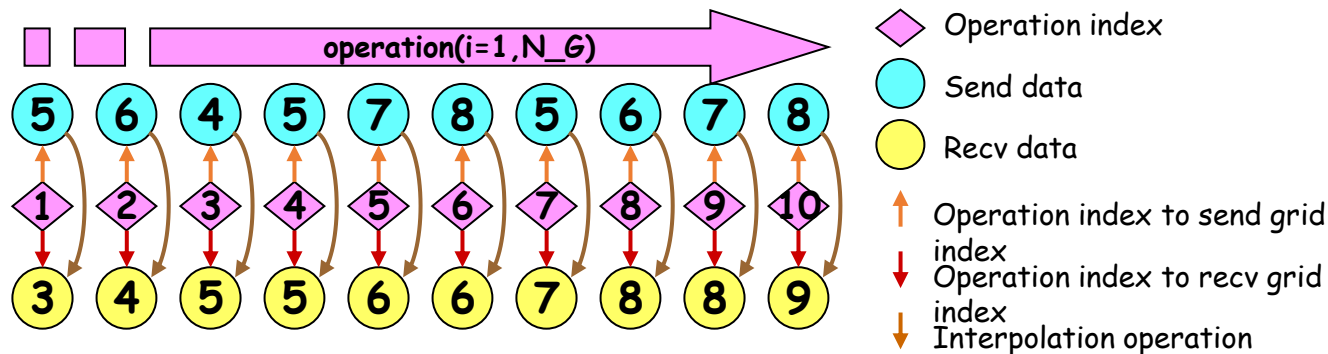
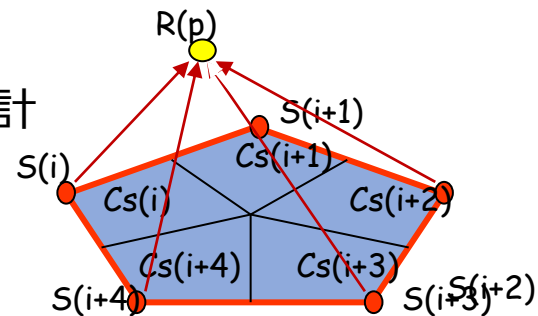
# 補間計算と補間テーブル

- 補間計算

- 受信側の格子点値は送信側の複数の格子点値と係数から計算される

- 補間テーブル

- 受信側格子点番号・送信側格子点番号・補間係数の列
- h3-Open-UTIL/MPではこれらの列をAPIの引数として与える



# h3-Open-UTIL/MPの特徴 (拡張機能)

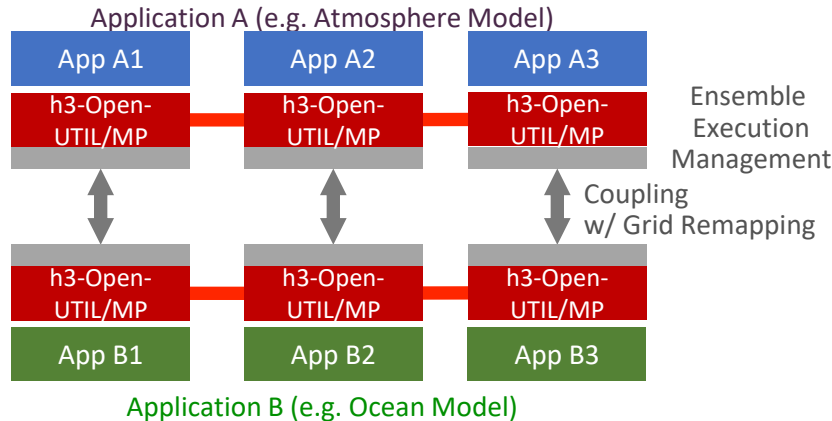
## • アンサンブル連成

### • アンサンブル計算

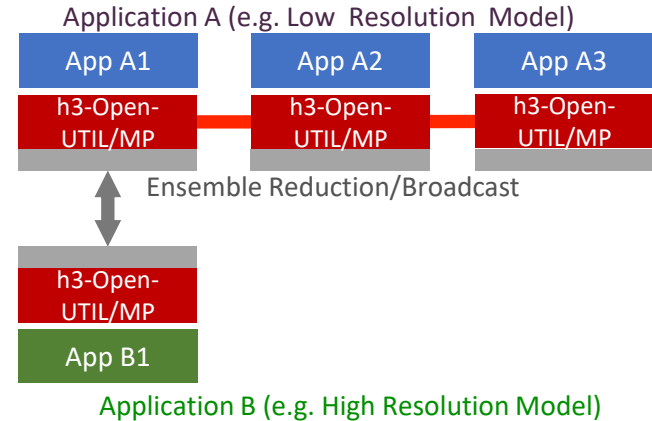
➤ 条件をわずかに変えたシミュレーションを多数実行し統計的な描像を得る手法

### • 連成モデルに対してアンサンブル計算可能

• アンサンブル方向の統計量もカプラが計算



**$M \times (A + B)$  execution**



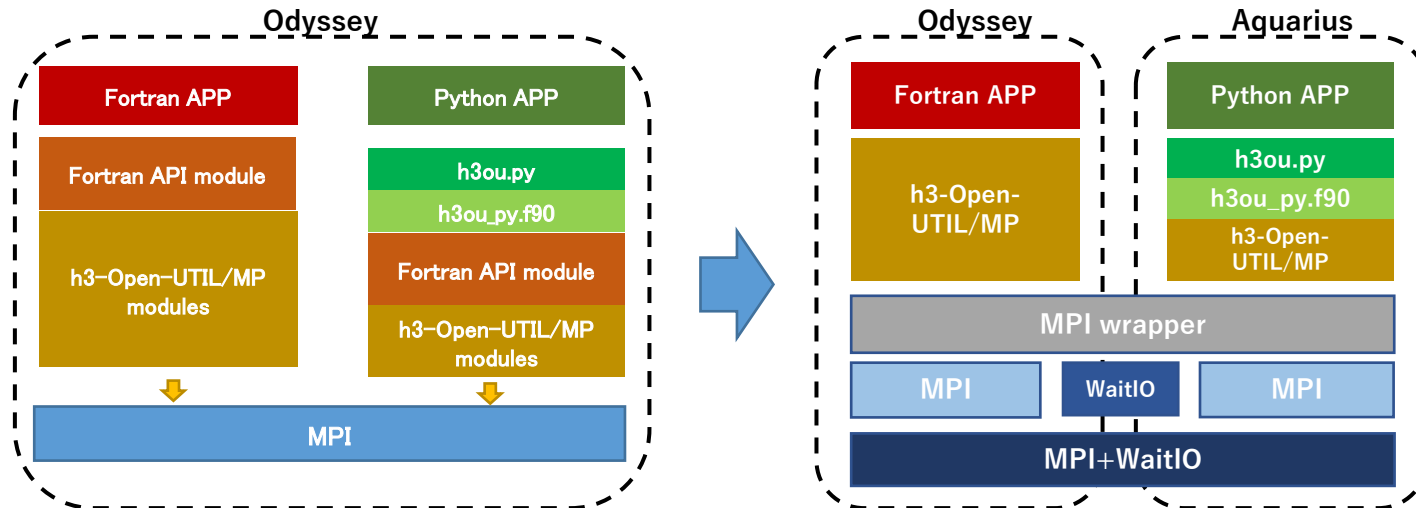
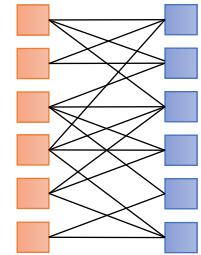
**$(M \times A) + B$  execution**

# h3-Open-UTIL/MPの特徴 (拡張機能)

- 異機種間連成とPythonインターフェース

- h3-Open-SYS/WaitIOと連携

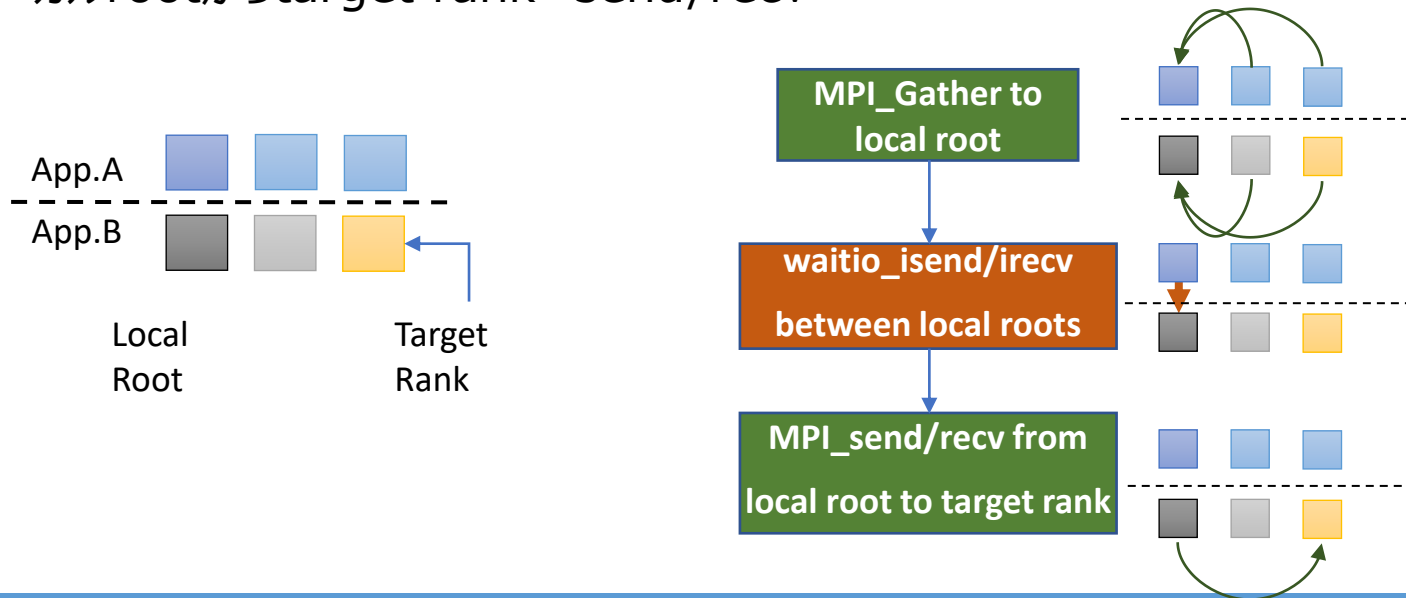
- モデル内の通信 : MPI
    - モデル間の1対1通信 : WaitIO(waitio\_isend, waitio\_irecv, waitio\_wait)
    - グローバル通信 : MPI+WaitIO



# グローバル通信の例

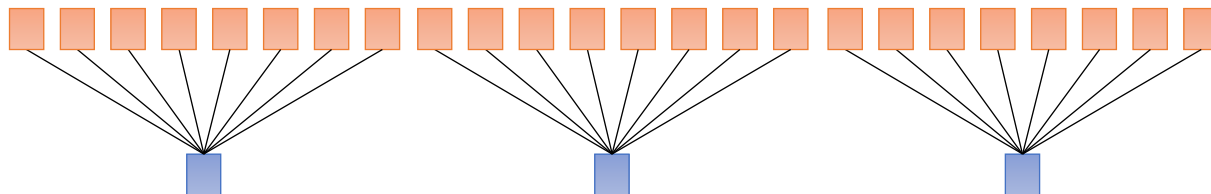
## • MPI\_Gather相当

- MPI\_Gatherでローカルrootに集める
- ローカルroot間でwaitio\_isend/waitio\_irecv
- ローカルrootからtarget rankへsend/recv



# 性能評価例

- 使用マシン
  - Wisteria-BDEC01:Odyssey + Aquarius (WaitIO/Socket + WaitIO/File)
  - 不老: Type I + Type II (WaitIO/Fileのみ)
- テスト条件
  - Odyssey, Type I : 160 プロセス, 40ノード
  - Aquarius, Type II : 20プロセス1ノード
- 測定項目
  - トイモデルで時間積分ループ中の通信時間を測定: waitio\_isend/waitio\_irecvの通信

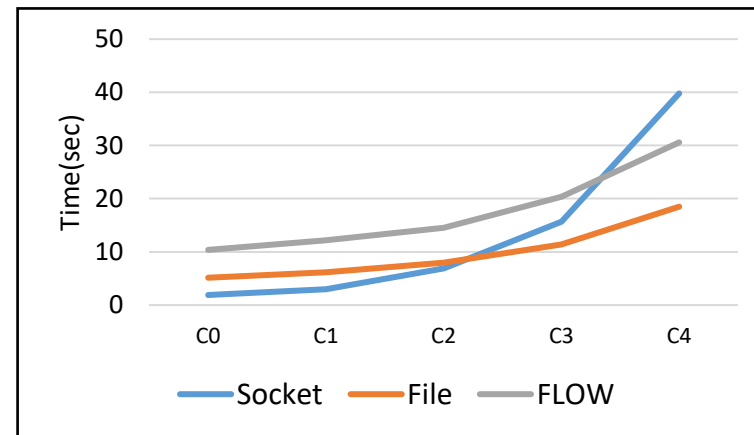




# 性能評価例

- 実験条件
  - 交換データサイズを2倍ずつ変化させて時間を測定
- 実験結果
  - データサイズが小さい場合はSocketが高速
  - データサイズに対する時間増加割合はFileの方が小さく、データサイズが大きくなるとFileが高速になる
  - Wisteriaと不老の比は約2倍←ハードウェアの性能に比例

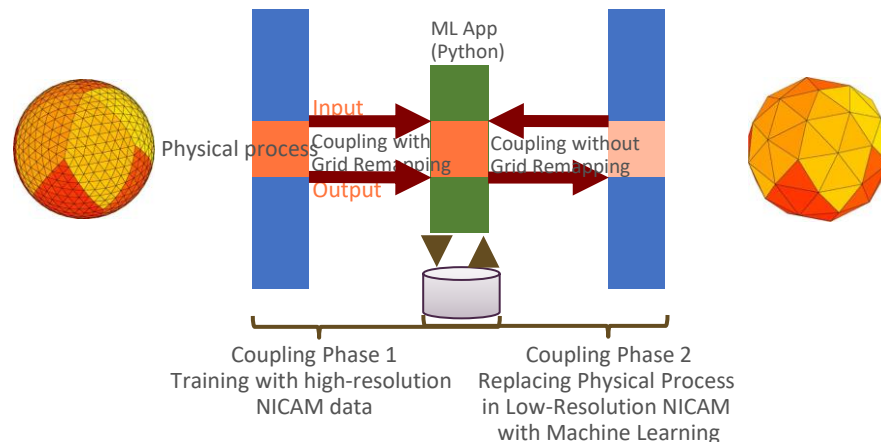
Case	# of grid	total size	size/process
C0	81920	26214400	163840
C1	163840	52428800	327680
C2	327680	1.05E+08	655360
C3	655360	2.1E+08	1310720
C4	1310720	4.19E+08	2621440



# h3-Open-UTIL/MP+WaitIOの適用事例

## • 大気モデル-AI連成

- 雲は気象・気候のシミュレーションに極めて大きな影響
- 雲を解像するには10km以下の解像度が必要で気候シミュレーションには実現困難
- 高解像度の雲物理計算をAIで学習し低解像度のシミュレーションに適用する



# 実験のデザイン

## 対象モデル

- NICAM: 正二十面体格子の非静力学全球大気モデル
- 水平格子数10240, 鉛直40層

## AI

- FrameworkはPyTorch
- 学習は3層のMLP

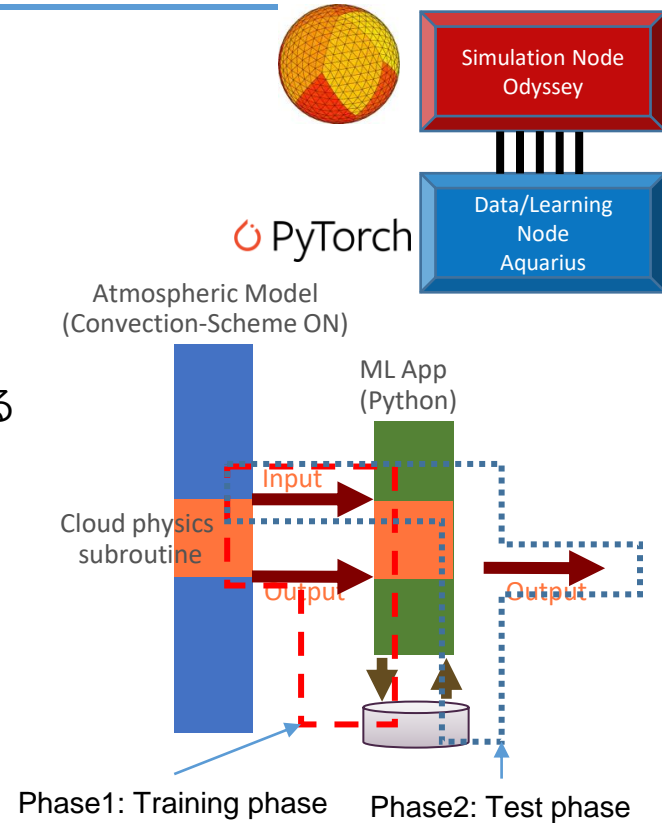
## 実験方法

- NICAMの雲物理サブルーチンの入力データから出力データを再現するように学習
- 学習結果と入力データから出力データを再現

## 用いたデータ

- 入力は大気密度( $\rho$ ), 内部エネルギー( $e_{in}$ ), 水蒸気密度( $\rho_q$ )の3変数
- 出力は上記3変数の時間変化量

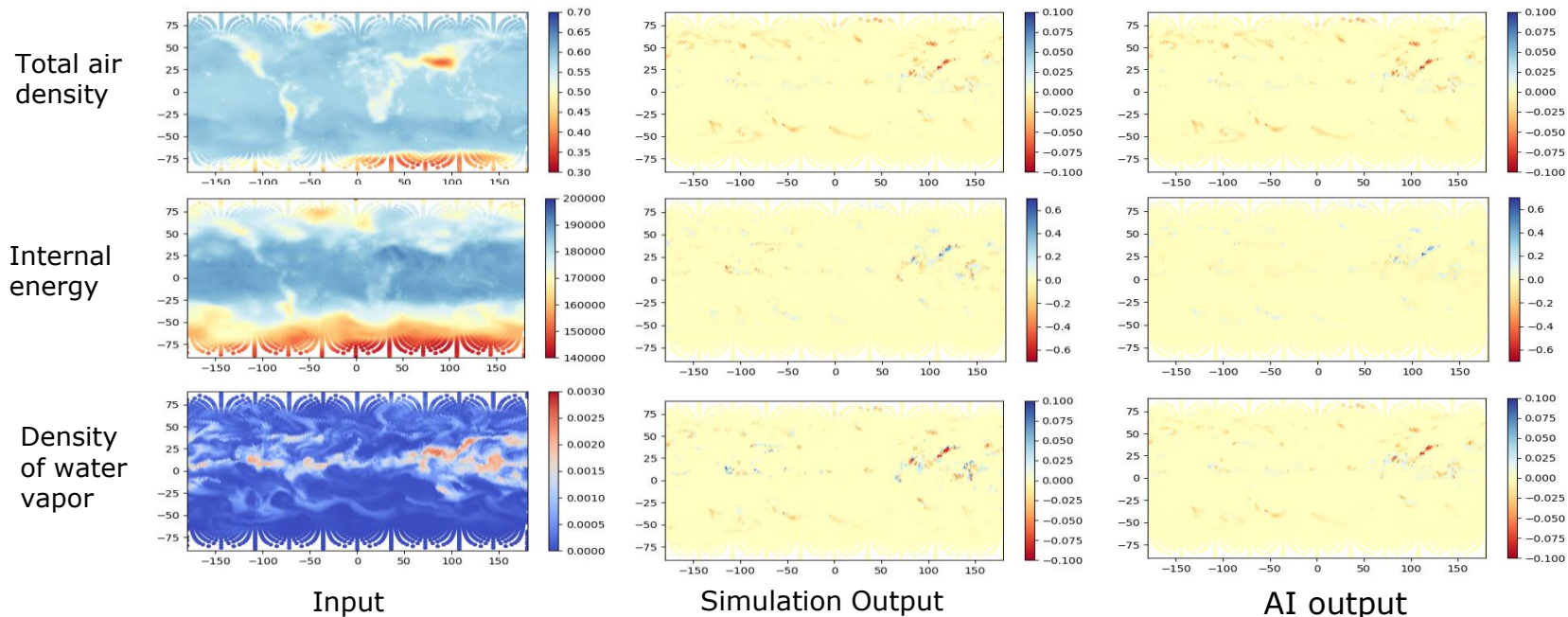
$$\frac{\Delta \rho}{\Delta T} \quad \frac{\Delta e_{in}}{\Delta T} \quad \frac{\Delta \rho_q}{\Delta T}$$



# 実験結果

## モデル計算とAI出力の比較

- 分布はよく再現されているが極値の再現性が不十分



# 実験のまとめ

Correlation between reference and AI output

	slope	intercept	correlation coef.
air density	0.53847887	0.00017136	0.6194595
internal energy	0.58866578	-0.00019053	0.6769882
water vapor	0.56042855	0.000008773	0.6494912

- AIによるテスト計算ではそれらしい場合は再現するが極値の再現性が不十分
- 改良点は多々ある
  - テスト変数の選択（サブルーチンの入出力変数はIN:23, OUT:27, INOUT:11）
  - データのプリプロセス（0近傍の値が多いのでそちらに引きずられる）
  - 適切な学習アルゴリズムの採用（時空間構造を反映できるような）

が、何はともあれカプラの機能としては目的を果たした

# まとめと今後の展開

---

- h3-Open-UTIL/MP
  - **汎用**カップリングライブラリ
  - 異機種間連成
  - アンサンブル連成
  - PythonAPI
- 今後の展開
  - 異機種間連成とPythonAPIについては実アプリに適用
  - アンサンブル連成の実アプリ適用を行う

# アンサンブル連成の実験デザイン

- NICAMのデータ同化システムにアンサンブル連成を導入
  - 低解像度NICAM x N + 高解像度NICAM x 1
    - Z, P, T, VX, VY, VZ

