

AI for Science:「カプラ」を通して ソフトウェアの持続可能性を探究する

中島 研吾

東京大学情報基盤センター

理化学研究所計算科学研究センター

第16回 自動チューニング技術の現状と応用に関するシンポジウム(ATTA2024)
2024年12月26日 工学院大学

共著者

- 住元真司(東京大学情報基盤センター)
- 荒川 隆(CliMTech・東京大学情報基盤センター)
- 八代 尚(国立環境研究所)
- 胡 曜(東京大学情報基盤センター)
- 山崎一哉(東京大学情報基盤センター)



Acknowledgements



東京大学
THE UNIVERSITY OF TOKYO

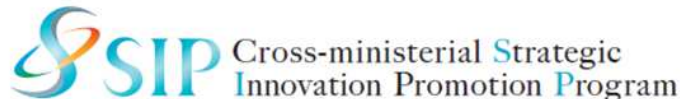


東京大学情報基盤センター
INFORMATION TECHNOLOGY CENTER, THE UNIVERSITY OF TOKYO

- JST-CREST, DFG-SPPEXA
- JSPS Grant-in-Aid for Scientific Research (S) (19H05662)
- New Energy & Industrial Technology Development Organization (NEDO): Cross-ministerial Strategic Innovation Promotion Program (SIP): Big-Data and AI-Enabled Cyberspace Technologies
- JHPC-quantm (NEDO)
- Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures (JHPCN)
- Information Technology Center, The University of Tokyo



新エネルギー・産業技術総合開発機構
New Energy and Industrial Technology Development Organization



Coupler: カプラ, カプラー

- In this talk, I focus on *coupler*.
- 複数の科学技術アプリケーションを弱連成 (loosely couple) させるツール: *Multiphysics Coupler* (マルチフィジックスカプラ): 大気・海洋, 構造・流体
- 科学技術アプリケーションは通常様々な計算プロセスを含む
 - メッシュ生成等前処理, メインソルバー, 可視化等後処理など
 - カプラが連成させるのは通常は「メインソルバー(シミュレーション)」
- 昨今, 「シミュレーション・データ同化・機械学習」を含むような複雑なワークフローが一般的になりつつある(例「S+D+L」融合)
- **各部分の最適化, 高速化はもちろん重要だが, 計算全体の実行時間(overall time-to-solution)を短縮することが最も重要**
- 最近10年余, 中島が関わってきたプロジェクト群では, カプラ(coupler)は重要な役割を果たし, 効率的なワークフローツールとしての進化を遂げてきた
- この歴史を振り返ることで, ソフトウェアの持続可能性(sustainability)について考える



**Wisteria
BDEC-01**



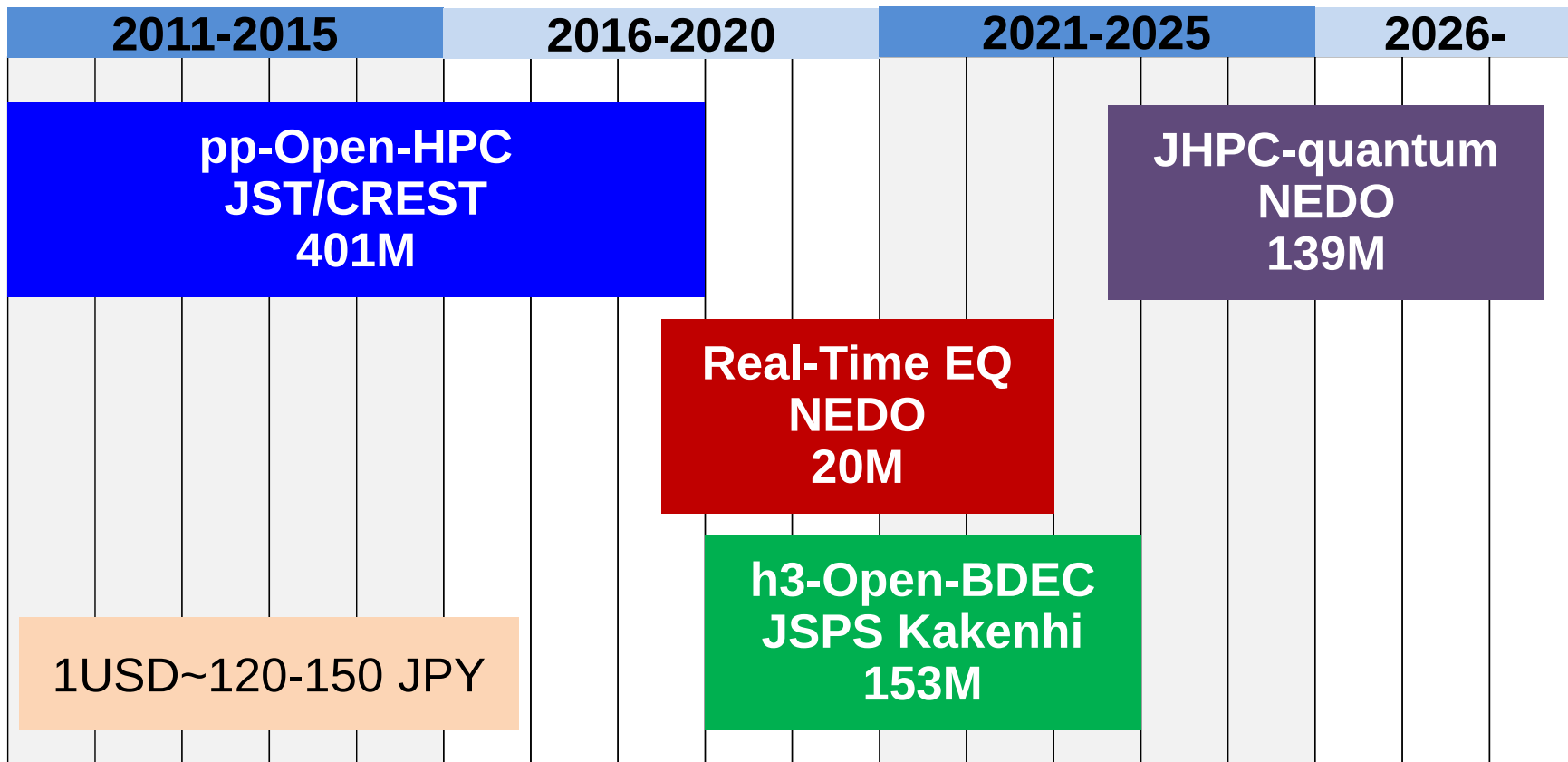
Towards Sustainable Promotion of Scientific Discovery by Integration of Simulation/Data/Learning (S+D+L) on Exascale Supercomputers



Kengo Nakajima
Information Technology Center
The University of Tokyo
RIKEN R-CCS

Sustainable Project Pathways for HPC Software and Applications WS
IEEE eScience 2024, September 17, 2024, Osaka, Japan

History of the Projects & Budgets (JPY)



- ppOpen-HPC
- h3-Open-BDEC on Wisteria/BDEC-01
- JHPC-quantum

2001-2005

2006-2010

2011-2015

2016-2020

2021-2025

2026-2030

Hitachi SR8000

1,024 GF

SR8000

Hitachi SR11000

J1, J2

5.35 TF, 18.8 TF

IBM Power5+

Hitachi SR16K/M1

Yayoi

54.9 TF

IBM Power7

Intel CLX

OBCX

(Fujitsu)

6.61 PF

疑似ベクトル

汎用CPU

加速装置付

NVIDIA GH200 +

Intel SPR/HBM

Miyabi/OFP-II

80+ PF

Hitachi

SR2201

307.2GF

HARP-1E

Hitachi

SR8000/MPP

2,073.6 GF

SR8000

Hitachi HA8000

T2K Todai

140 TF

AMD Opteron

Intel Xeon Phi

Oakforest-
PACS (Fujitsu)

25 PF

Fujitsu FX10

Oakleaf-FX

1.13 PF

SPACR64 IXfx

Reedbush-
UI/H/L (SGI-HPE)

3.36 PF

Intel BDW +

NVIDIA P100

Wisteria
BDEC-01 Fujitsu

33.1 PF

A64FX,
Intel Icelake+
NVIDIA A100

BDEC-

02

150+ PF

Accelerators

Ipomoea-01 25PB

Ipomoea-03

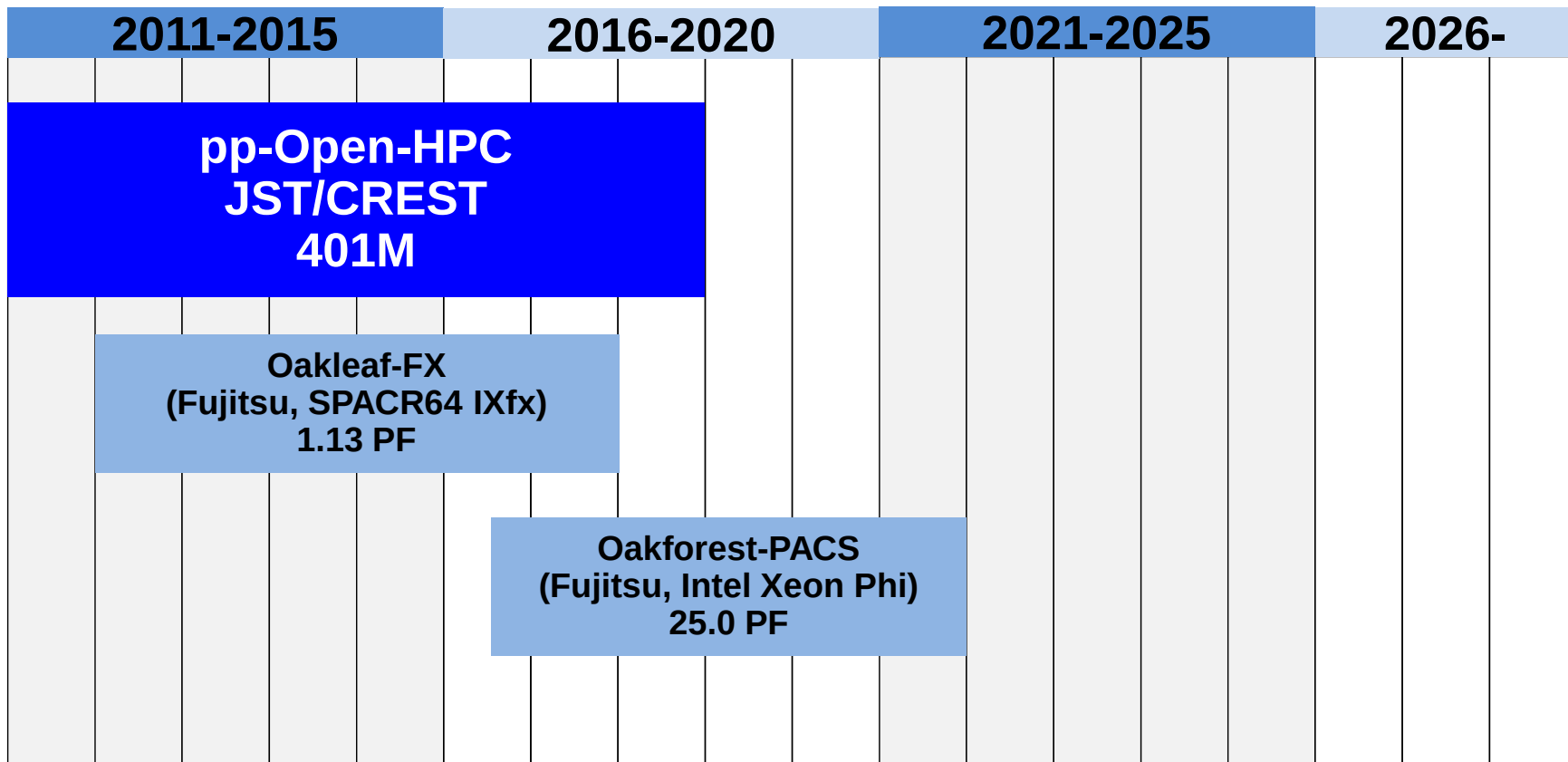
Ipomoea-02

東京大学情報基盤 センターのスパコン

利用者2,600+名

55%は学外

History of the Projects & Budgets (JPY)





ppOpen-HPC: Overview

- Application Framework with Automatic Tuning (AT)

✓ “pp” : post-peta-scale

- Five-year project (FY.2011-2015) (since April 2011)

✓ P.I.: Kengo Nakajima (ITC, The University of Tokyo)

✓ Part of “Development of System Software Technologies for Post-Peta Scale High Performance Computing” funded by JST/CREST (Supervisor: Prof. M. Sato, RIKEN R-CCS)

✓ Finally, it was extended to FY.2018 under collaborations with German Projects (SPPEXA/DFG)

- Team with 7 institutes, >50 people (5 PDs) from various fields: Co-Design

- Open Source Software

✓ <http://ppopenhpc.cc.u-tokyo.ac.jp/>

✓ <https://github.com/Post-Peta-Crest/ppOpenHPC>

✓ English Documents, MIT License



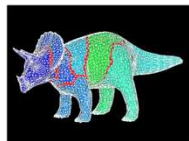


Framework
Appl. Dev.

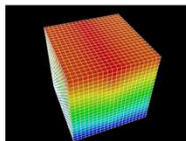
Math
Libraries

Automatic
Tuning (AT)

System
Software



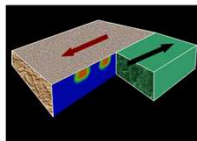
FEM
Finite Element Method



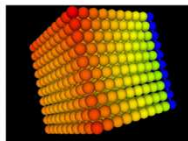
FDM
Finite Difference Method



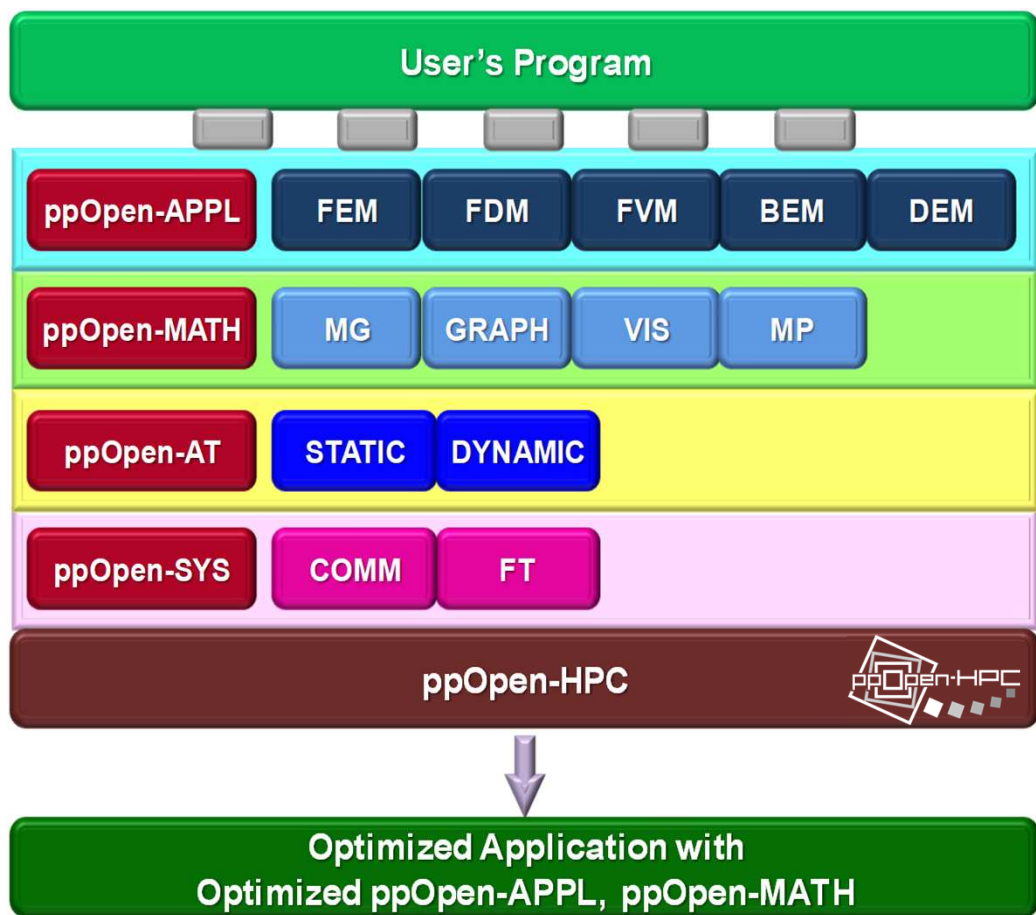
FVM
Finite Volume Method



BEM
Boundary Element Method



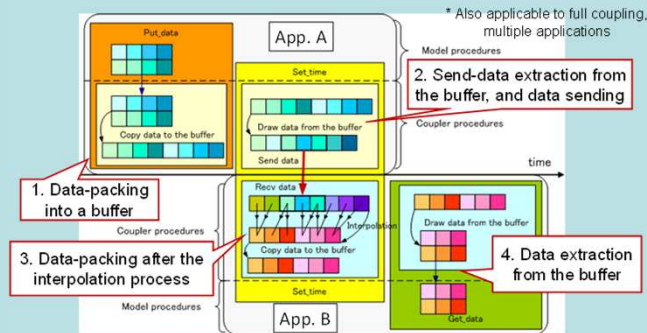
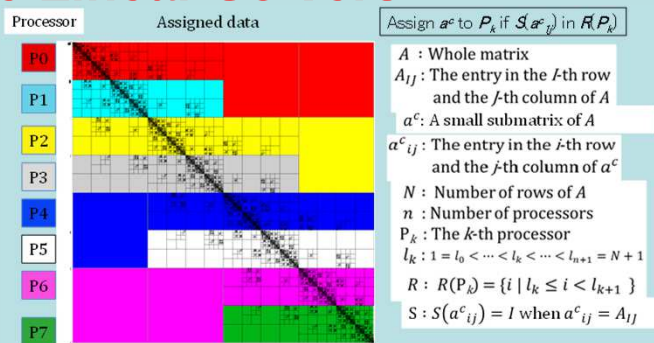
DEM
Discrete Element Method





Featured Developments

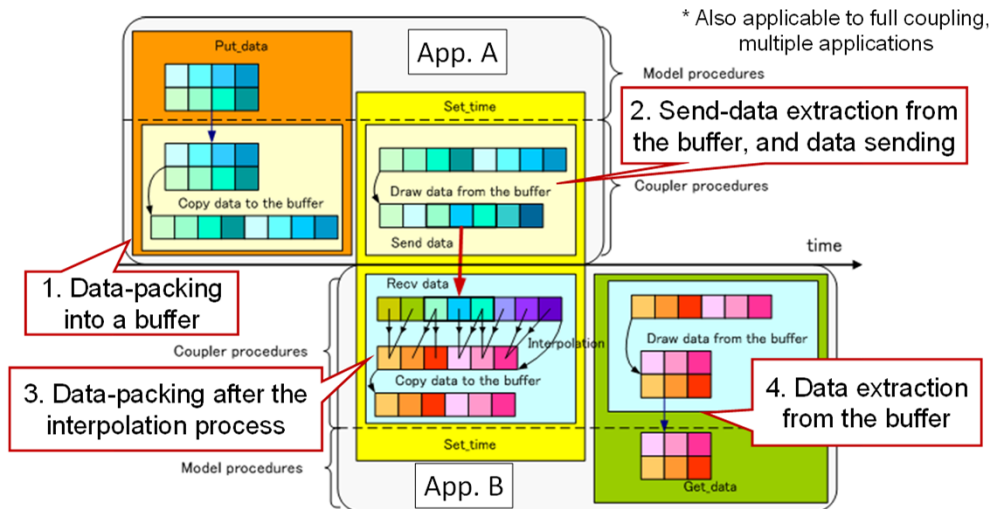
- ppOpen-AT: AT Language for Loop Optimization
- HACApK library for H-matrix comp. in ppOpen-APPL/BEM (OpenMP/MPI Hybrid Version)
 - First Open Source Library by OpenMP/MPI Hybrid
- **ppOpen-MATH/MP (Coupler for Multiphysics Simulations, Loose Coupling of FEM & FDM)**
- **Sparse Linear Solvers**



ppOpen-MATH/MP

Weak-Coupling of Multiple Applications

- Each application does a single computation
- Ocean-Atmosphere
- Fluid-Structure



- Originally developed as a coupler for NICAM (atmosphere, unstructured), and COCO (ocean, structured) in global climate simulations using K computer
- Non-Centralized/Fully Distributed Operations, No Master Process**
- Developed coupler is extended to more general use.

全球気候シミュレーション 大気-海洋連成(NICAM-COCO)

• 物理モデル

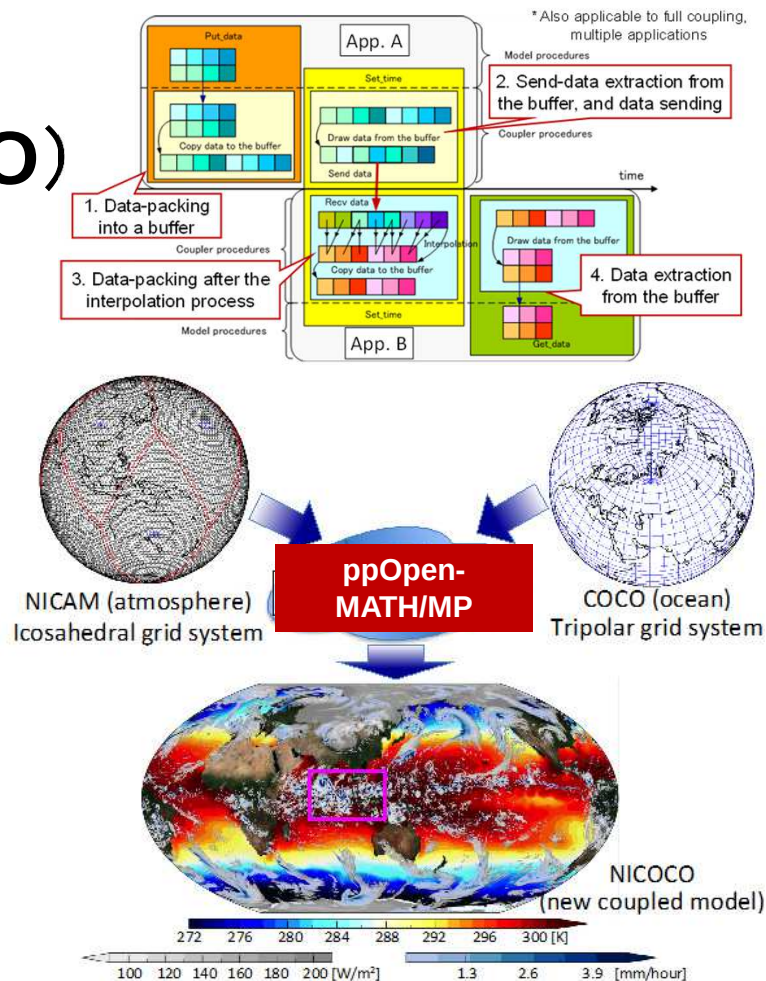
- 大気: NICAM 全球大気の「積乱雲」解像
- 海洋: COCO 全球海洋の「渦」解像

• ppOpen-MATH/MPカプラー(Coupler)

- マルチフィジックス連成のためのツール

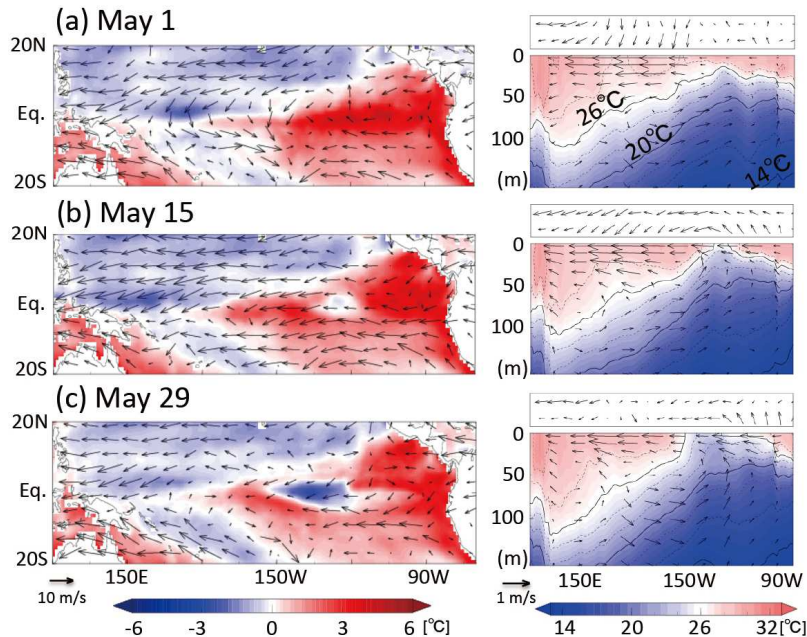
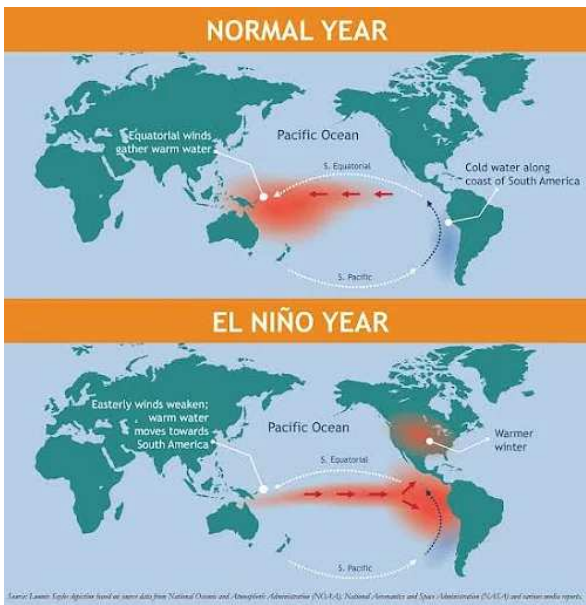
• 計算規模(2ヶ月シミュレーション)

- 大気: 14km, 1.1億格子, 17万ステップ
- 海洋: 0.25-1.00°, 580万格子, 6千ステップ
- 京コンピュータ(640+32)ノード, 28時間(2017年頃)



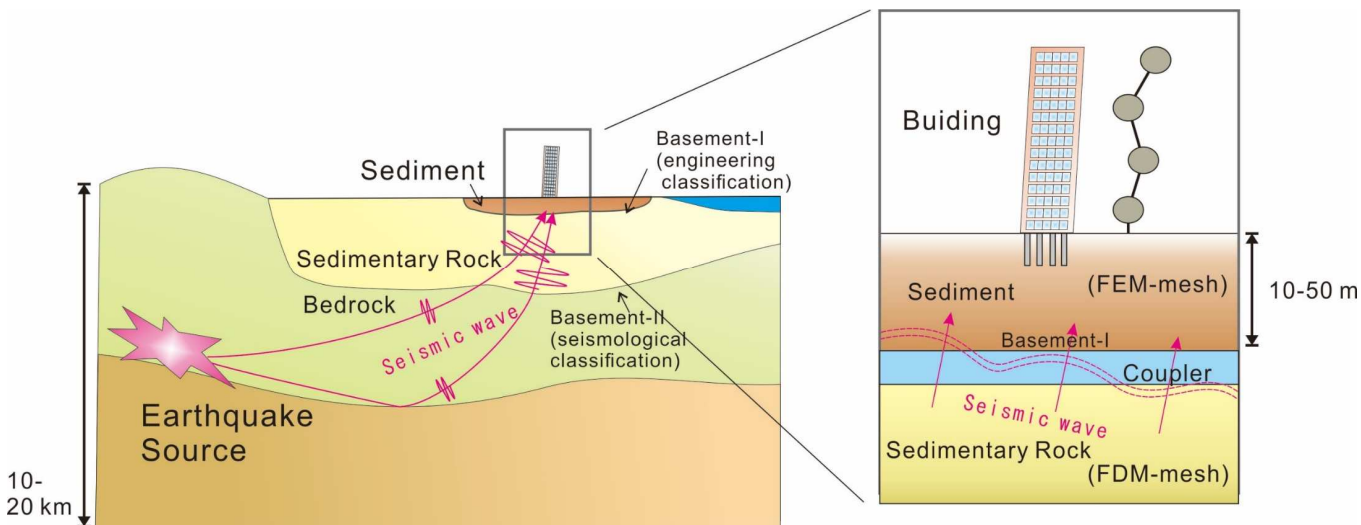
Super El Niño

1997/1998年のスーパーエルニーニョの突然の終了を全球大気海洋連成シミュレーション(NICAM-COCO)によって解明
カプラ(ppOpen-MATH/MP)については言及ナシ



[宮川, 八代, 佐藤他]

Coupling Simulation of Seismic Waves and Building Vibrations

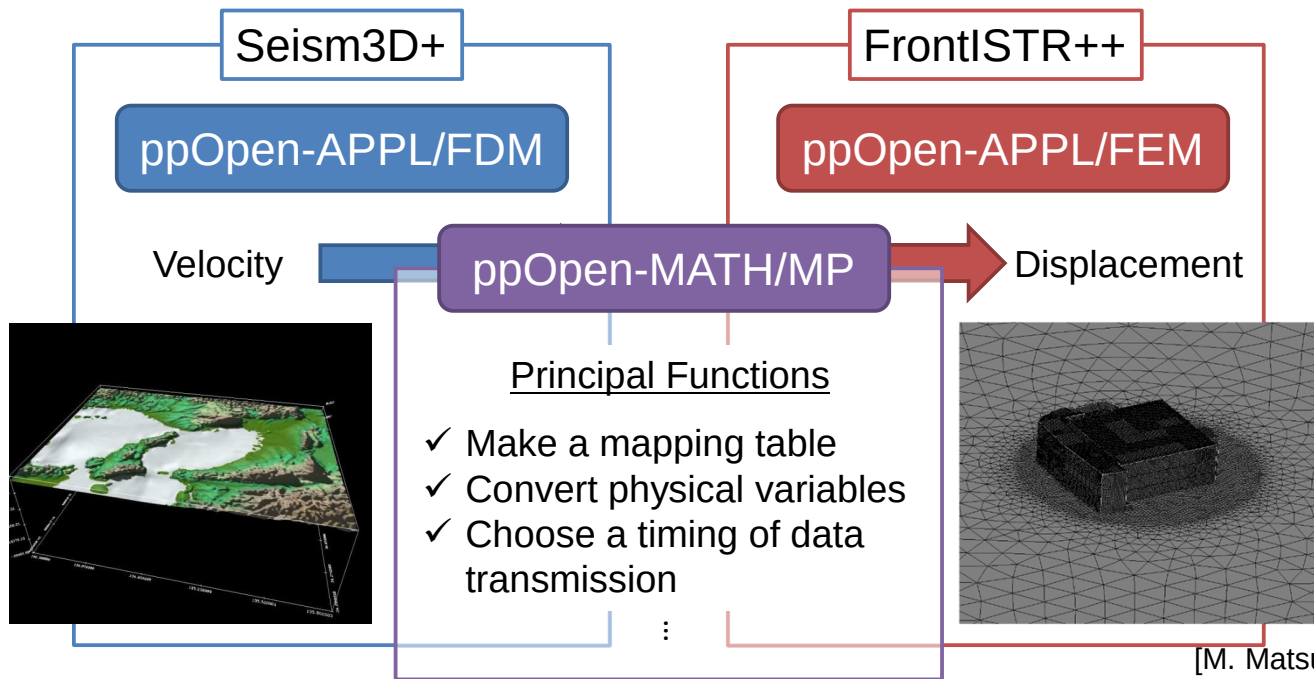


The coupling simulation refers to one-way data communication from **FDM** (seismic wave propagation) to **FEM** (dynamic structure).



Weak-Coupled Simulation by the ppOpen-HPC Libraries

Two kinds of applications (Seism3D+ based on FDM, and FrontISTR++ based on FEM) are connected by the ppOpen-MATH/MP coupler.





Practical Simulation on Oakleaf-FX

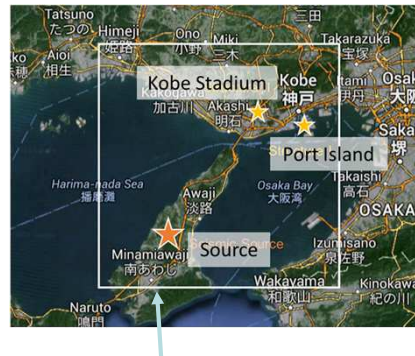
The simulation target is the earthquake that occurred at Awaji Island on 13 April, 2013. The computational domain of Seism3D+ is 60 km² from Awaji Island and that of FrontISTR++ is the actual building of RIKEN Advanced Institute for Computational Science (AICS), Port Island, Kobe, modeled by an unstructured mesh.

Seism3D+ Grid Points $(x, y, z) = (1536, 1536, 1600)$
Parallelization 2560 processes/16 threads

FrontISTR++ Grid Points 600 million (AICS building)
Parallelization 1000 processes/16 threads
(@Port Island)
1000 processes/16 threads
(@Kobe Stadium)

Total 4560 nodes on Oakleaf-FX

(**Seism3D+**: 2560 nodes, **FrontISTR++**: 2000 nodes)

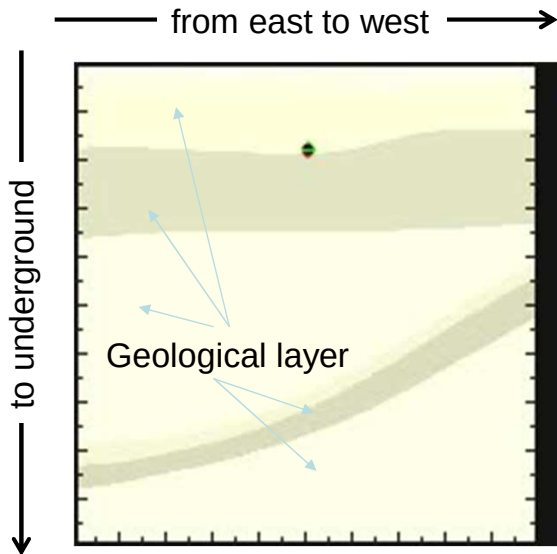


Computational domain
of **Seism3D+**

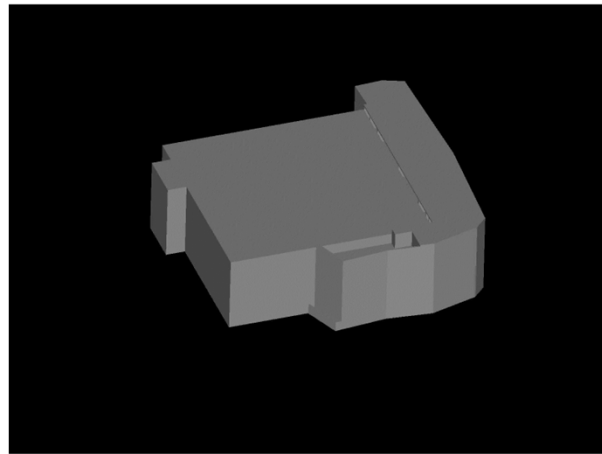
[M. Matsumoto et al. 2015]



**2,560 nodes for FDM, 2,000 nodes
for FEM = 4,560 nodes of FX10**



Seismic wave propagation by **Seism3D+**
(Red:P-wave, Green:S-wave)



Building vibration by **FrontISTR++**

2,560 nodes for FDM, 2,000 nodes for FEM = 4,560 nodes of FX10

- ✓ Coupling simulation was executed on large-scale computational resources of Oakleaf-FX supercomputer system.
- ✓ Seismic wave propagations (**Seism3D+**) for the simulation time of 90 sec., and building vibrations (**FrontISTR++**) for the simulation time of 20 sec. were calculated.

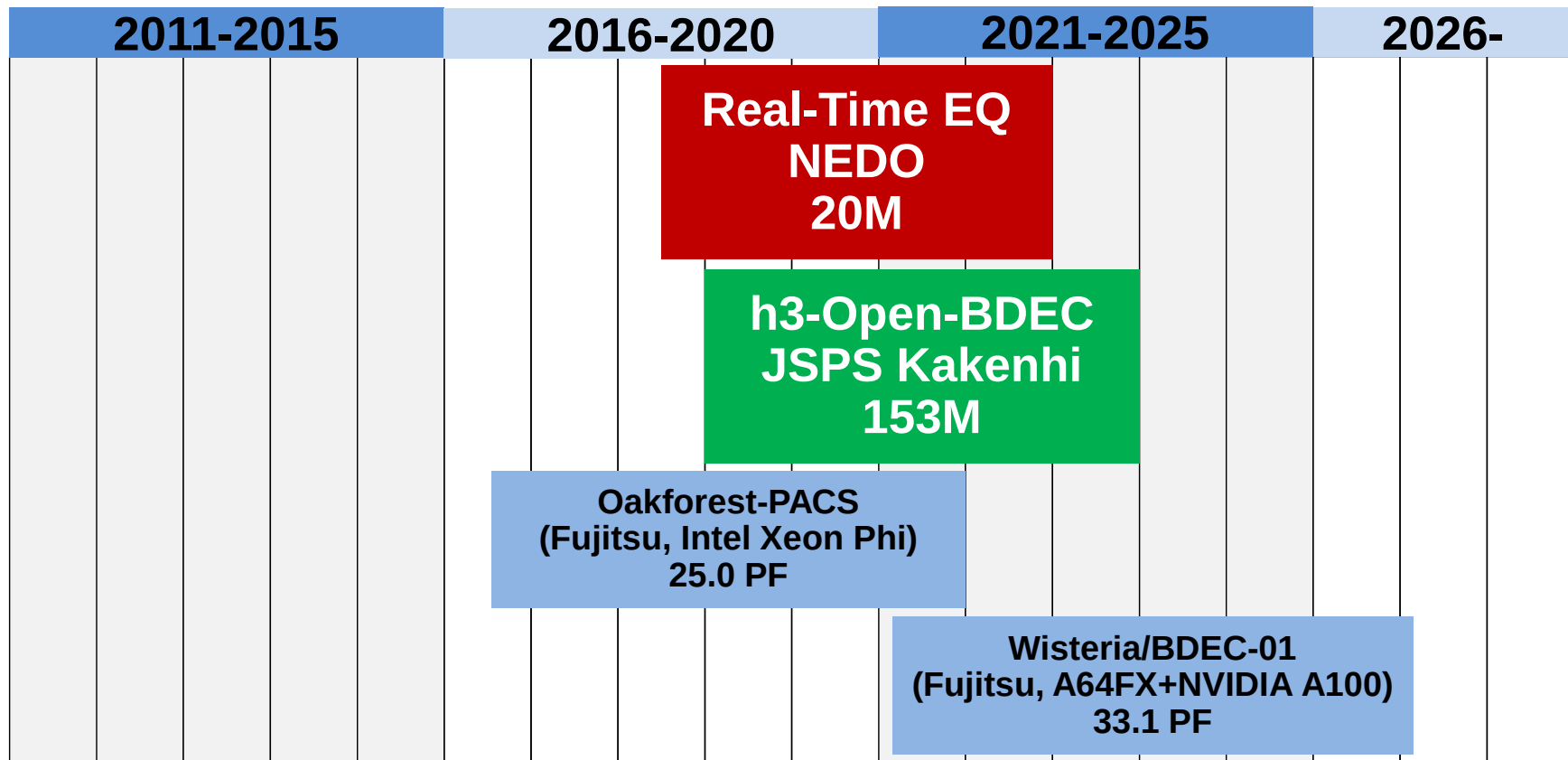
Comparison between sim. time and exe. time

	Sim. Time	Exe. Time
Seism3D+	90 sec.	6 hours
FrontISTR++	20 sec.	16 hours

- ✓ It was revealed that the manner in which memory allocation occurs in the coupler has some problem when such a large-scale simulation is performed.

- ppOpen-HPC
- **h3-Open-BDEC on Wisteria/BDEC-01**
- JHPC-quantum

History of the Projects & Budgets (JPY)



2001-2005

2006-2010

2011-2015

2016-2020

2021-2025

2026-2030

Hitachi SR8000

1,024 GF

SR8000

Hitachi SR11000

J1, J2

5.35 TF, 18.8 TF

IBM Power5+

Hitachi SR16K/M1

Yayoi

54.9 TF

IBM Power7

Intel CLX

OBCX

(Fujitsu)

6.61 PF

疑似ベクトル

汎用CPU

加速装置付

NVIDIA GH200 +

Intel SPR/HBM

Miyabi/OFP-II

80+ PF

Hitachi

SR2201

307.2GF

HARP-1E

Hitachi

SR8000/MPP

2,073.6 GF

SR8000

Hitachi HA8000

T2K Today

140 TF

AMD Opteron

Intel Xeon Phi

Oakforest-

PACS (Fujitsu)

25.0 PF

Fujitsu FX10

Oakleaf-FX

1.13 PF

SPACR64 IXfx

Reedbush-

UI/H/L (SGI-HPE)

3.36 PF

Intel BDW +

NVIDIA P100



Wisteria

BDEC-01 Fujitsu

33.1 PF

A64FX,

Intel Icelake+

NVIDIA A100

BDEC-

02

150+ PF

Accelerators

Ipomoea-01 25PB

Ipomoea-

03

Ipomoea-02

東京大学情報基盤 センターのスパコン

利用者2,600+名

55%は学外

(計算・データ・学習)融合 : (S+D+L)

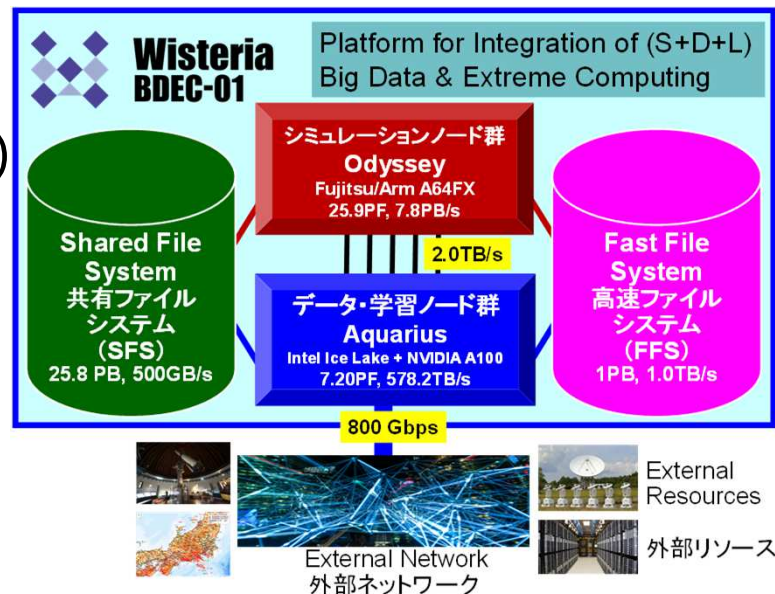
- スパコンのワークロードの多様化 : S・D・L
 - 計算科学, 計算工学 : Simulations (S)
 - 大規模データ解析 : Data (D)
 - AI, 機械学習 : Learning (L)
- 東大情報基盤センターでは, 2015年頃から「(S+D+L)融合」による新しい科学の開拓を実現する, ハードウェア, ソフトウェア, アプリケーション, アルゴリズムに関する研究開発を開始
 - BDEC計画 (Big Data & Extreme Computing)
 - 「データ+学習」による, より高度な「シミュレーション」⇒ AI for HPC
 - 地球科学関連では自然な発想 (すでに実施されている)
- 2021年5月に運用を開始した「Wisteria/BDEC-01」は「BDEC計画」の1号機
 - 「計算・データ・学習 (S+D+L)」融合を実現する, 世界でも初めてのプラットフォーム



Wisteria/BDEC-01

- 2021年5月14日運用開始
 - 東京大学柏Ⅱキャンパス
- 33.1 PF, 8.38 PB/sec. , **富士通製**
 - ~4.5 MVA(空調込み), ~360m²
- Hierarchical, Hybrid, Heterogeneous (h3)
- **2種類のノード群**
 - シミュレーションノード群(S, SIM) : **Odyssey**
 - 従来のスパコン
 - **Fujitsu PRIMEHPC FX1000 (A64FX), 25.9 PF**
 - 7,680ノード(368,640 コア), 20ラック, Tofu-D
 - データ・学習ノード群(D/L, DL) : **Aquarius**
 - データ解析, 機械学習
 - **Intel Xeon Ice Lake + NVIDIA A100, 7.2 PF**
 - 45ノード(Ice Lake:90基, A100:360基), IB-HDR
 - 外部リソース(ストレージ, サーバー, センサーネットワーク他)に直接接続
 - ファイルシステム: 共有(大容量) + 高速

BDEC:「計算・データ・学習(S+D+L)」
融合のためのプラットフォーム
(Big Data & Extreme Computing)



**Wisteria
BDEC-01**

Simulation Nodes Odyssey

25.9 PF, 7.8 PB/s

Fast File
System
(FFS)
1.0 PB,
1.0 TB/s

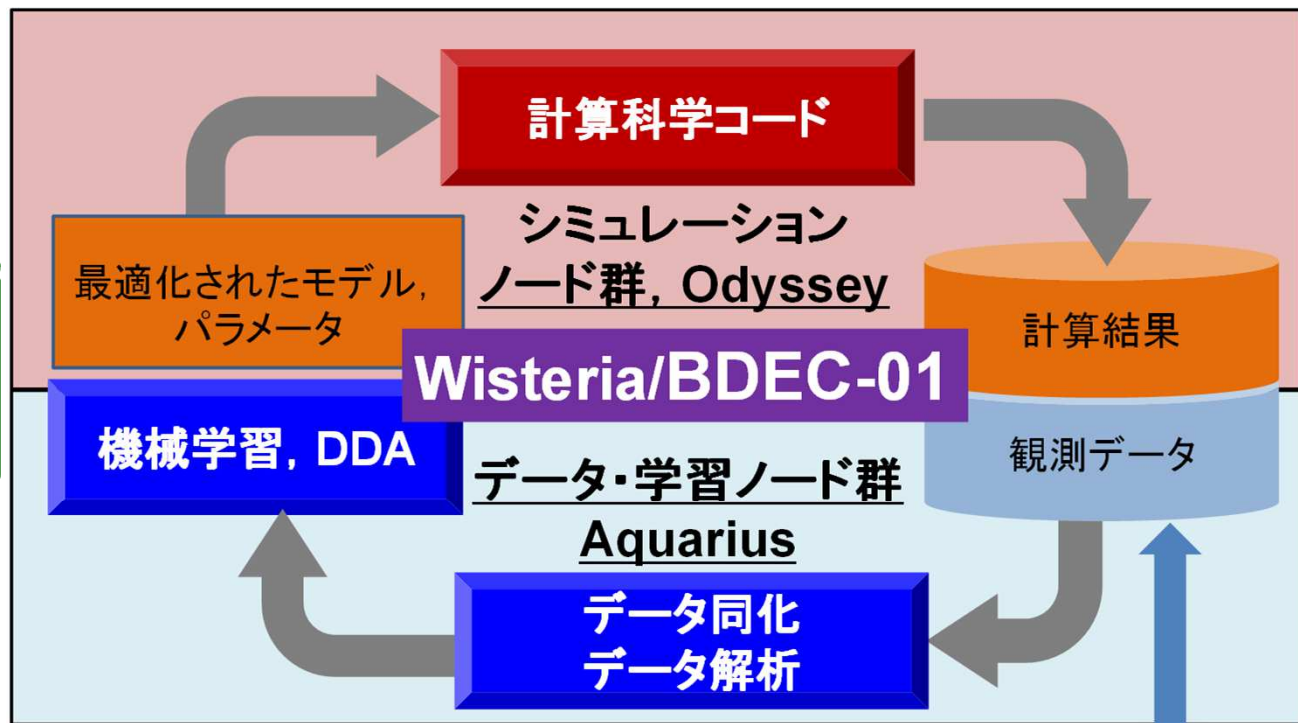
Shared File
System
(SFS)
25.8 PB,
0.50 TB/s

Data/Learning Nodes Aquarius

7.20 PF, 578.2 TB/s



Wisteria BDEC-01



サーバー
ストレージ
DB
センサー群
他



外部ネットワーク



外部
リソース

Simulation Nodes Odyssey

25.9 PF, 7.8 PB/s

Fast File
System
(FFS)
1.0 PB,
1.0 TB/s

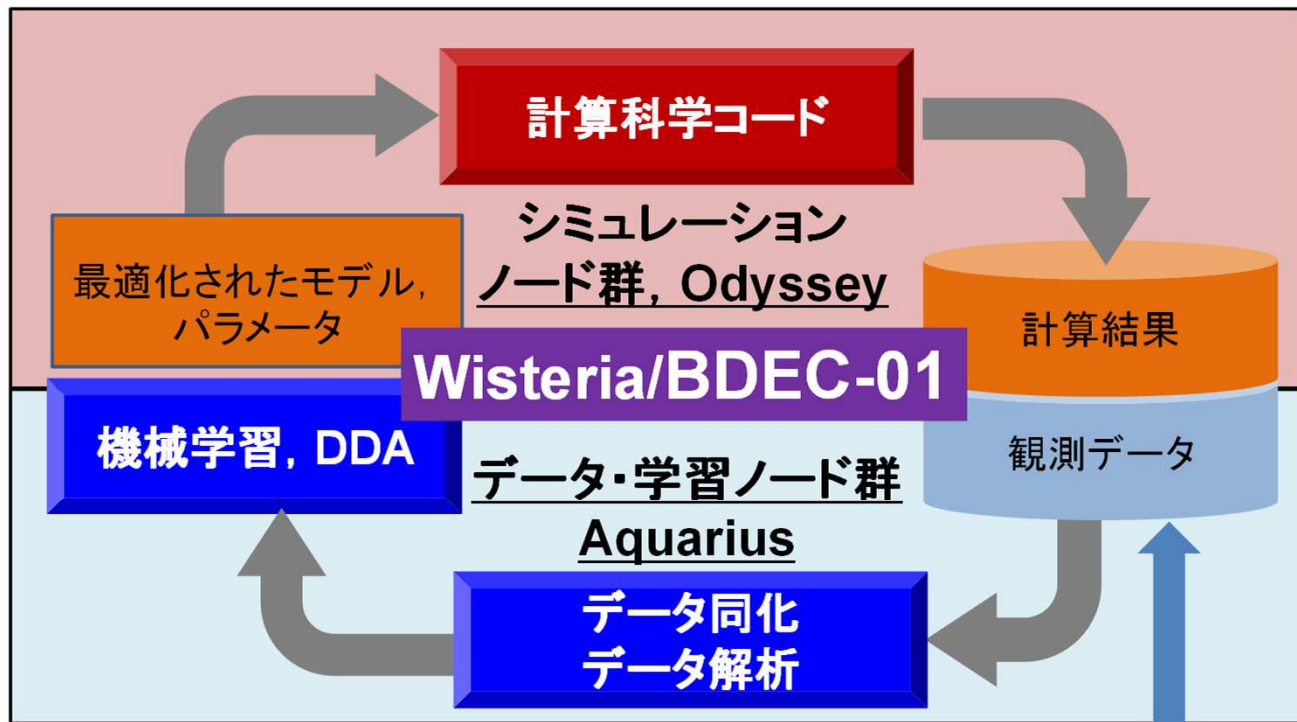
Shared File
System
(SFS)
25.8 PB,
0.50 TB/s

Data/Learning Nodes Aquarius

7.20 PF, 578.2 TB/s



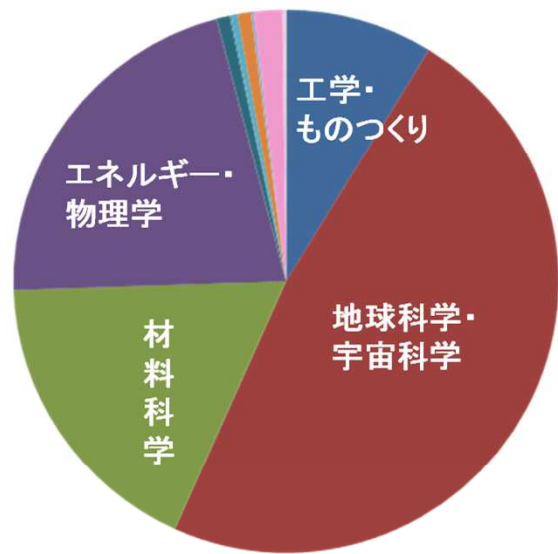
**Wisteria
BDEC-01**



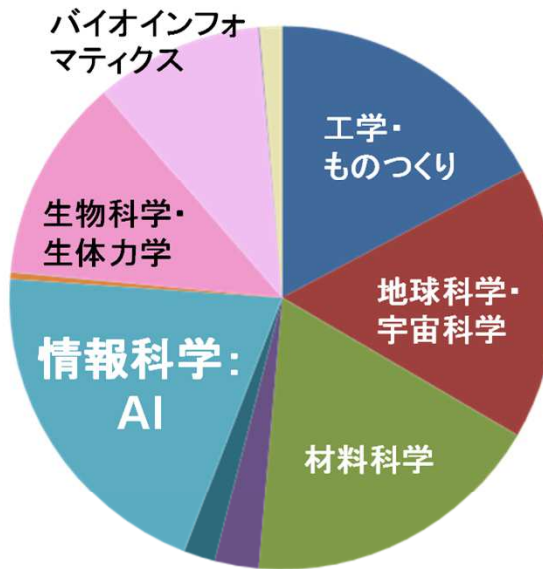
シミュレーションのためのモデル・パラメータのデータ解析, AI/機械学習による最適化 (S+D+L)

2024年度分野別計算資源利用割合(11月末まで)

■ 汎用CPU, ■ GPU



**Odysseey
A64FX**



**Aquarius
A100**

- 工学・ものづくり
- 地球科学・宇宙科学
- 材料科学
- エネルギー・物理学
- 情報科学: システム
- 情報科学: アルゴリズム
- 情報科学: AI
- 教育
- 産業利用
- 生物科学・生体力学
- バイオインフォマティクス
- 社会科学・経済学
- データ科学・データ同化

h3-Open-BDEC

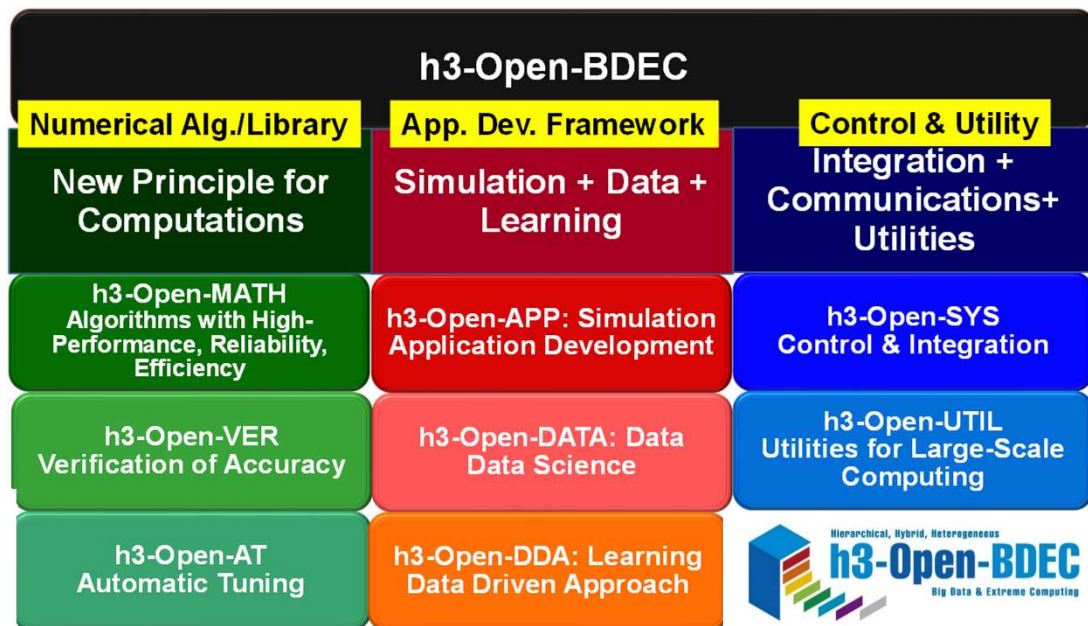
「計算＋データ＋学習」融合を実現する革新的ソフトウェア基盤
科研費基盤研究(S)(2019年度～23年度, 代表: 中島研吾)

<https://h3-open-bdec.cc.u-tokyo.ac.jp/>

Hierarchical,
Hybrid,
Heterogeneous

Big Data &
Extreme
Computing

- ① 変動精度演算・精度保証・自動チューニングによる新計算原理に基づく革新的数値解法
- ② 階層型データ駆動アプローチ等に基づく革新的機械学習手法
- ③ ヘテロジニアス環境 (e.g. Wisteria/BDEC-01) におけるソフトウェア, ユーティリティ群



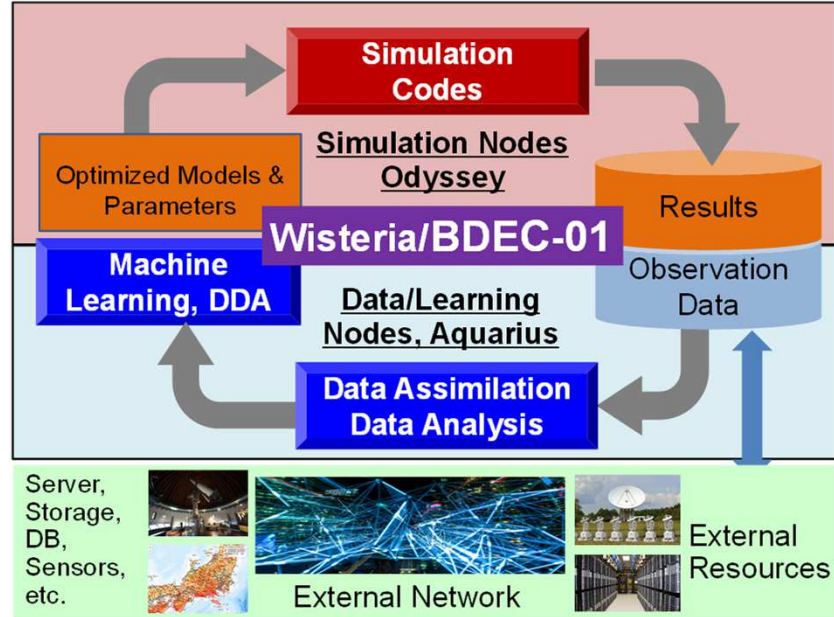
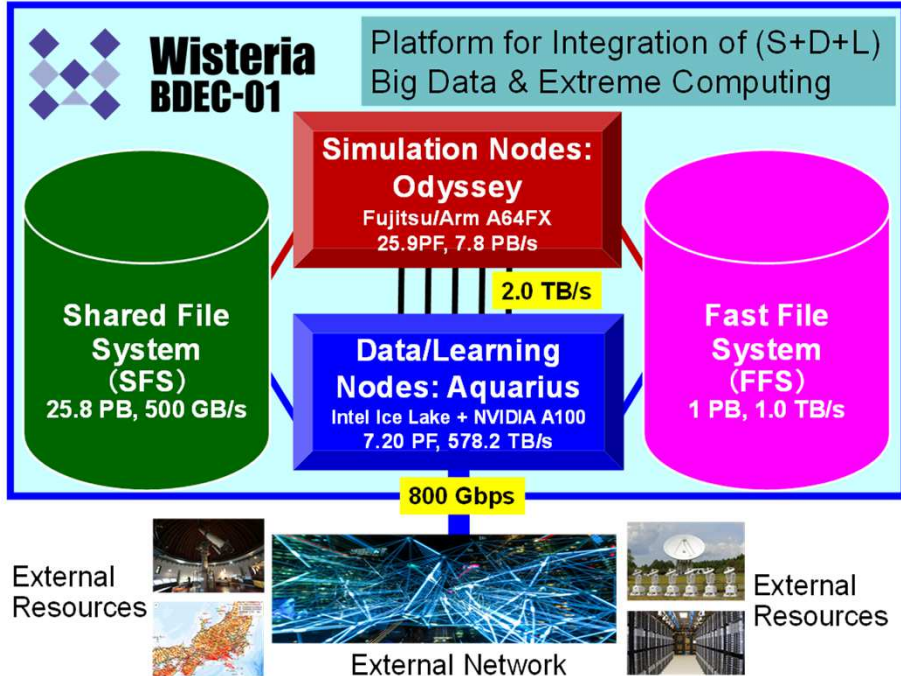
Members (Co-PI's) of h3-Open-BDEC Project

Computer Science, Computational Science, Numerical Algorithms,
Data Science, Machine Learning

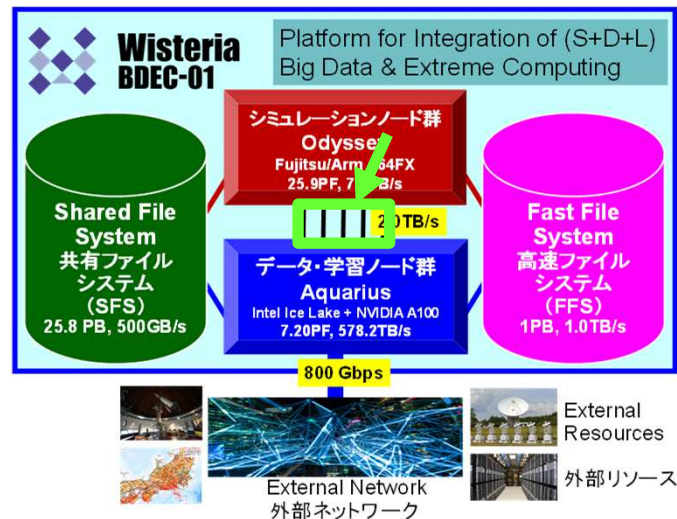
- Kengo Nakajima (ITC/U.Tokyo, RIKEN), Leading-PI
- Takeshi Iwashita (Hokkaido U), Co-PI, Algorithms
- Hisashi Yashiro (NIES), Co-PI, Coupling, Utility
- Hiromichi Nagao (ERI/U.Tokyo), Co-PI, Data Assimilation
- Takashi Shimokawabe (ITC/U.Tokyo), Co-PI, ML/hDDA
- Takeshi Ogita (Waseda U.), Co-PI, Accuracy Verification
- Takahiro Katagiri (Nagoya U), Co-PI, Appropriate Computing
- Hiroya Matsuba (ITC/U.Tokyo, Hitachi), Co-PI, Container



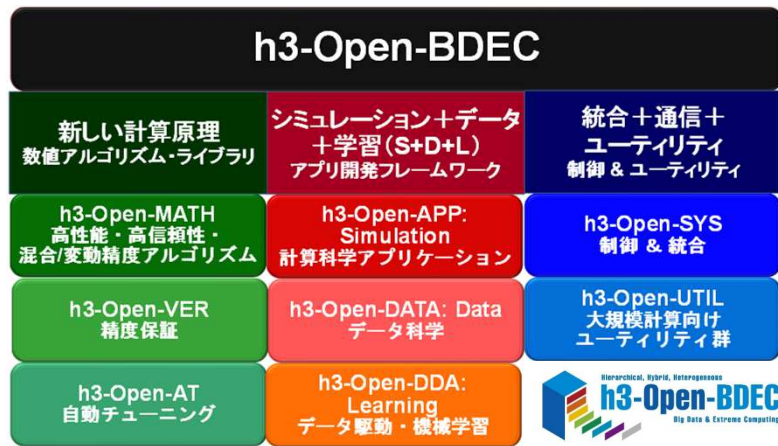
Wisteria/BDEC-01: The First “Really Heterogenous” System in the World



(S+D+L)融合 の実現へ向けて

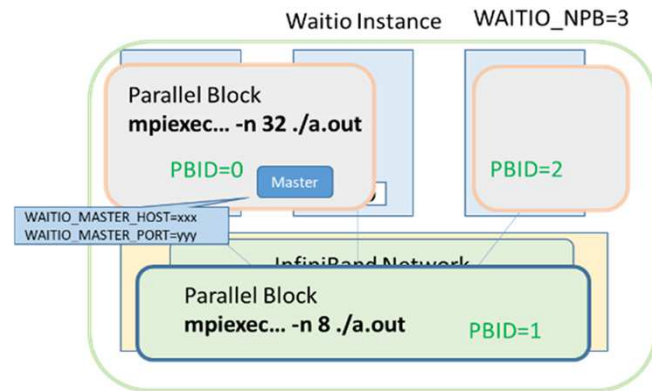


- Odyssey-Aquarius連携
 - MPIによる通信は不可
 - O-Aを跨いでMPIプログラムは動かない
 - Odyssey-Aquarius間にはInfiniband-EDR (2TB/sec)で結合されている
- ソフトウェア開発
 - 高機能カプラー: h3-Open-UTIL/MP
 - O-A間通信: h3-Open-SYS/WaitIO
 - IB-EDR経由 (WaitIO-Socket)
 - 高速ファイルシステム (FFS) 経由連携 (WaitIO-File)
 - MPIライクなインタフェース



API of h3-Open-SYS/WaitIO-Socket PB (Parallel Block): Each Application

WaitIO API	Description
<code>waitio_isend</code>	Non-Blocking Send
<code>waitio_irecv</code>	Non-Blocking Receive
<code>waitio_wait</code>	Termination of <code>waitio_isend/irecv</code>
<code>waitio_init</code>	Initialization of WaitIO
<code>waitio_get_nprocs</code>	Process # for each PB (Parallel Block)
<code>waitio_create_group</code> <code>waitio_create_group_wranks</code>	Creating communication groups among PB's
<code>waitio_group_rank</code>	Rank ID in the Group
<code>waitio_group_size</code>	Size of Each Group
<code>waitio_pb_size</code>	Size of the Entire PB
<code>waitio_pb_rank</code>	Rank ID of the Entire PB



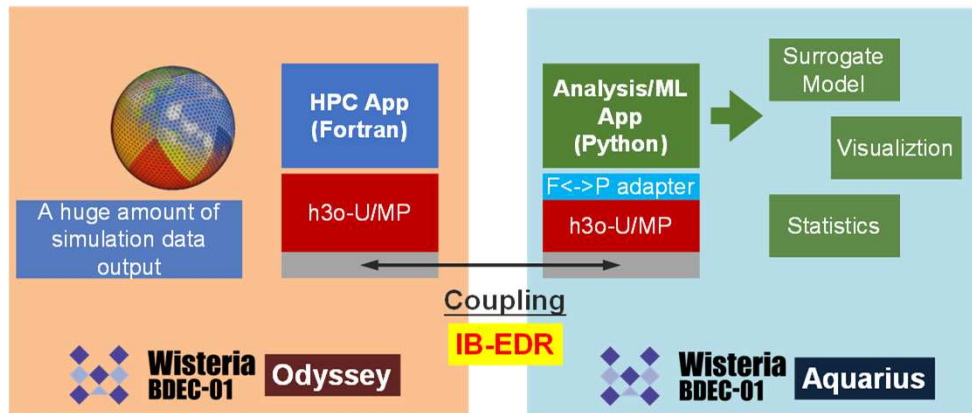
[Sumimoto et al. 2021]

h3-Open-UTIL/MP

Multilevel Coupler/Data Assimilation Integration of (S+D+L)

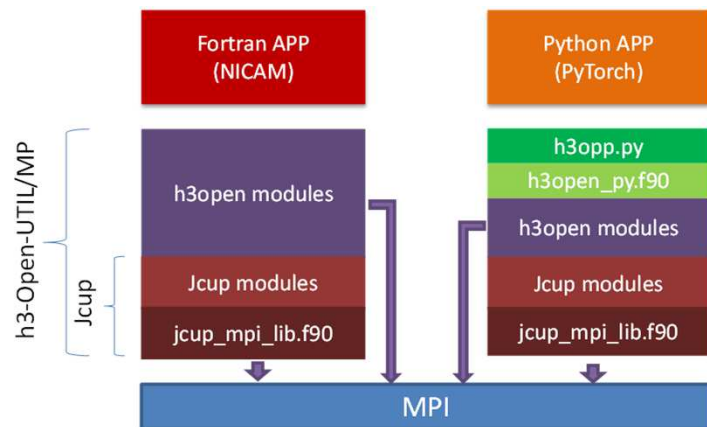


- Extended Version of Multi-Physics Coupler
- Data Assimilation (Multiple Computations: Ensemble)
 - Assimilation of Computations with Different Resolutions
 - Data Assimilation by Coupled Codes
 - e.g. Atmosphere-Ocean
 - Ensemble Coupling
- Coupling of Simulations on Odyssey and AI on Aquarius

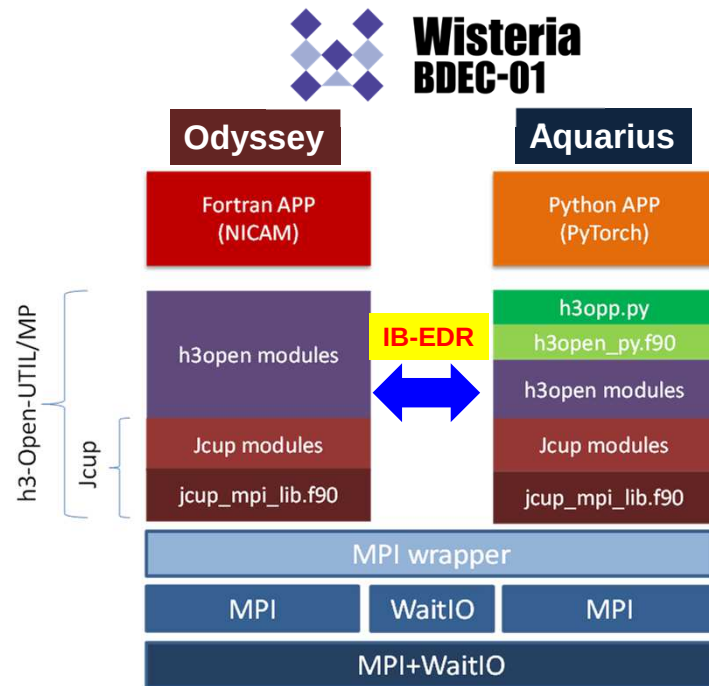


h3-Open-UTIL/MP + h3-Open-SYS/WaitIO-Socket

Available in June 2022

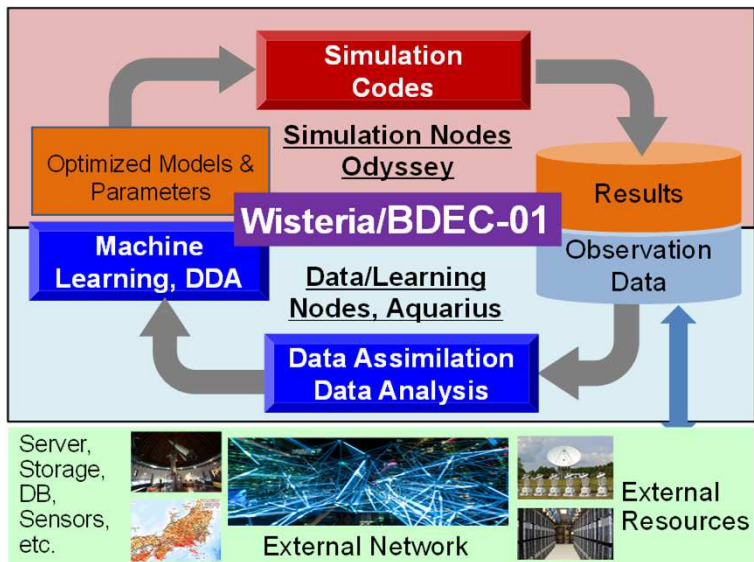


May 2021: MPI Only



June 2022: Coupler + WaitIO

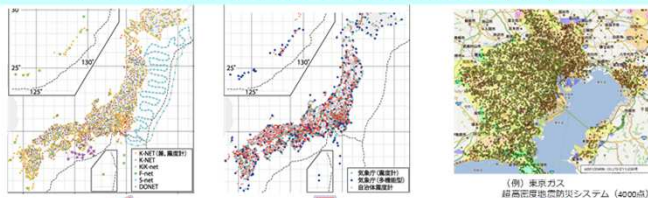
- **Applications on Wisteria/BDEC-01 with h3-Open-BDEC**
 - **Earthquake Simulations**
 - (Global Cloud Simulation+AI) Coupling
 - Ensemble Coupling
 - JHPCN



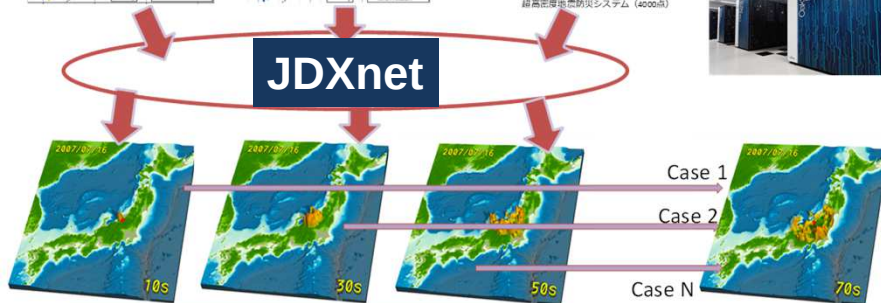
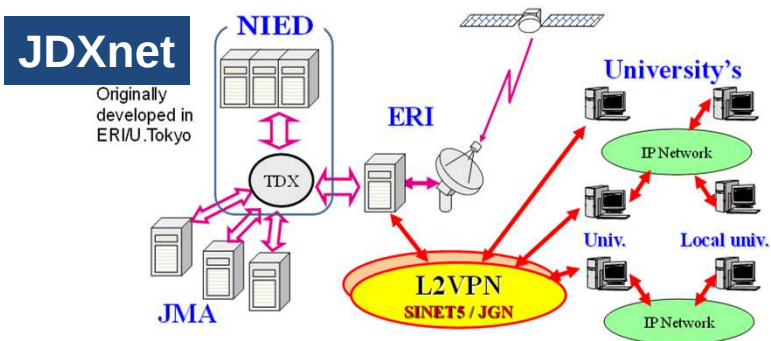
3D Earthquake Simulation with Real-Time Data

Observation/Assimilation Simulation of Strong Motion (Wave Propagation) by 3D FDM

Observation Network for Earthquake: $O(10^5)$ Points



[c/o Furumura]



Real-Time Data/Simulation Assimilation
Real-Time Update of Underground Model

[c/o Prof. T.Furumura
(ERI/U.Tokyo)]

Starting from (A+S: Assim+Sim.) to (Pure S: Pure Simulation)

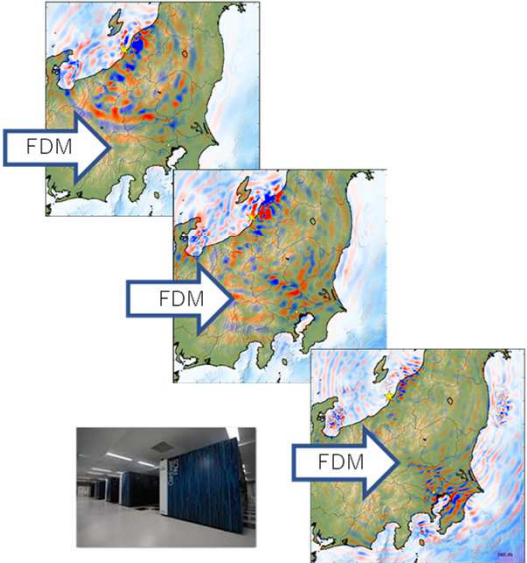
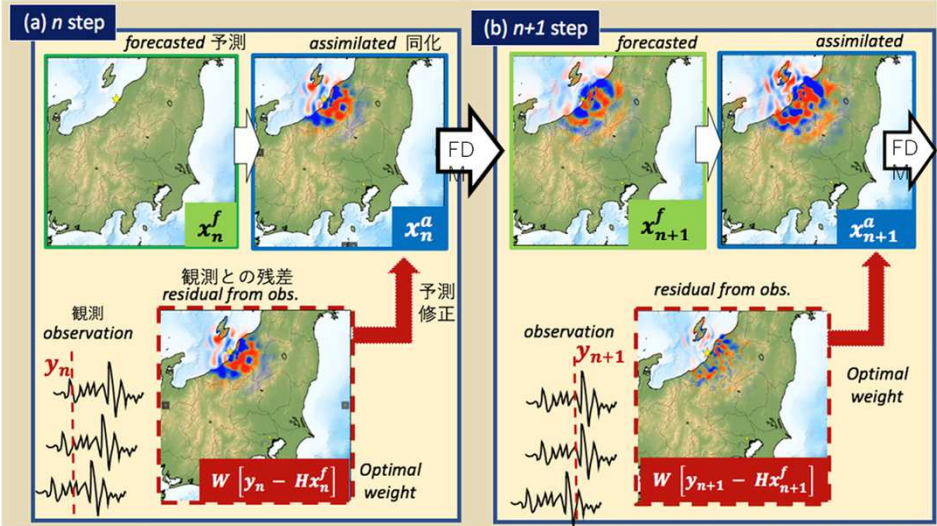
Assim. Comp. Obs. Residual Comp.

$$\mathbf{x}_n^a = \mathbf{x}_n^f + \mathbf{W}(\mathbf{y}_n - \mathbf{H}\mathbf{x}_n^f)$$

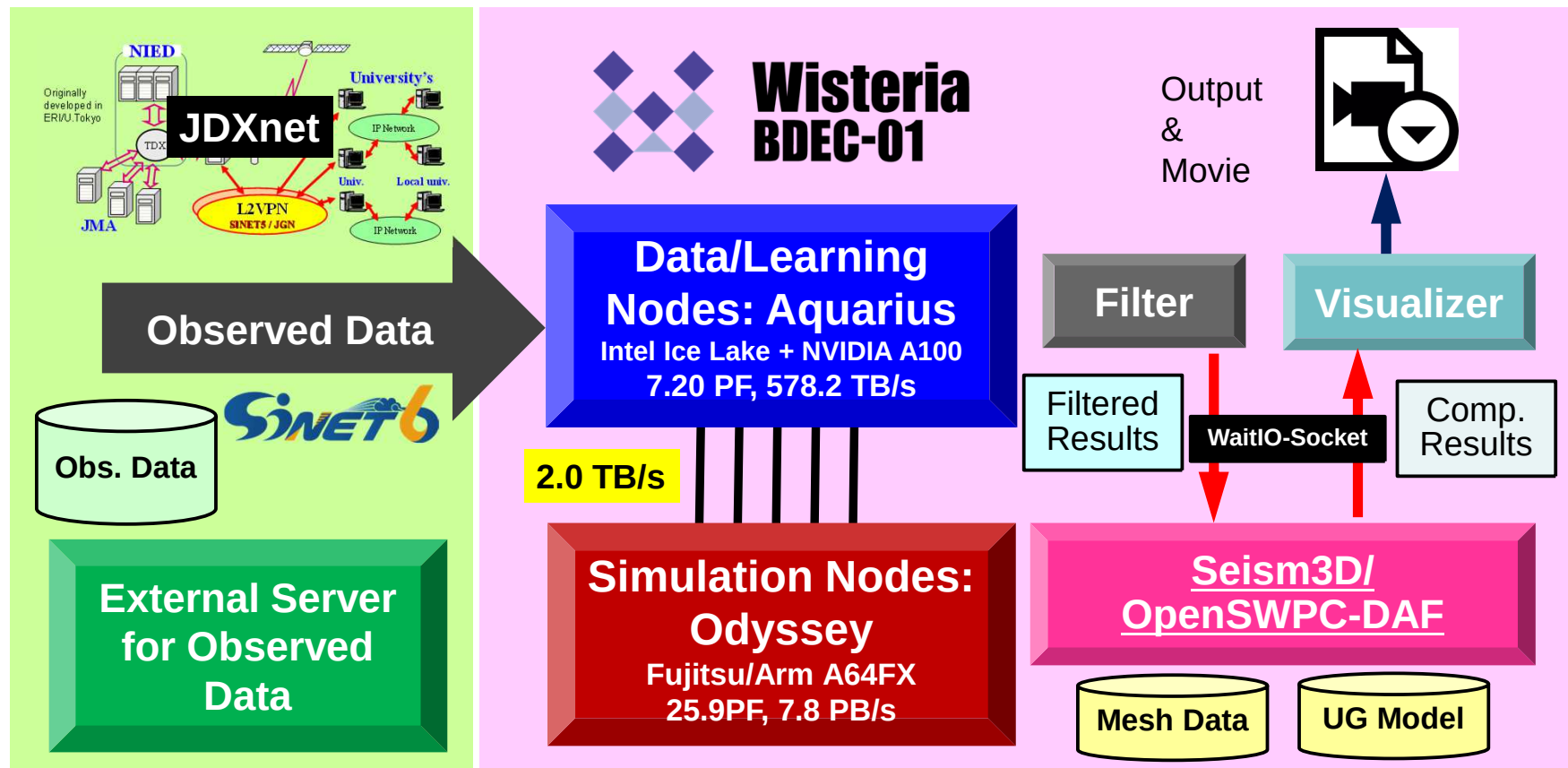
Comp. Assim.

$$\mathbf{x}_{n+1}^f = \mathbf{F}\mathbf{x}_n^a$$

n : Time Step
 $\mathbf{W}$: Weighting Matrix
 $\mathbf{F}$: Wave Propagation simulation



長周期地震動シミュレーション＋観測データ同化



Communications by WaitIO-Socket

[Kasai et al. 2021]

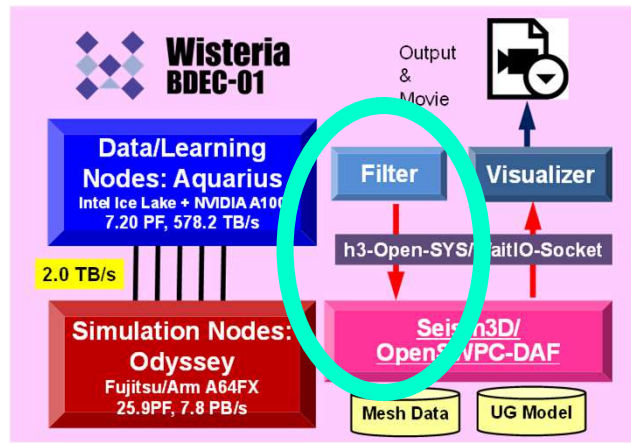
Aquarius: SEND

```
program dmy_filter
<省略: 型宣言等>
call mpi_init (ierr)
call mpi_comm_size (MPI_COMM_WORLD, nprocs, ierr)
call mpi_comm_rank (MPI_COMM_WORLD, myrank, ierr)
call WAITIO_CREATE_UNIVERSE (WAITIO_COMM_UNIVERSE, ierr)

if (myrank==0) then
open(100,file='./obsfile_list.txt', form='formatted', status='old', iostat=ierr)
do i=1,300
  <省略: obsデータ読み込み処理>
  print *, "Send obs data ....."
  call WAITIO_MPI_ISEND (NTMAX1_o, 1, WAITIO_MPI_INTEGER, 2,1, WAITIO_COMM_UNIVERSE, req(1,1), ierr)
  call WAITIO_MPI_ISEND (DT_o, 1, WAITIO_MPI_FLOAT, 2,2, WAITIO_COMM_UNIVERSE, req(1,2), ierr)
  call WAITIO_MPI_ISEND (NST_o, 1, WAITIO_MPI_INTEGER, 2,3, WAITIO_COMM_UNIVERSE, req(1,3), ierr)
  call WAITIO_MPI_ISEND (AT_o, 1, WAITIO_MPI_INTEGER, 2,4, WAITIO_COMM_UNIVERSE, req(1,4), ierr)
  call WAITIO_MPI_ISEND (T0_o, 1, WAITIO_MPI_FLOAT, 2,5, WAITIO_COMM_UNIVERSE, req(1,5), ierr)
  call WAITIO_MPI_ISEND (ISO_X_o, NSMAX, WAITIO_MPI_INTEGER, 2,6, WAITIO_COMM_UNIVERSE, req(1,6), ierr)
  call WAITIO_MPI_ISEND (ISO_Y_o, NSMAX, WAITIO_MPI_INTEGER, 2,7, WAITIO_COMM_UNIVERSE, req(1,7), ierr)
  call WAITIO_MPI_ISEND (ISO_Z_o, NSMAX, WAITIO_MPI_INTEGER, 2,8, WAITIO_COMM_UNIVERSE, req(1,8), ierr)
  call WAITIO_MPI_ISEND (ISTX_o, NST, WAITIO_MPI_INTEGER, 2,9, WAITIO_COMM_UNIVERSE, req(1,9), ierr)
  call WAITIO_MPI_ISEND (ISTY_o, NST, WAITIO_MPI_INTEGER, 2,10, WAITIO_COMM_UNIVERSE, req(1,10), ierr)
  call WAITIO_MPI_ISEND (ISTZ_o, NST, WAITIO_MPI_INTEGER, 2,11, WAITIO_COMM_UNIVERSE, req(1,11), ierr)
  call WAITIO_MPI_ISEND (STC_o, 6*NST, WAITIO_MPI_INTEGER, 2,12, WAITIO_COMM_UNIVERSE, req(1,12), ierr)
  call WAITIO_MPI_ISEND (STC_o, 6*NST, WAITIO_MPI_INTEGER, 2,12, WAITIO_COMM_UNIVERSE, req(1,12), ierr)
  call WAITIO_MPI_ISEND (VxAll_obs, NST*NOBS_LEN, WAITIO_MPI_FLOAT, 2,13, WAITIO_COMM_UNIVERSE, req(1,13), ierr)
  call WAITIO_MPI_ISEND (VyAll_obs, NST*NOBS_LEN, WAITIO_MPI_FLOAT, 2,14, WAITIO_COMM_UNIVERSE, req(1,14), ierr)
  call WAITIO_MPI_ISEND (VzAll_obs, NST*NOBS_LEN, WAITIO_MPI_FLOAT, 2,15, WAITIO_COMM_UNIVERSE, req(1,15), ierr)
  call WAITIO_MPI_WAITALL (15, req, status, ierr)
  call sleep(1)
enddo
close (100)
endif
call WAITIO_FINALIZE (ierr)
call mpi_finalize (ierr)
end
```

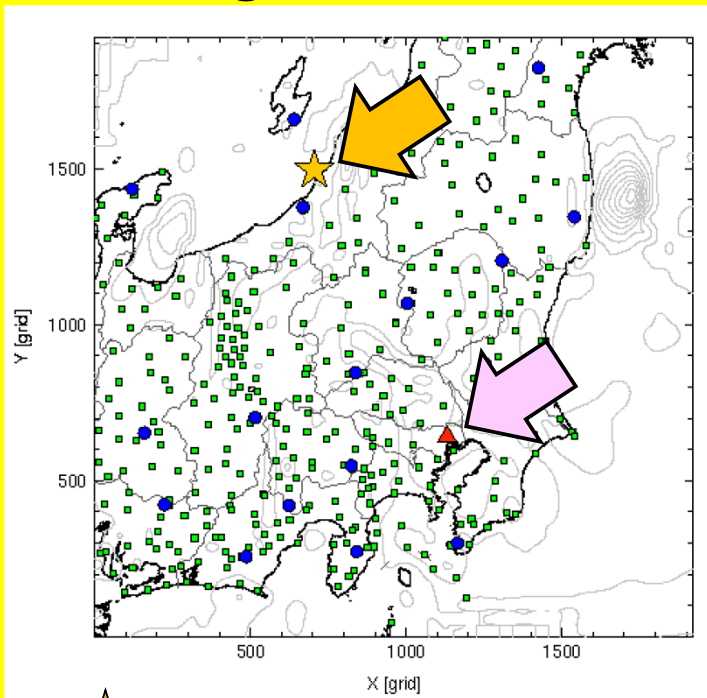
Odyssey: RECV

```
call WAITIO_MPI_RECV (NTMAX1_o, 1, WAITIO_MPI_INTEGER, 0,1, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (DT_o, 1, WAITIO_MPI_FLOAT, 0,2, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (NST_o, 1, WAITIO_MPI_INTEGER, 0,3, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (AT_o, 1, WAITIO_MPI_FLOAT, 0,4, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (T0_o, 1, WAITIO_MPI_FLOAT, 0,5, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (ISO_X_o, NSMAX, WAITIO_MPI_INTEGER, 0,6, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (ISO_Y_o, NSMAX, WAITIO_MPI_INTEGER, 0,7, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (ISO_Z_o, NSMAX, WAITIO_MPI_INTEGER, 0,8, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (ISTX_o, NST, WAITIO_MPI_INTEGER, 0,9, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (ISTY_o, NST, WAITIO_MPI_INTEGER, 0,10, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (ISTZ_o, NST, WAITIO_MPI_INTEGER, 0,11, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (STC_o, 6*NST, WAITIO_MPI_INTEGER, 0,12, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (STC_o, 6*NST, WAITIO_MPI_INTEGER, 0,12, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (VxAll_obs, NST*NOBS_LEN, WAITIO_MPI_FLOAT, 0,13, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (VyAll_obs, NST*NOBS_LEN, WAITIO_MPI_FLOAT, 0,14, WAITIO_COMM_UNIVERSE,...)
call WAITIO_MPI_RECV (VzAll_obs, NST*NOBS_LEN, WAITIO_MPI_FLOAT, 0,15, WAITIO_COMM_UNIVERSE,...)
```



Off Niigata 2007 Mw6.6 Earthquake

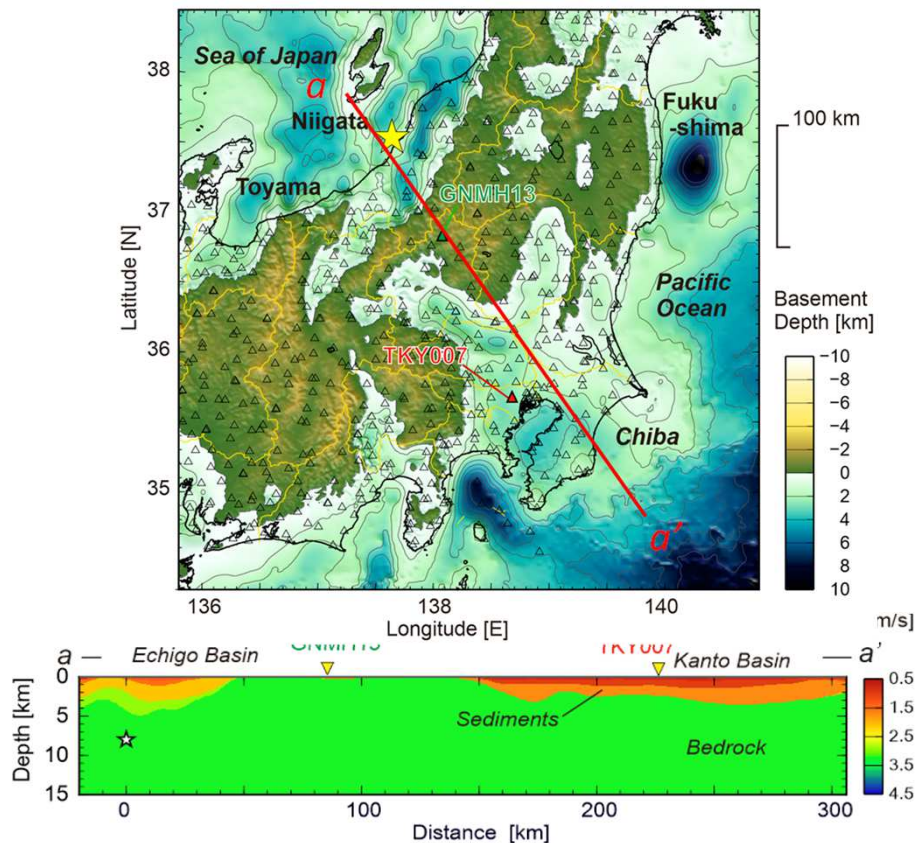
[c/o Prof. T. Furumura,
ERI/U.Tokyo]



★ Epicenter

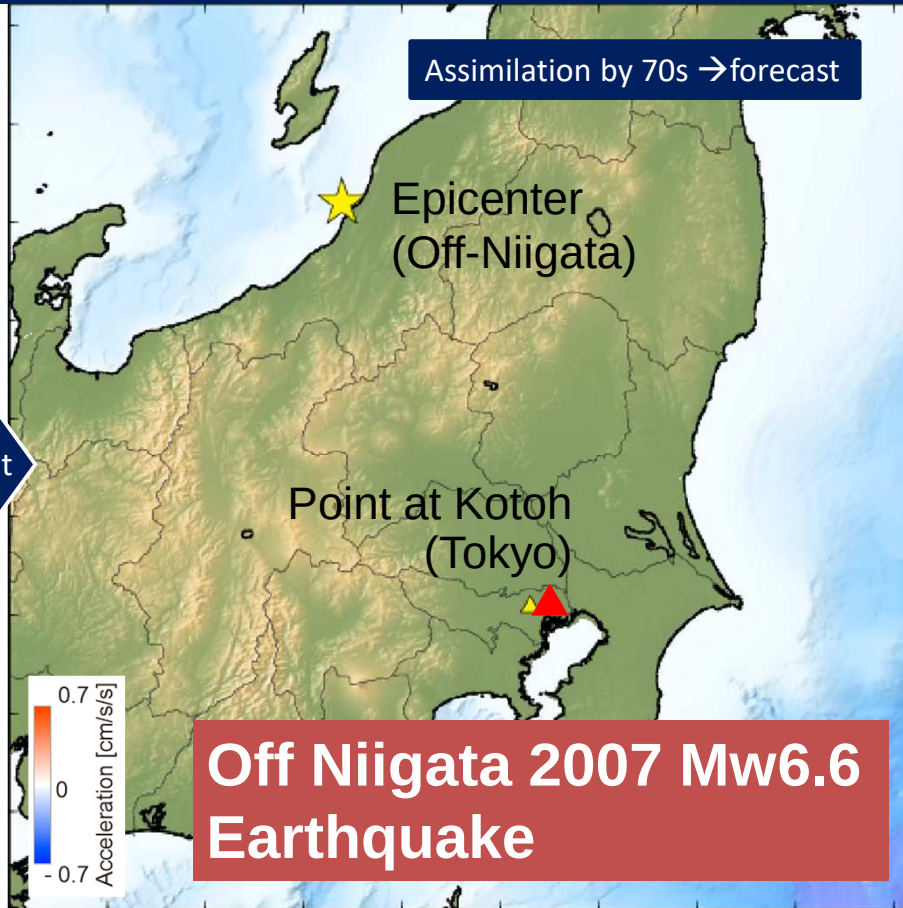
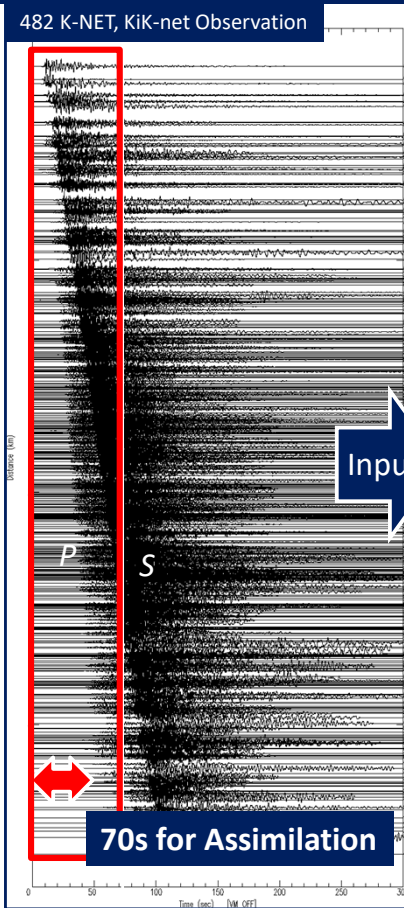
■ Hi-net (Short Period) 349 pts

● F-net (Broadband) 18 pts

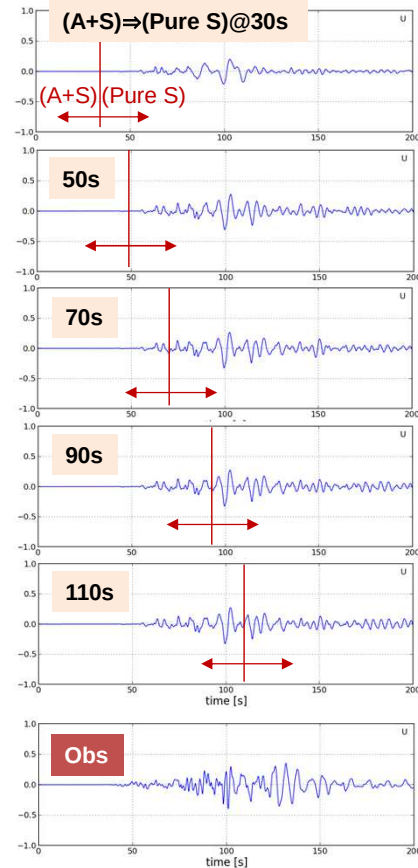


Data Assimilation + Pure Simulation/Forecast

482 K-NET, KiK-net Observation



Results at Kotoh ▲ (N.KOTH)
N 35° 37.0'
E 139° 46.9'



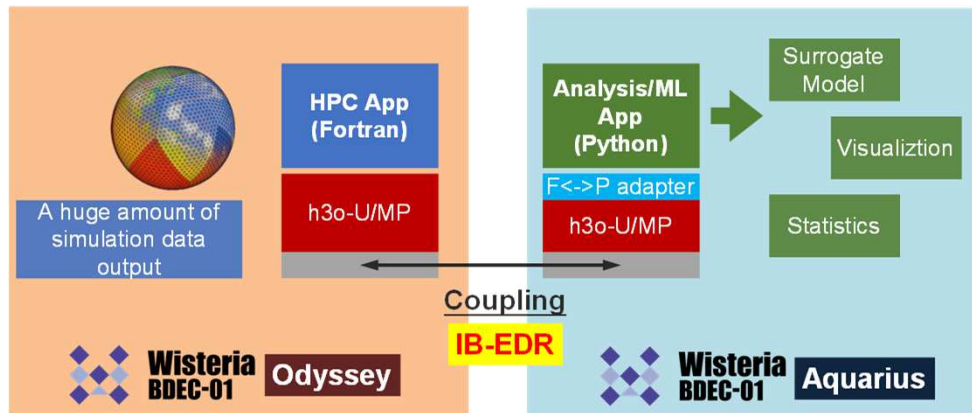
- **Applications on Wisteria/BDEC-01 with h3-Open-BDEC**
 - Earthquake Simulations
 - **(Global Cloud Simulation+AI) Coupling**
 - Ensemble Coupling
 - JHPCN

h3-Open-UTIL/MP

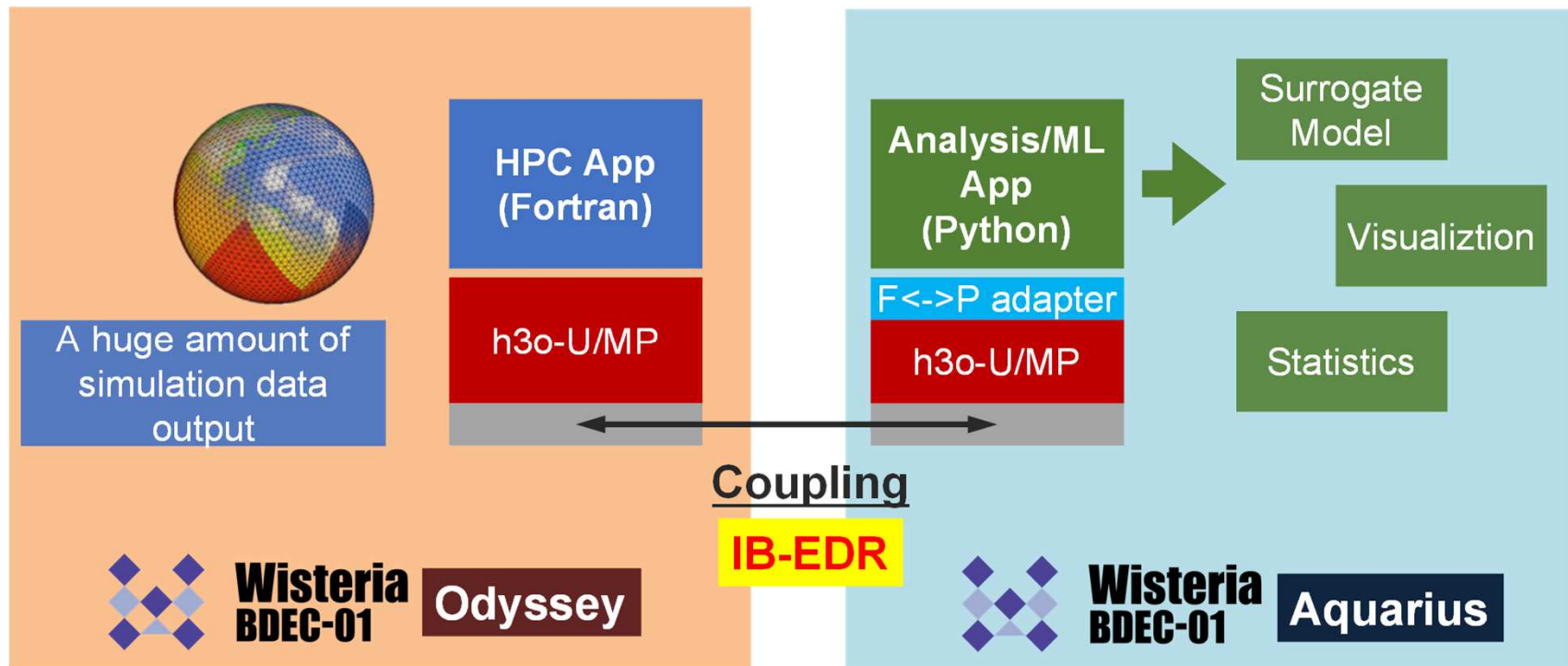
Multilevel Coupler/Data Assimilation Integration of (S+D+L)



- Extended Version of Multi-Physics Coupler
- Data Assimilation (Multiple Computations: Ensemble)
 - Assimilation of Computations with Different Resolutions
 - Data Assimilation by Coupled Codes
 - e.g. Atmosphere-Ocean
 - Ensemble Coupling
- Coupling of Simulations on Odyssey and AI on Aquarius



h3-Open-UTIL/MP (h3o-U/MP) Extended Multiphysics Coupler

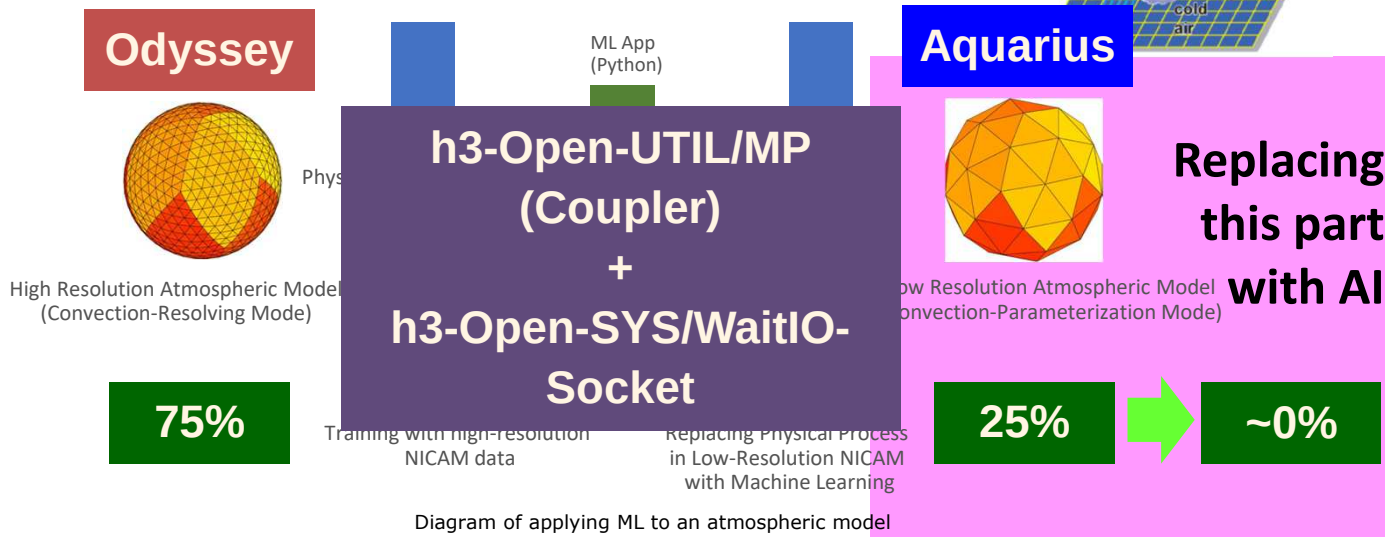
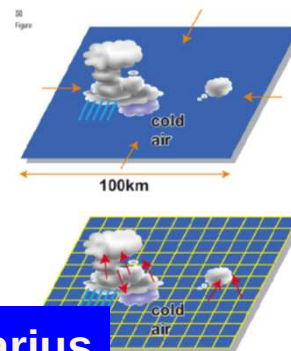


Atmosphere-ML Coupling

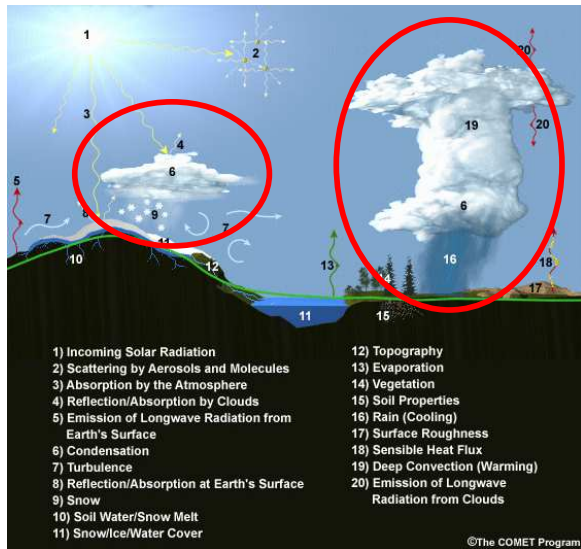
[Yashiro (NIES), Arakawa (ClimTech/U.Tokyo)]

- Motivation of this experiment

- Two types of Atmospheric models: Cloud resolving VS Cloud parameterizing
- Cloud resolving model is difficult to use for climate simulation
- Parameterized model has many assumptions
- Replacing low-resolution cloud processes calculation with ML!



Atmosphere-ML Coupling



- Model component emulation (surrogation)

- The emulation target in this study is cloud microphysical processes (phase changes, collision, coagulation, and precipitation)
- Atmospheric pressure, temperature, and vertical distribution of water will change between before and after computing the cloud microphysical processes

- Atmospheric model and ML Library

- NICAM (global non-hydrostatic model with icosahedral grid) + Pytorch (three layers MLP)

- Methodology

- ML is trained to reproduce output variable from input variables of cloud microphysical subroutine

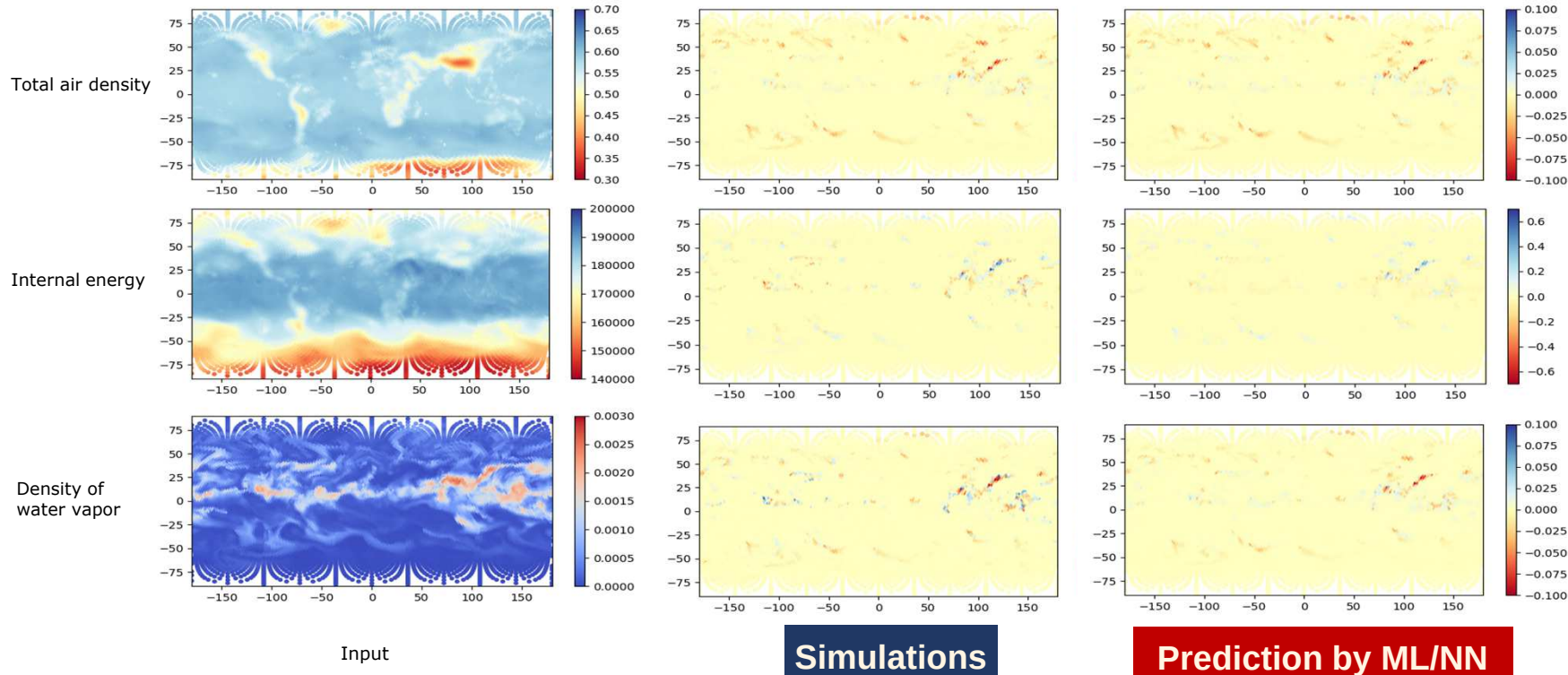
- Training data

- Input : total air density (ρ), internal energy (e), density of water vapor (ρ_q)
- Output : tendencies of input variables computed within the cloud physics subroutine

$$\frac{\Delta \rho}{\Delta T} \quad \frac{\Delta e}{\Delta T} \quad \frac{\Delta \rho_q}{\Delta T}$$

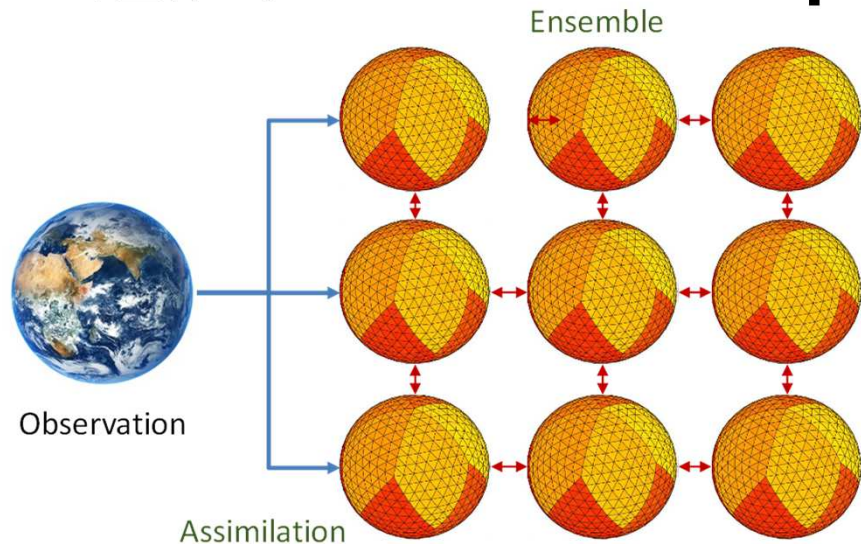
Test calculation

- Compute output variables from input variables and PyTorch
 - The rough distribution of all variables is well reproduced
 - The reproduction of extreme values is no good



- **Applications on Wisteria/BDEC-01 with h3-Open-BDEC**
 - Earthquake Simulations
 - (Global Cloud Simulation+AI) Coupling
 - **Ensemble Coupling [Yashiro, Arakwa]**
 - JHPCN

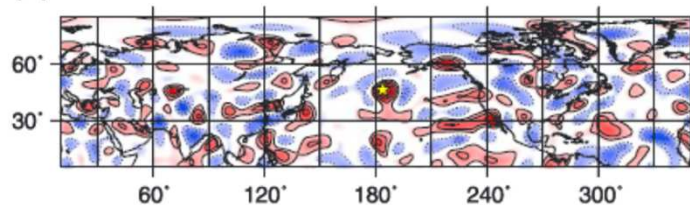
アンサンブル連成 (Ensemble Coupling) (1/3)



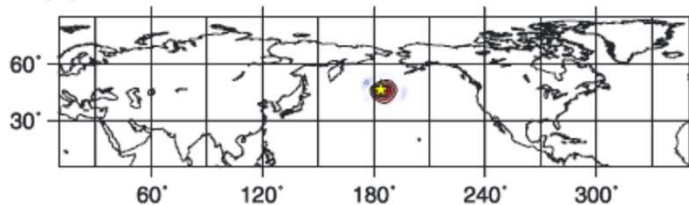
- 全地球規模の気象・気候シミュレーションでは、パラメータスタディ(アンサンブル計算)とデータ同化を組み合わせた、アンサンブルデータ同化(Ensemble-based Data Assimilation)が広く使用されている。
 - $O(10^3)$ ケース程度のアンサンブル計算を実施すれば、高精度な予測が可能であるが、高解像度モデルを使用したアンサンブルデータ同化(高解像アンサンブル)では、計算機リソース、計算時間を考慮すると $O(10^2)$ ケース程度が現実的である[Miyoshi 2014]

アンサンブルデータ同化 [Miyoshi et al. 2014]

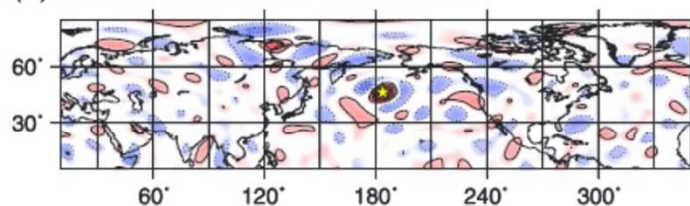
(a) 20 members w/o localization



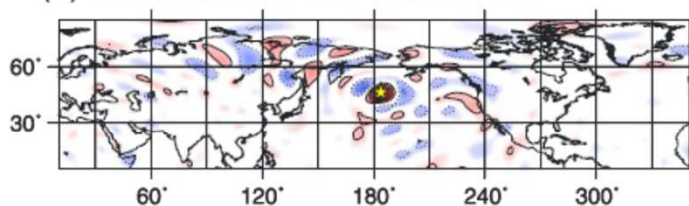
(b) 20 members w/ 700-km localization



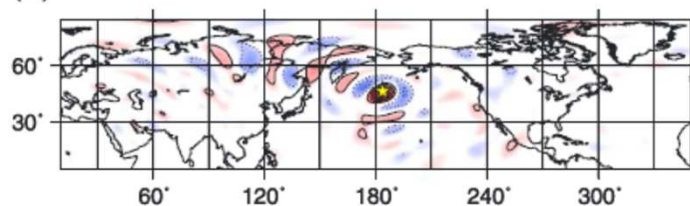
(c) 80 members w/o localization



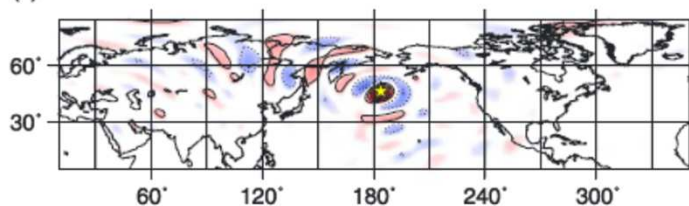
(d) 320 members w/o localization



(e) 1280 members w/o localization

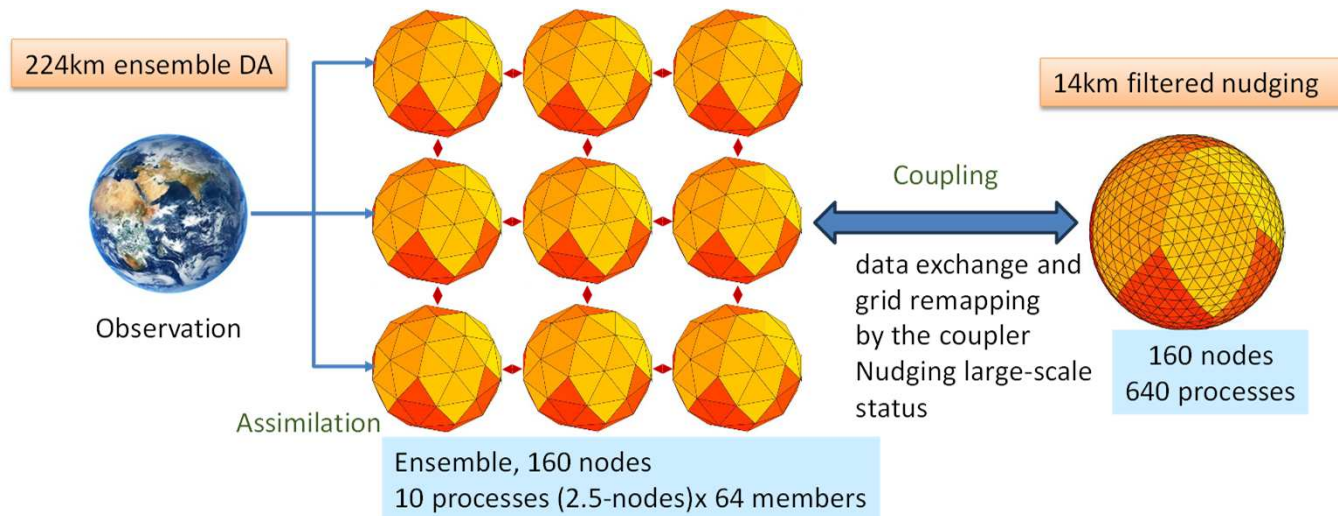


(f) 10240 members w/o localization



アンサンブル連成 (Ensemble Coupling) (2/3)

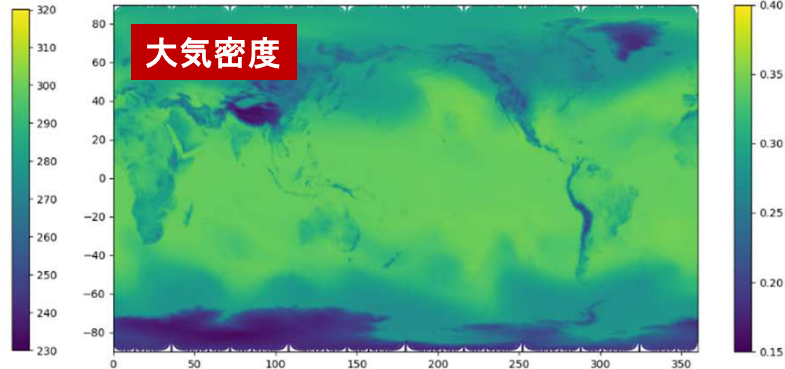
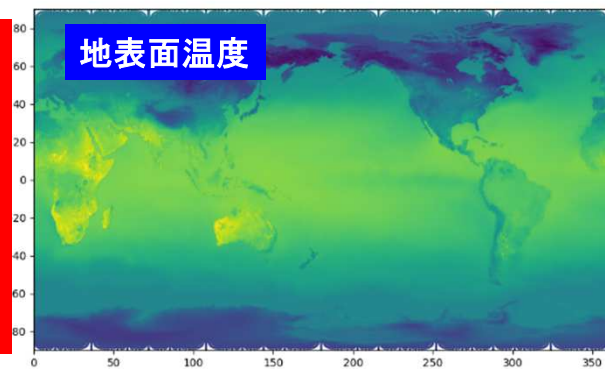
- アンサンブル連成 (Ensemble Coupling) はアンサンブル計算と連成計算を組み合わせたものである。
 - この機能を用いると例えば大気モデルアンサンブルと海洋モデルアンサンブルを連成させた大気海洋アンサンブル連成が可能となり高い再現性を得ることが期待できる。
 - 本研究では、低解像度大気モデルアンサンブルと高解像度大気モデルの連成を実施



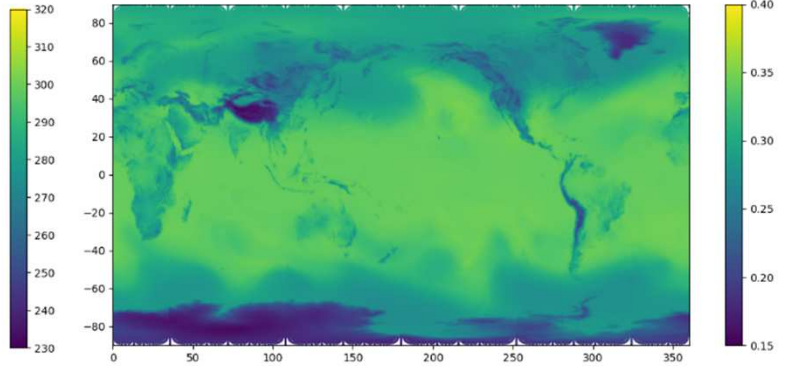
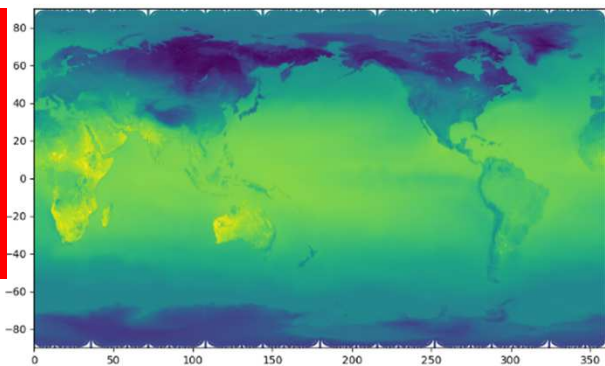
アンサンブル連成(Ensemble Coupling) (3/3)

- 低解像度アンサンブル・高解像度連成により, 高解像度大気モデルのアンサンブル計算に比して計算リソースを大幅に削減することが可能となる。
- さらに, 大域的な場の再現性に優れた低解像度モデルと, 詳細な場の再現性に優れた高解像度モデルを連成することにより, 高解像度モデル単体のアンサンブル計算に比してより高い再現性を得ることが可能となる。
- 大気モデルNICAMによる検証
 - ✓ 低解像度側: 水平解像度約220kmのモデルを「64」アンサンブル
 - ✓ 高解像度側: 水平解像度約14kmの単体モデル
 - ✓ Wisteria/BDEC-01のOdysseyノード群を用いモデル時間9時間の積分で用いたリソースは19.3ノード時間であった(FP32)
 - この結果は高解像度モデルを64アンサンブルで9時間積分した場合の使用リソース2,240ノード時間(FP64)に対して112倍の効率化を達成したことを意味する。
 - 1,000アンサンブル相当のアンサンブル連成を64アンサンブルよりも短い時間(10分の1程度)で実施可能

14kmメッシュ
詳細モデルに
よるデータ同化
(従来型)
2,240 NH



アンサンブル
連成(224km+
14km)
19.3 NH

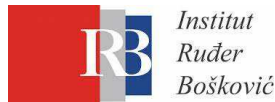


- 9時間積分ではアンサンブル計算と単体実行でほとんど差異はなく、アンサンブル計算でも妥当な結果が得られている(3日積分でも確認⇒HPC Asia 2025に採択)
- 詳細な再現性の検証については気候場の比較が必要であり少なくとも1ヶ月程度の積分と気象学的な解析が要求されるため今後の課題である。

- **Applications on Wisteria/BDEC-01 with h3-Open-BDEC**
 - Earthquake Simulations
 - (Global Cloud Simulation+AI) Coupling
 - Ensemble Coupling [Yashiro, Arakwa]
 - **JHPCN**

FY.2023-2025, JHPCN Project

Innovative Computational Science by Integration of
Simulation/Data/Learning on Heterogeneous Supercomputers



- ✓ Jülich Supercomputing Centre (JSC)
- ✓ Rudjer Boskovic Institute, Centre for Informatics and Computing, Croatia
- ✓ Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
- ✓ French Atomic Energy Commission (CEA)
- ✓ Bergische Universität Wuppertal (BUW)
- ✓ Karlsruher Institut für Technologie (KIT)

Organization: FY.2024

Proposal must be renewed and reviewed every year

- (Leading-PI) Kengo Nakajima (ITC/U.Tokyo)   東京大学
THE UNIVERSITY OF TOKYO
- (Co-PI) Takashi Furumura (ERI/U.Tokyo)   東京大学
THE UNIVERSITY OF TOKYO
- (Co-PI) France Boillod-Cerneux (CEA)  
- (Co-PI) Edoardo Di Napoli (JSC)   

Target Applications



– Terrestrial Systems Modeling Platform (TSMP)

- Coupling: Groundwater Flow & Atmosphere
- <https://www.terrsysmp.org/>

– Chebyshev Accelerated Subspace Eigensolver (ChASE)

- Quantum Chemistry, Heterogeneous Environment
- <https://github.com/ChASE-library>

– Brain Aneurysm Simulations

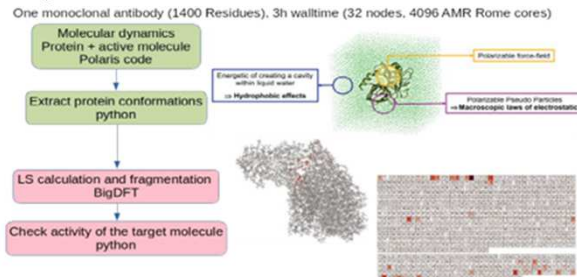
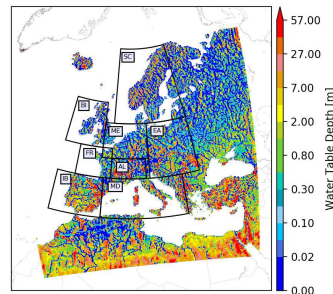
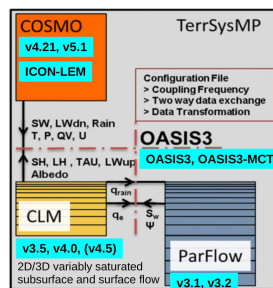
- Multiscale, Multiphysics
- CFD Codes (m-AIA) at JSC
- <https://www.hpccoe.eu/2021/06/04/m-aia/>



– Selection of inhibitors of the SARS-CoV-2 Main Protease

- BigDFT + Polaris/GENESIS

– Earthquake Simulation (PSD), ML with Causality



ALGORITHM 1: The ChASE algorithm: SI plus our original contributions

Require: Hermitian matrix A , number of desired eigenpairs nev , threshold tolerance for residuals tol , initial polynomial degree deg , search space increment $step$, and $opton$ flags, vector matrix $\tilde{V}$ is $\{\tilde{V}_1 \dots \tilde{V}_{nev}\}$ and estimates μ_1 and μ_{nev}

Ensure: new external eigenpairs $(\tilde{\lambda}, \tilde{V})$, with $\tilde{\lambda} = [\tilde{\lambda}_1 \dots \tilde{\lambda}_{nev}]$ and $\tilde{V}$ is $\{\tilde{V}_1 \dots \tilde{V}_{nev}\}$ and their residuals $Res(\tilde{\lambda}, \tilde{V}) \dots Res(\tilde{\lambda}_{nev}, \tilde{V}_{nev})$

```

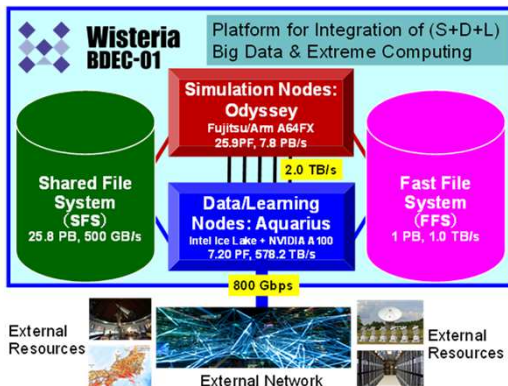
1:  $m := nev$ ;  $maxdeg := deg$                                 ▷ INITIAL CONSTANT DEGREE
2:  $(happ, P) := PREPARE(\tilde{V}) := LANGOSQ(A, approx)$           ▷ BUILD AND OPTIMIZE INPUT
3: while  $size(\tilde{V}) < nev$  do
4:    $\tilde{V} := FILTER(A, happ, P; PREPARE(\tilde{V}, P, m, opton))$       ▷ USE ARRAY OF DEGREES
5:    $\tilde{Q} := ORTHOGONALIZE(\tilde{V}, \tilde{V})$                              ▷ QR FACTORIZATION
6:    $\tilde{Q} := [\tilde{Q}_{(size(\tilde{V})+1)} \dots \tilde{Q}_{(nev)}]$ 
7:    $(\tilde{V}, \tilde{\lambda}) := BACKLICK-RTQA(\tilde{Q}, \tilde{Q})$ 
8:   Compute the residuals  $Res(\tilde{V}, \tilde{\lambda})$ 
9:    $(\tilde{V}, \tilde{\lambda}, \tilde{r}) := INFLATION \& LOCKING(\tilde{V}, \tilde{\lambda}, Res(\tilde{V}, \tilde{\lambda}), \tilde{r})$ 
10:   $\mu_1 := min(|\tilde{\lambda}_1|, |\tilde{\lambda}_m|)$ ;  $\mu_{nev} := max(|\tilde{\lambda}_1|, |\tilde{\lambda}_m|)$ 
11:   $c := \frac{happ(\mu_{nev})}{happ(\mu_1)}$ ;  $\epsilon := \frac{happ(\mu_{nev})}{happ(\mu_1)}$ 
12:  for  $a = 1 \rightarrow size(\tilde{V})$  do
13:     $\mu_a := DEGREESUBST. Res(\tilde{V}_{(a)}, \tilde{\lambda}_a, c, \epsilon)$           ▷ COMPUTE PHENOMENAL DEGREE
14:  end for
15:  Sort  $Res(\tilde{V}, \tilde{\lambda}, \tilde{r}, \tilde{\mu})$  according to  $m$ 
16: end while
  
```

Collaborations related to Heterogeneous Computing (U.Tokyo-JSC)

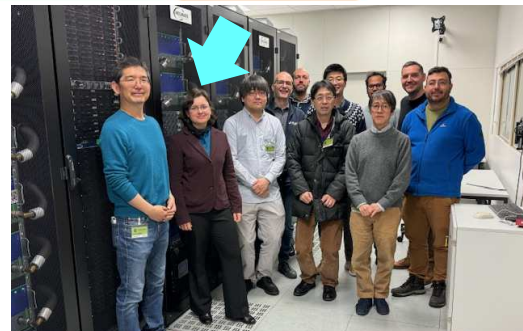
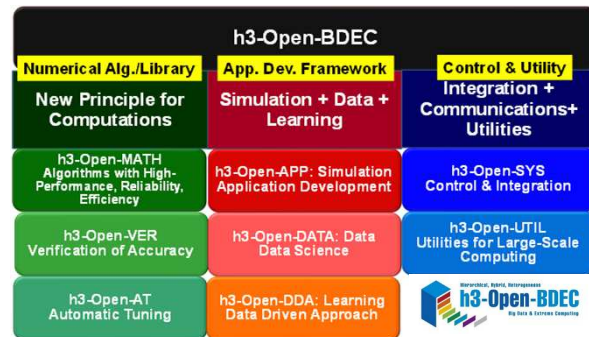
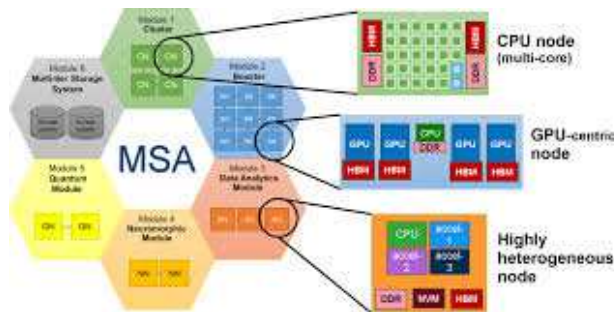


SC22以降, SC-XY, ISC-HPCでBoF共催

Wisteria/BDEC-01
h3-Open-BDEC
U.Tokyo
 K. Nakajima et al.



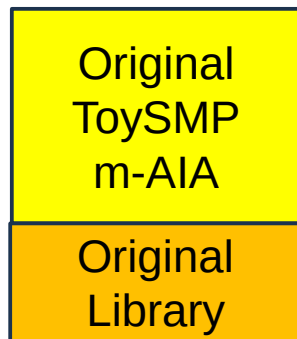
Modular Supercomputing Architecture (MSA), JSC
 E. Suarez et al.



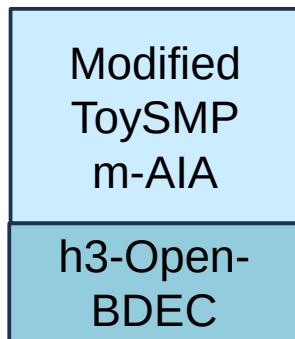
合宿 Mar.13-15, JSC
 taken by KN

Comparisons for various data-sets

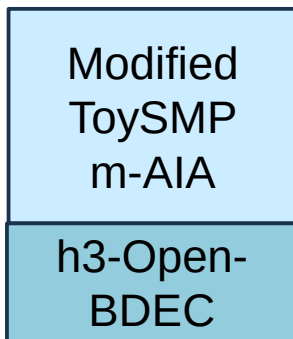
DEEP: Xeon + Xeon/A100 + InfiniBand



DEEP



DEEP



**Wisteria/
BDEC-01**



Caviedes-Voullieme, D., Koh, S.-R., Poll, S., Suarez, E., Arakawa, T., Nakajima, K., Sumimoto, S., Portability of Multiphysics Applications on Heterogeneous Modular Supercomputers, Proceedings of Workshop on Advancements of Global Challenges Applications in conjunction with PPAM24, 2024

Target Applications



- Terrestrial Systems Modeling Platform (TSMP)

- Coupling: Groundwater Flow & Atmosphere

- <https://www.terrsysmp.org/>

- Chebyshev Accelerated Subspace Eigensolver (ChASE)

- Quantum Chemistry, Heterogeneous Environment

- <https://github.com/ChASE-library>

- Brain Aneurysm Simulations

- Multiscale, Multiphysics

- CFD Codes (m-AIA) at JSC

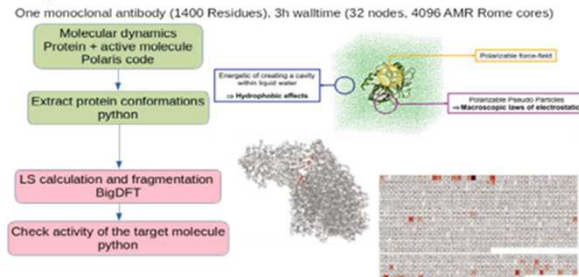
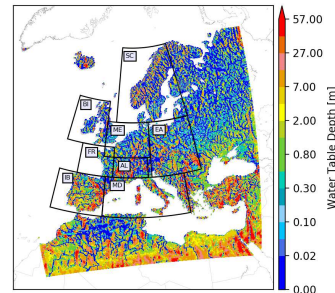
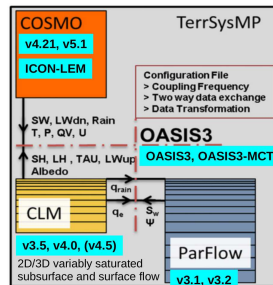
- <https://www.hpccoe.eu/2021/06/04/m-aia/>



- Selection of inhibitors of the SARS-CoV-2 Main Protease

- BigDFT + Polaris/GENESIS

- Earthquake Simulation (PSD), ML with Causality



ALGORITHM 1: The ChASE algorithm: SI plus our original contributions

Require: Hermitian matrix A , number of desired eigenpairs nev , threshold tolerance for residuals tol, initial polynomial degree deg, search space increment nes , `approx` and `optm` flags, vector matrix $\tilde{V}$ $\in [p_1 \dots p_{\text{columns}}]$ and estimates μ_1 and μ_{columns}

Ensure: new external eigenpairs $(\tilde{\Lambda}, \tilde{V})$, with $\tilde{\Lambda} = [\lambda_1 \dots \lambda_{nev}]$ and $\tilde{V} \in [p_1 \dots p_{\text{columns}}]$ and their residuals $[\text{Res}(\tilde{\Lambda}, \tilde{V}) \dots \text{Res}(\tilde{\Lambda}_{nev}, \tilde{V}_{nev})]$

```

1:  $m_{\text{columns}} \leftarrow \text{deg}$                                 > INITIAL CONSTANT DEGREE
2:  $(\text{flag}, p_1, p_{\text{columns}}, \tilde{V}) \leftarrow \text{LANGOS}(A, \text{approx})$     > BOUND AND OPTIMAL INPUT

3: while size( $\tilde{V}$ ) < nev do
4:    $\tilde{V} \leftarrow \text{FILTER}(A, \text{flag}, p_1, p_{\text{columns}}, \tilde{V}, m, \text{optm})$     > VIES ARRAY OF DEGREES
5:    $\tilde{Q} \leftarrow \text{ORTHOGONALIZE}(\tilde{V}, \tilde{V})$                                 > QR FACTORIZATION
6:    $\tilde{Q} \leftarrow [\tilde{Q}_{\text{columns}(\tilde{V})+1} \dots \tilde{Q}_{m_{\text{columns}}}]$                 > REDUCE TO ACTIVE SUBSPACE
7:    $(\tilde{V}, \tilde{\Lambda}) \leftarrow \text{EIGSOLVER-ATTA}(A, \tilde{Q})$ 
8:   Compute the residuals  $\text{Res}(\tilde{V}, \tilde{\Lambda})$ 
9:    $(\tilde{V}, \tilde{\Lambda}, \tilde{T}) \leftarrow \text{INFLATION OF LOCKING}(\tilde{V}, \tilde{\Lambda}, \text{Res}(\tilde{V}, \tilde{\Lambda}), \tilde{T})$ 
10:   $\mu_1 \leftarrow \min(\tilde{\Lambda}, \tilde{\Lambda}_0)$ ;  $\mu_{\text{columns}} \leftarrow \max(\tilde{\Lambda}, \tilde{\Lambda}_0)$ 
11:   $c \leftarrow \frac{\mu_1 - \mu_{\text{columns}}}{2}$ ;  $\epsilon \leftarrow \frac{\mu_1 - \mu_{\text{columns}}}{2}$ 
12:  for  $a = 1 \rightarrow \text{size}(\tilde{V})$  do
13:     $\mu_a \leftarrow \text{DEGREES}(\text{Res}(\tilde{V}_{\text{columns}}, \tilde{\Lambda}_a, c, \epsilon))$     > COMPUTE PHENOMENAL DEGREE
14:  end for
15:  Sort  $\text{Res}(\tilde{V}, \tilde{\Lambda}, \tilde{T}, \tilde{\Lambda}, m)$  according to  $m$ 
16: end while
  
```

Big-DFT with GENESIS for SARS-CoV-2 Main Protease (CEA, RIKEN, U.Tokyo) (1/2)



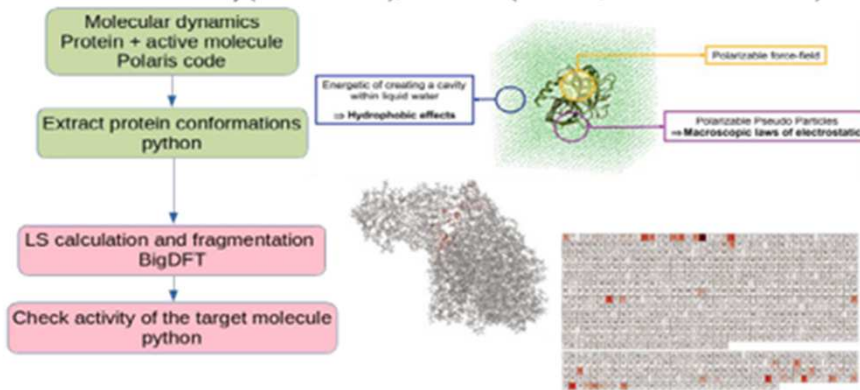
東京大学
THE UNIVERSITY OF TOKYO

- Developing medicines for viruses like SARS-CoV-2 faces challenges, including drug resistance (SARS-CoV-2: Virus, COVID-19: Infection)
 - Understanding and predicting drug resistance involves modeling structural changes from point mutations, utilizing long trajectories from classical molecular dynamics (MD/MM).
 - Mechanistic insight into mutation effects can benefit from quantum mechanical (QM) modeling.
- QM-MM Workflow:**
 - MM creates input (long trajectories) for QM**
 - Initial Idea: “Polaris” for MM developed by CEA
 - Now it is by “GENESIS” by RIKEN

**GENESIS (RIKEN)
Polaris (CEA)
(MM/MD)**

BigDFT (QM)

One monoclonal antibody (1400 Residues), 3h walltime (32 nodes, 4096 AMR Rome cores)



Members

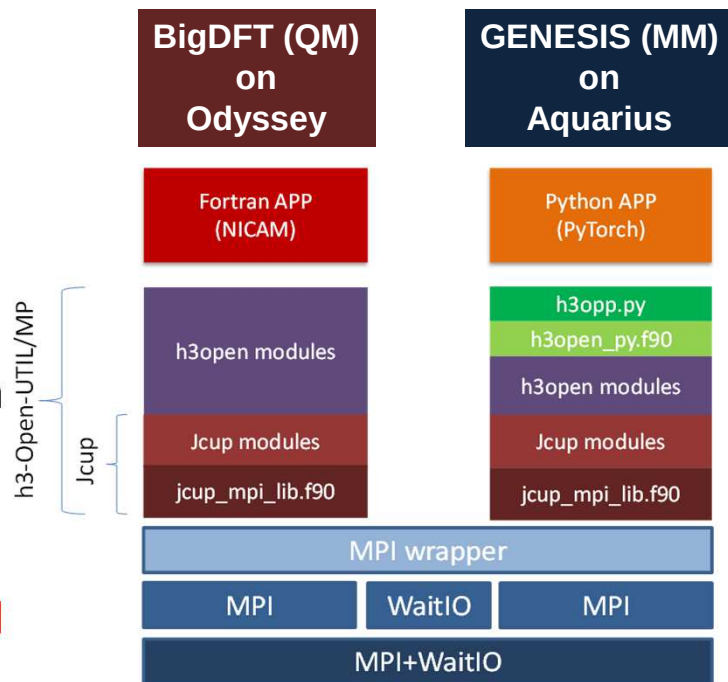
- CEA
 - France Boillod-Cerneux
 - Luigi Genovese
- U.Tokyo/RIKEN
 - Kengo Nakajima
 - Shinji Sumimoto
- RIKEN
 - Yuji Sugita
 - Shingo Ito
 - Kiyoshi Yagi
 - William Dawson

Big-DFT with GENESIS for SARS-CoV-2 Main Protease (CEA, RIKEN, U.Tokyo) (2/2)



東京大学
THE UNIVERSITY OF TOKYO

- We will exploit the heterogeneous architecture of Wisteria/BDEC-01 to build a coupled QM-MM workflow.
 - The MM workflow will run the “GENESIS” (RIKEN) on Aquarius to exploit its GPU nodes and provide samples from a trajectory that are sent to the QM-MM workflow running “BigDFT” on Odyssey.
 - BigDFT was already optimized for A64FX architecture under CEA-RIKEN collaboration.
- In FY.2024, we will construct preliminary version of QM-MM workflow using h3-Open-BDEC on Wisteria/BDEC-01, and make evaluations.
- While we do not need real-time coupling between QM-MM in this case, we may extend this idea to interaction between QM and MM.

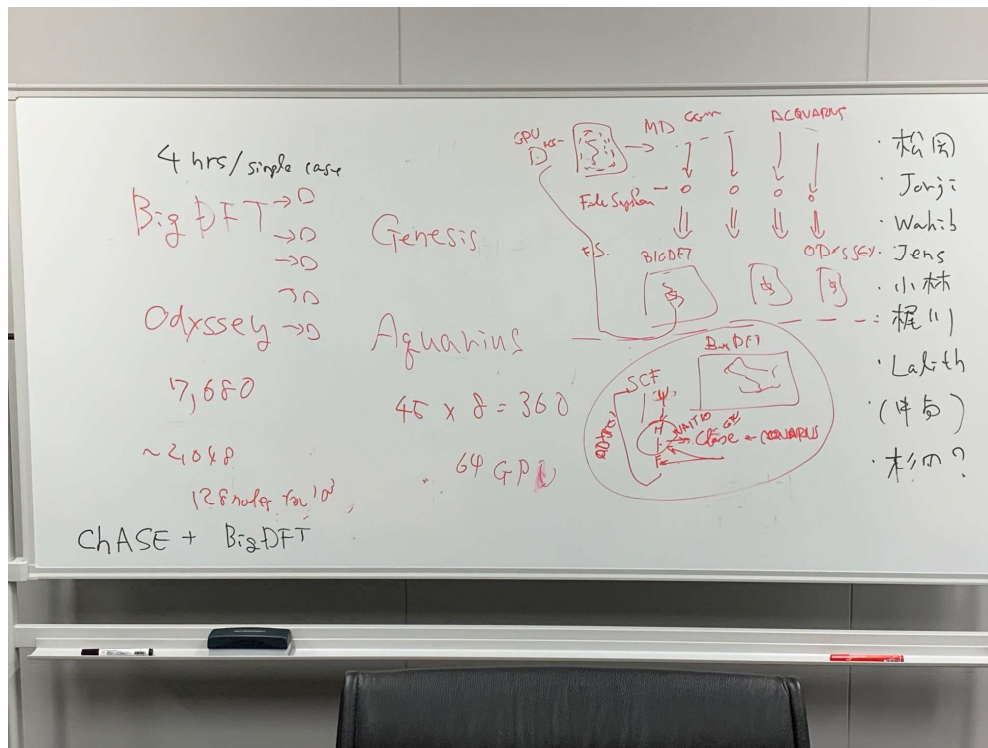


WaitIO inspired something for Luigi

I still keep the discussion on Dec. 5, 2023 at my office in Kobe on my white board



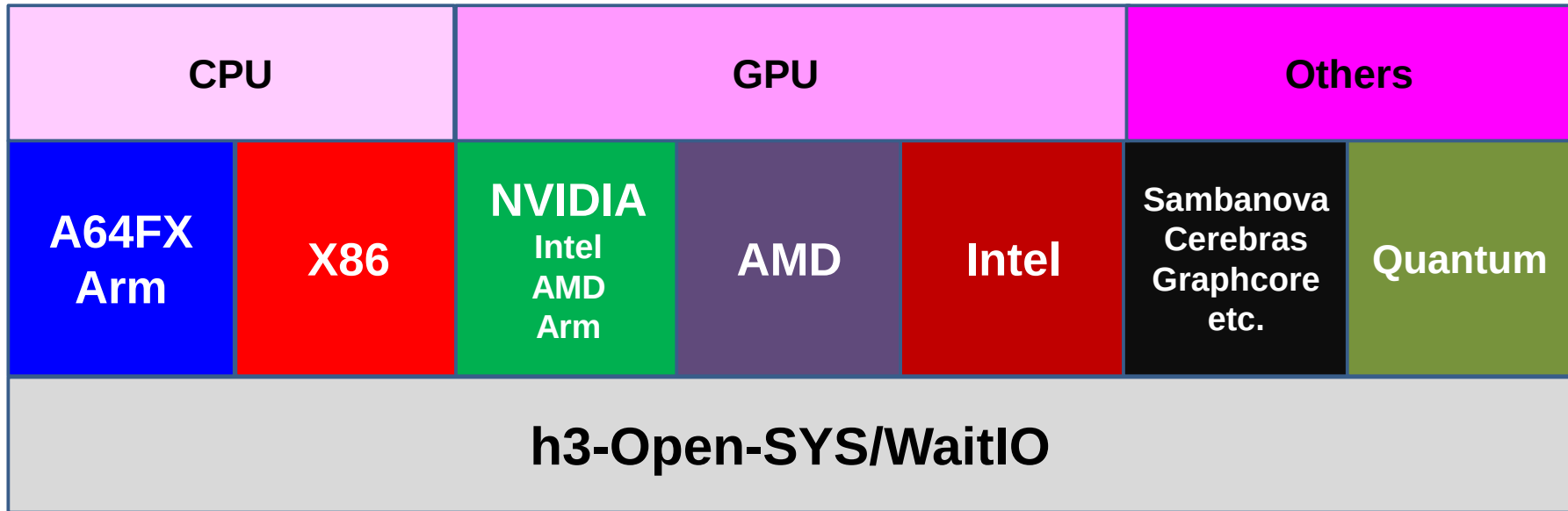
東京大学
THE UNIVERSITY OF TOKYO



- ppOpen-HPC
- h3-Open-BDEC on Wisteria/BDEC-01
- **JHPC-quantum**

Anything is possible with WaitIO

WaitIO over Internet/cloud is possible



量子・スパコン連携プラットフォーム (JHPC-quantum)

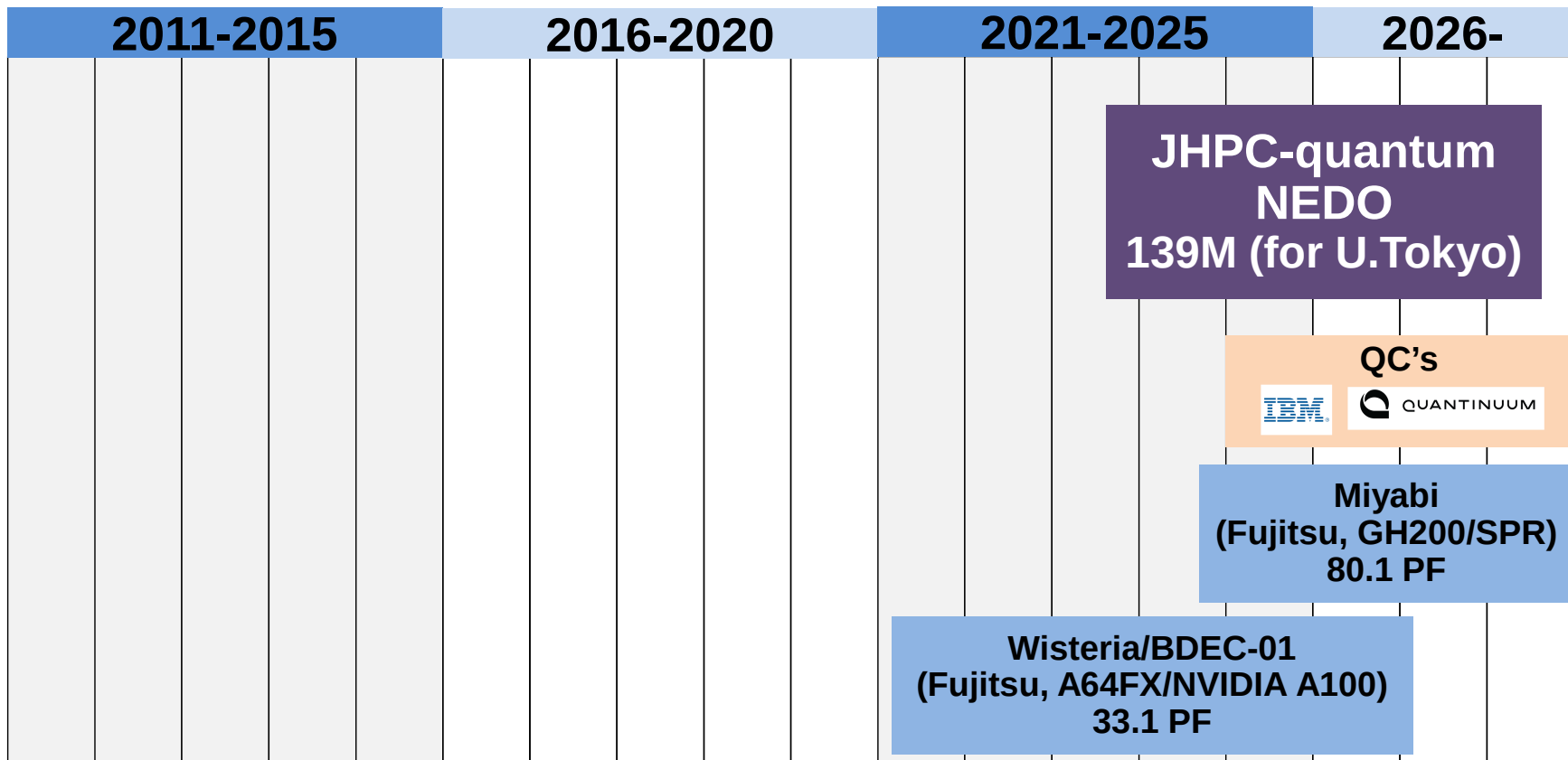
<http://jhpc-quantum.org/>

- JHPC-quantum (FY.2023-FY.2028)
 - 理研R-CCS, ソフトバンク, 東大, 阪大
 - 経産省・NEDOポスト5G関連事業, 産業応用への期待
- 2基のリアルな量子コンピュータを導入
 - IBM(超伝導型), 100+Qubit, 理研神戸
 - Quantinuum(Ion-Trap型), 20+Qubit, 理研和光
- QC-HPCハイブリッド連携
 - QCを加速装置として扱う, **追加**
 - 量子科学, 量子機械学習+シミュレーション
- 東大の役割: 連携環境向けシステムソフトウェア
 - h3-Open-BDECの拡張
- 2025年度末: 富岳・東大スパコンとQCの連携開始

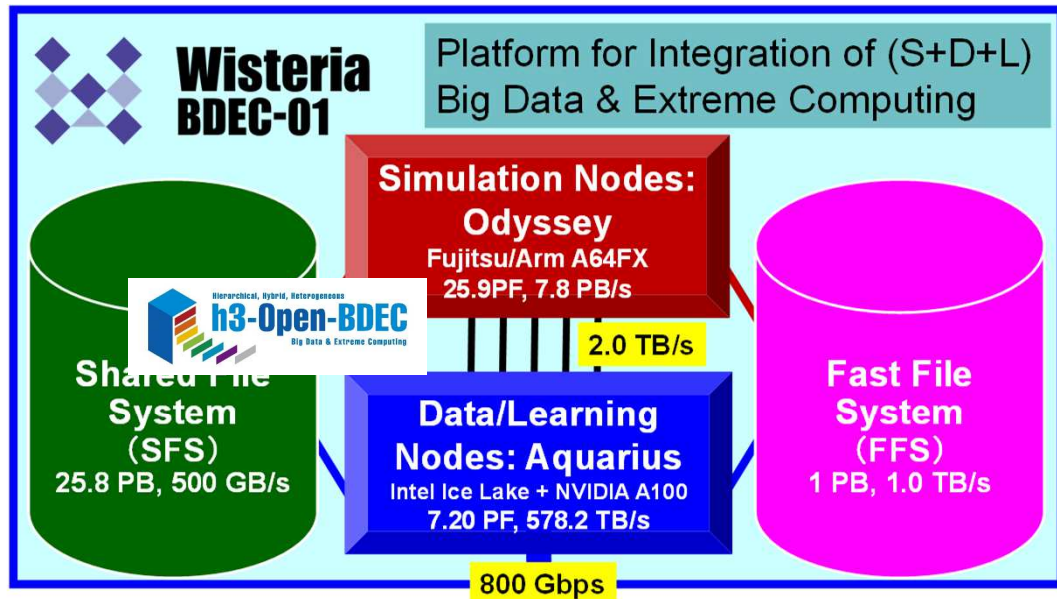


CPU		GPU			Others	
A64FX Arm	X86	NVIDIA Intel AMD Arm	AMD	Intel	Sambanova Cerebras Graphcore etc.	Quantum
h3-Open-SYS/WaitIO						





現在: Odyssey-AquariusにQC-HPCハイブリッド シミュレーション環境を構築



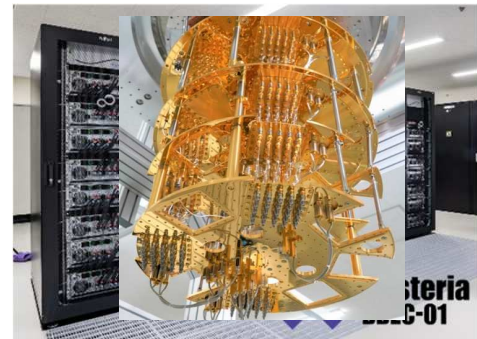
External
Resources



External Network



External
Resources



『CUDA-Q+Wisteria/BDEC-01+h3-Open-BDEC』で『量子・HPCハイブリッド』を体験してみよう

- CUDA-Q
 - NVIDIAが開発している量子古典ハイブリッド計算のためのオープンソースプラットフォーム
 - Aquariusにインストール済み
- 本日は、疑似QC-HPCハイブリッド環境でVQE (Variational Quantum Eigensolver) を体験
 - Aquarius (GPU) with CUDA-Q: QC
 - Odyssey (CPU): HPC
 - Pythonで記述した簡単なプログラム
 - h3-Open-BDEC
 - WatIOのPython版によるQC-HPC連携
- 世界で初めての「本当の」QC-HPCが繋がったハイブリッド環境 (QCは本物ではないが)



『CUDA-Q+Wisteria/BDEC-01+h3-Open-BDEC』で 『量子・HPCハイブリッド』を体験してみよう

- QC-HPCハイブリッド環境でVQE (Variational Quantum Eigensolver) を体験

10:00-10:10 10:10-10:30	挨拶＋趣旨背景 NVIDIA Quantumの紹介(座学)
10:30 - 12:00	CUDA-Q入門(座学＋演習) ログイン, すごく簡単な例
13:00 - 14:30	発展的な利用例(座学＋演習) 入門の続き, VQEの話し
15:00 - 17:00	QC-HPCハイブリッド環境(座学＋演習)

- 次回はMiyabi(NVIDIA GH200搭載, 2025年1月運用開始)を使ったチュートリアルを予定



Development for QC-HPC Hybrid Environment

- Extension of WaitIO Library
 - WaitIO4Py
 - Python Interface
 - WaitIO-Router
 - Communication Library among Supercomputer Centers over Sinet
- FY.2024: Simulators on Wisteria/BDEC-01
- FY.2025: Real QC-HPC Environment

WaitIO4Py: Sample Code

• PingPong Program

```
import waitio
import numpy as np
```

```
def main():
```

```
    grp = waitio.Waitio_group()
    rankp = waitio.Waitio_rank()
    sizep = waitio.Waitio_size()
```

```
    waitio.waitio_create_universe(grp)
```

```
    waitio.waitio_group_rank(grp, rankp)
    waitio.waitio_group_size(grp, sizep)
```

```
    myrank = rankp.wval.value # read ctypes.value
    mysize = sizep.wval.value # read ctypes.value
```

```
    buf = np.zeros((10,2), dtype=np.uint16)
    buf1 = np.zeros((10,2), dtype=np.uint16)
    req = waitio.Waitio_request()
    req1 = waitio.Waitio_request()
```

```
    len = 32
    tag = 0
    num = 0
    while num < 100:
        print("loop=", num)
        if(myrank == 0):
            dst = 1
            buf[0,0] = 100 + num
            ret = waitio.waitio_isend(grp, dst, buf, len, tag, req)
            waitio.waitio_wait(req)
            ret = waitio.waitio_irecv(grp, dst, buf1, len, tag, req1)
            waitio.waitio_wait(req1)
            print (" data=", buf1[0,0])
        else:
            dst = 0
            ret = waitio.waitio_irecv(grp, dst, buf1, len, tag, req1)
            waitio.waitio_wait(req1)
            print (" data1=", buf1[0,0])
            buf[0,0] = num
            ret = waitio.waitio_isend(grp, dst, buf, len, tag, req)
            waitio.waitio_wait(req)

        num = num + 1

    waitio.waitio_finalize()
```



```
waitio-test.py@wisteria03
File Edit Options Buffers Tools Python Help

#!/usr/bin/env python
import waitio
import numpy as np

def main():
    grp = waitio.Waitio_group()
    rankp = waitio.Waitio_rank()
    sizep = waitio.Waitio_size()

    waitio.waitio_create_universe(grp)

    waitio.waitio_group_rank(grp, rankp)
    waitio.waitio_group_size(grp, sizep)
    myrank = rankp.wval.value # read ctypes.value
    mysize = sizep.wval.value # read ctypes.value
    print (rankp.wval)
    print (sizep.wval)
    print (grp)

    print (myrank, mysize)

    #buf = np.array([0,0,0,0,9,0,0], dtype=np.uint8)
    buf = np.zeros((10,2), dtype=np.uint32)
    buf1 = np.zeros((10,2), dtype=np.uint32)

    req = waitio.Waitio_request()
    req1 = waitio.Waitio_request()
    len = 32
    tag = 0
    num = 0
    while num < 100:
        print("loop=", num)
        if(myrank == 0):
            dst = 1
            buf[0,0] = 10000 + num
            ret = waitio.waitio_isend(grp, dst, buf, len, tag, req)
            waitio.waitio_wait(req)
            ret = waitio.waitio_irecv(grp, dst, buf1, len, tag, req1)
            waitio.waitio_wait(req1)
            print (" data=", buf1[0,0])
        else:
            dst = 0
            ret = waitio.waitio_irecv(grp, dst, buf1, len, tag, req1)
            waitio.waitio_wait(req1)
            print (" data1=", buf1[0,0])
            buf[0,0] = num
            ret = waitio.waitio_isend(grp, dst, buf, len, tag, req)
            waitio.waitio_wait(req)

        num = num + 1

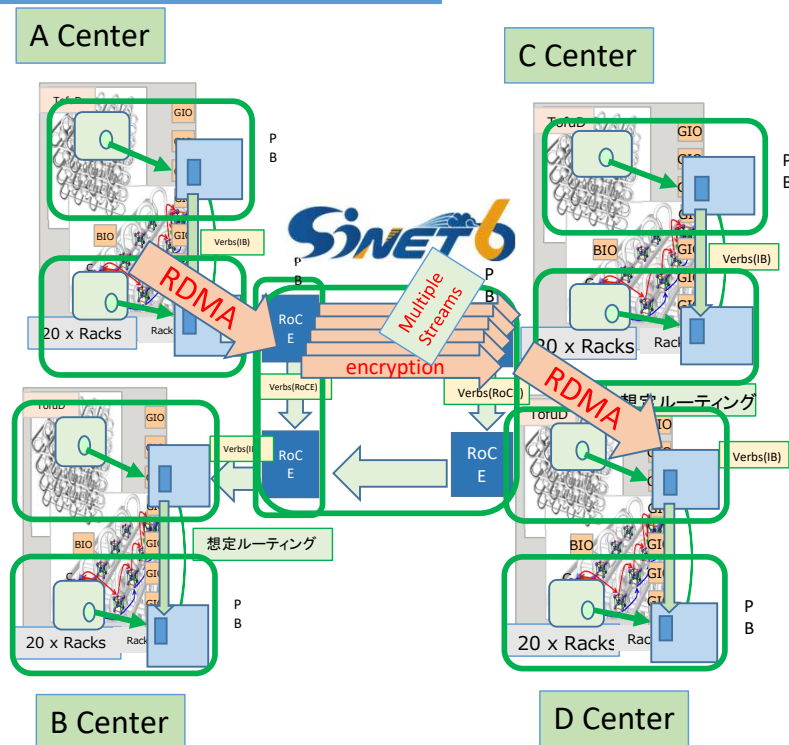
    waitio.waitio_finalize()

main()
print ("end of routine")

<--- waitio-test.py All L24 Git:main (Python EIDoc) 11:09AM 3.94
<M-Scroll Lock> Is undefined
```

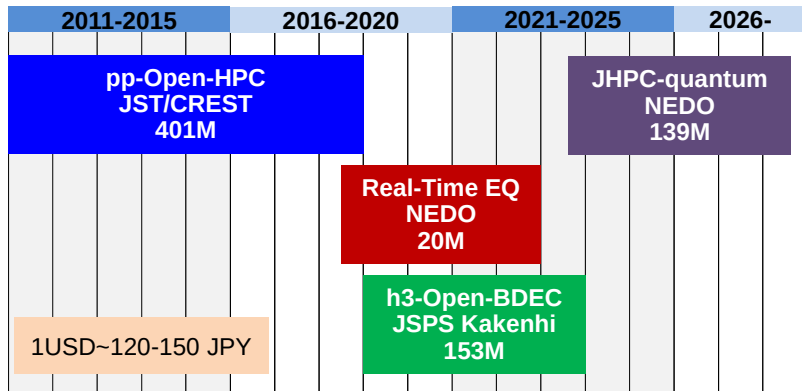
WaitIO-Router for Center-Wide Communication

- In a Center: High performance communication using RDMA and Zero-Copy Routing.
 - WaitIO-Verbs and WaitIO-Tofu
 - WaitIO-Router for Zero-Copy Routing
- Among Centers: Converting Single Stream to Multiple Streams
 - For Performance: Multiple RDMA(RoCE) and Socket streams
 - For Security: with Encryption and Scramble packets



まとめ(1/2)カプラーの歴史

- ppOpen-HPC
 - MPI-based
 - 大気(NICAM)-海洋(Ocean)
 - +その他のアプリケーション
- h3-Open-BDEC
 - ヘテロジニアスな環境(Wisteria/BDEC-01)
 - WaitIOとの結合
 - WaitIOは「世界を変えた」
 - (S+D+L)融合, アンサンブル連成他
- JHPC-quantum
 - QC-HPCハイブリッド環境
 - WaitIOの更なる進化(WaitIO4Py, WaitIO-Router)



JHPC-quantum

まとめ(2/2)持続可能性(Sustainability)

- ソフトウェアのSustainability⇒開発, 保守のための財源を継続して確保することに依存する
- 新しい対象アプリケーション, 対象システムの開拓, とそれに向けた新機能の開発(e.g. WaitIO)がとても重要
 - 「カプラ+WaitIO」は「AI for Science」においても重要なワークフローツールとなる
- 関係者はいつも大きく目を見開いて, 自分のやっていることがどこか新しいことに使えないか, そのためには何をすれば良いか(e.g. 追加機能の開発), 考えていなければならない
 - Those involved must always keep their eyes open and pay attention to whether it can be applied somewhere.
- 他分野にわたる多くのグループが, 同じように考えて, 情報発信も含めて積極的に活動すれば, 何かの拍子に, 相乗効果が生まれるかもしれない

NCTS School@国立台湾大学

Feb.17-18, 2025 (before HPC Asia 2025)

https://ncts.ntu.edu.tw/events_3_detail.php?nid=358

Day-1	09:10-10:00	Overview: Wisteria/BDEC-01, h3-Open-BDEC	Nakajima
	10:10-11:00	Earthquake Simulations, Hands-on	
	11:10-12:00	Overview of h3-Open-SYS/WaitIO, Parallel Computation on Heterogeneous Environment, Hands-on	Sumimoto
	13:10-14:00		
	14:10-15:00		
	15:10-16:00		
	16:10-17:00	Introduction to Global Atmospheric Simulation Overview of h3-Open-UTIL/MP	Yashiro Arakawa
Day-2	09:10-10:00	Coupling of Global Atmospheric Simulation and AI, Hands-on	
	10:10-11:00		
	11:10-12:00		
	13:10-14:00		
	14:10-15:00	Future Issues, QC (Quantum Computing) -HPC Hybrid	Nakajima Arakawa
	15:10-16:30	In-Depth Discussions	




Integration of Simulation, Data, and Machine Learning on A Heterogeneous Supercomputer System

Time

2025.2.17
09:00-17:00

2025.2.18
09:00-15:00

Venue

Room 505,
Cosmology Building-NTU

Speakers

Kengo Nakajima
The University of Tokyo
RIKEN R-CCS

Takashi Arakawa
CiMLech,
The University of Tokyo

Shinji Sumimoto
The University of Tokyo

Hisashi Yashiro
National Institute for
Environmental Studies

Organizers

Weichung Wang
National Taiwan University

Feng-Nan Hwang
National Central University

Hung-Chi Kuo
Department of Atmospheric Sciences,
National Taiwan University

Introduction & Purposes

In this 2-day intensive course, participants will learn about the development of coupled applications on heterogeneous systems using WaitIO and UtilMP, and in particular the methods of combining simulation for computational science and machine learning. Starting from the basics of coupled computation, this course focuses on how to develop coupled applications between Odyssey and Aquarius on Wisteria/BDEC-01 using WaitIO and UtilMP.



Registration

Outline & Descriptions

First, participants will learn how to rewrite weakly coupled applications via files using WaitIO with MPI-like communication API.

Next, they will learn about integration of simulation and machine learning using UtilMP. In particular, by coupling simulation code written in Fortran/C with machine learning libraries written in Python, they will learn about application coordination that takes advantage of the characteristics of Wisteria/BDEC-01, not limited to multi-physics coupling. Furthermore, hands-on exercises using Wisteria/BDEC-01 are provided.