Road to "AI for Science": Exploring Software Sustainability through "Couplers"

Kengo Nakajima
The University of Tokyo & RIKEN R-CCS

1st International Workshop on Distributed and Parallel Programming for Extreme-scale AI (DP2E-AI 2025)
June 16-17, 2025, Mines Paris-PSL, Paris, France

Co-Authors

- Shinji Sumimoto (The University of Tokyo, Japan)
- Takashi Arakawa (CliMTech/The University of Tokyo, Japan)
- Hisashi Yashiro (National Institute for Environmental Studies, Japan)
- Yao Hu (The University of Tokyo, Japan)
- Kazuya Yamazaki (The University of Tokyo, Japan)
- Ikko Hamamura (NVIDIA)

NVIDIA

CUDA QUANTUM

東京大学 THE UNIVERSITY OF TOKYO

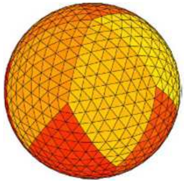
CliMTech

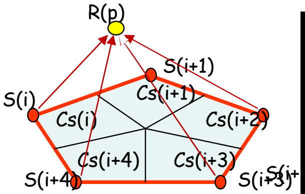
国立研究開発法人 国立環境研究所 National Institute for Environmental Studies

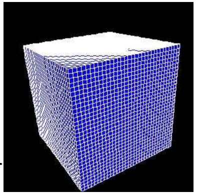
Acknowledgements

- 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

Coupler

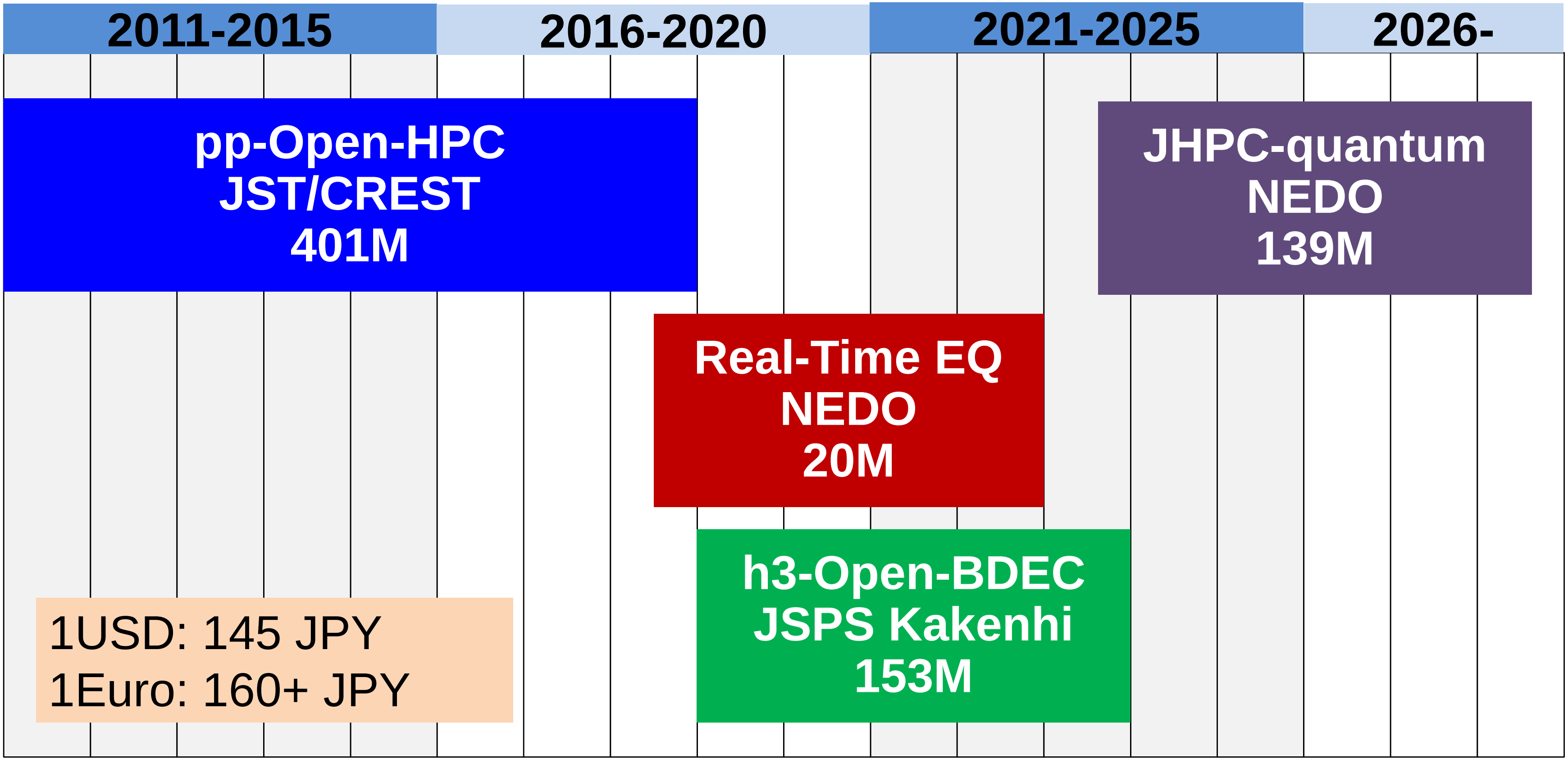




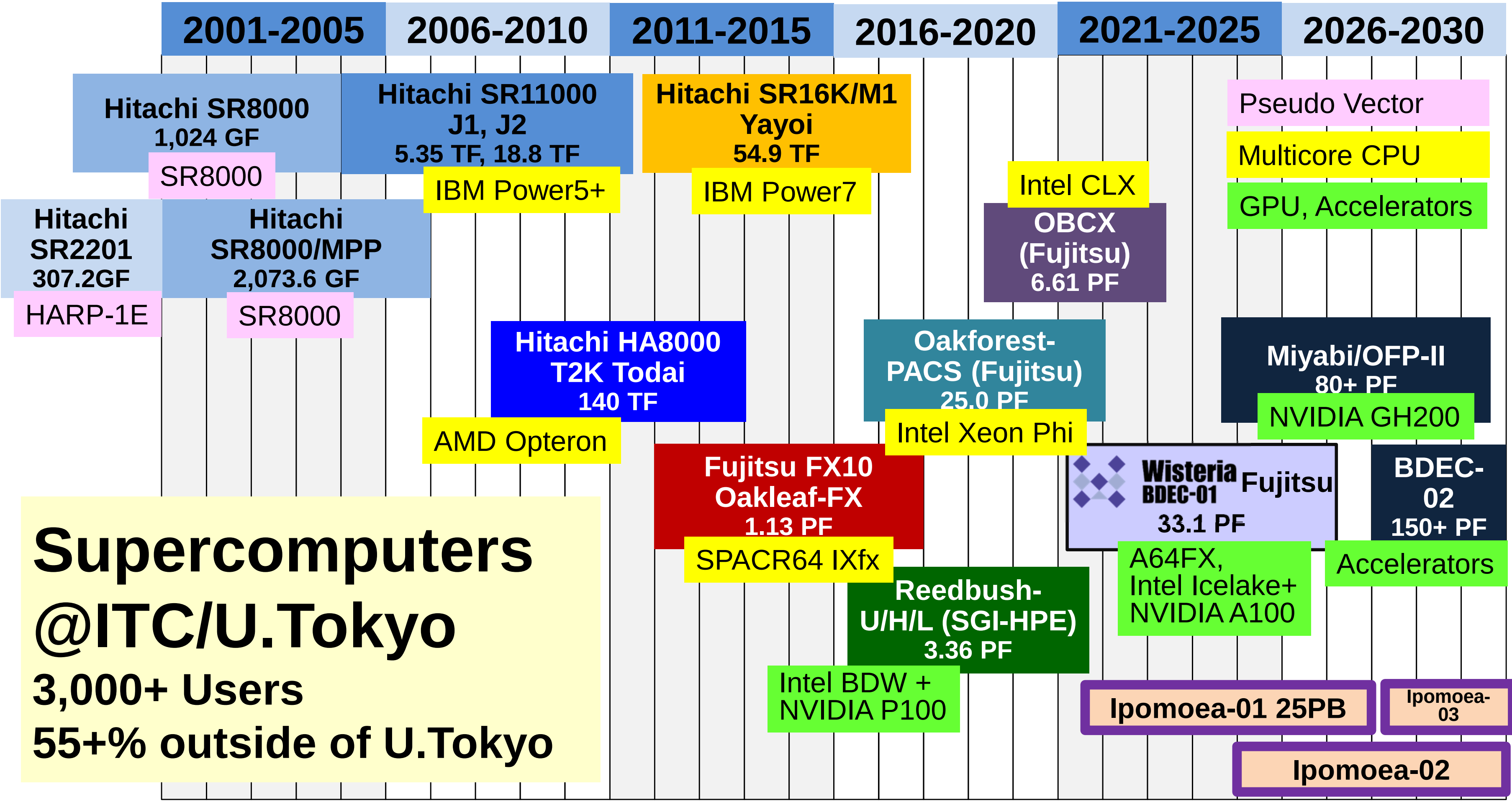


- In this talk, I focus on *coupler.*
- *Coupler* is originally a tool for coupling multiple scientific applications loosely : *Multiphysics Coupler: Ocean-Atmosphere, Fluid-Structure etc.*
  - Connecting multiple applications using various grid/mesh systems: *e.g. FEM-FDM*
  - Interpolation/Extrapolation
- Complex workflows for **<u>REAL-TIME INTEGRATION</u>** of scientific simulations, data assimilation/analytics, and machine learning have become very common in these days (Integration of (S+D+L)).
- **Overall time-to-solution is an important issue from the view point of performance, while optimization of each procedure is also essential.**
- In my projects, coupler has been extended to a tool for efficient workflow in recent 10+ years.
- I try to show the pathways towards sustainability through the history.

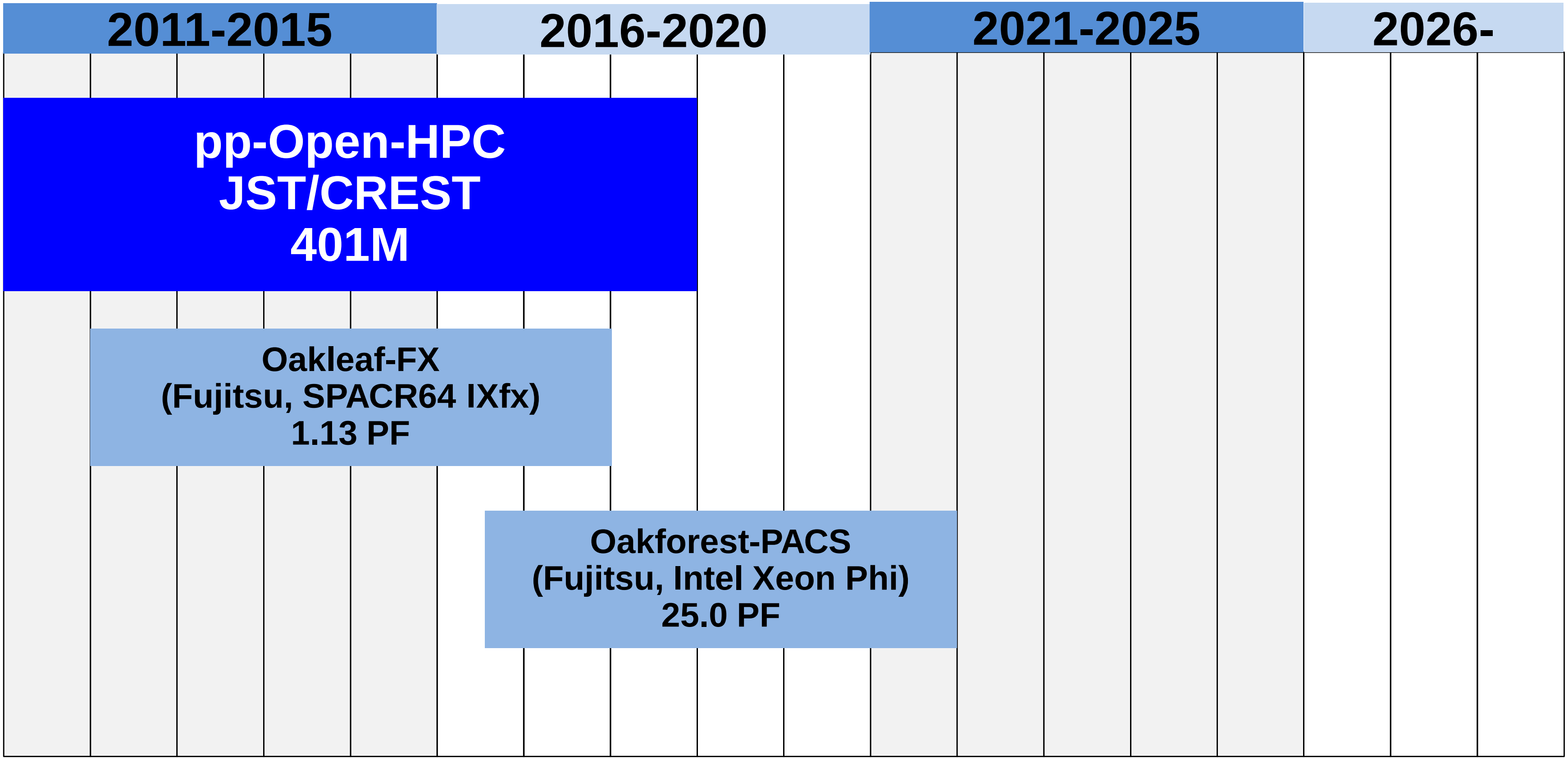
History of the Projects & Budgets (JPY)



- 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
Pseudo Vector
Multicore CPU
GPU, Accelerators
Intel CLX
OBCX
(Fujitsu)
6.61 PF
Hitachi
SR2201
307.2GF
HARP-1E
Hitachi
SR8000/MPP
2,073.6 GF
SR8000
Hitachi HA8000
T2K Todai
140 TF
AMD Opteron
Oakforest-
PACS (Fujitsu)
25.0 PF
Intel Xeon Phi
Miyabi/OFP-II
80+ PF
NVIDIA GH200
Fujitsu FX10
Oakleaf-FX
1.13 PF
SPACR64 IXfx
Wisteria
BDEC-01
Fujitsu
33.1 PF
A64FX,
Intel Icelake+
NVIDIA A100
BDEC-
02
150+ PF
Accelerators
Reedbush-
U/H/L (SGI-HPE)
3.36 PF
Intel BDW +
NVIDIA P100
Ipomoea-01 25PB
Ipomoea-
03
Ipomoea-02
Supercomputers
@ITC/U.Tokyo
3,000+ Users
55+% outside of U.Tokyo

History of the Projects & Budgets (JPY)



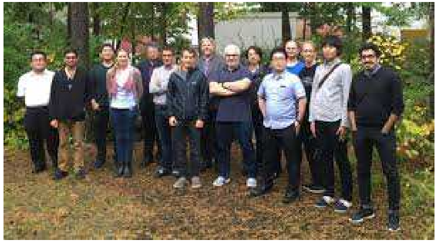
ppOpen-HPC: Overview

- 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 until FY.2018 under collaborations with German Projects (SPPEXA/DFG): ESSEX-II (Prof. Gerhard Wellein (FAU Erlangen))
- 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

独立行政法人 科学技術振興機構 Japan Science and Technology Agency

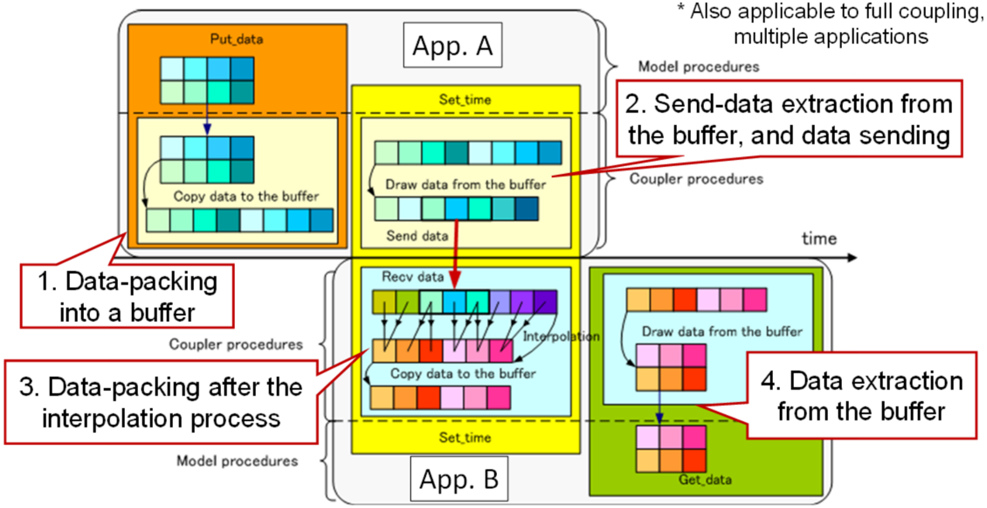
CREST 戦略的創造研究推進事業 Core Research for Evolutionary Science and Technology

DFG Deutsche Forschungsgemeinschaft



SPPEXA

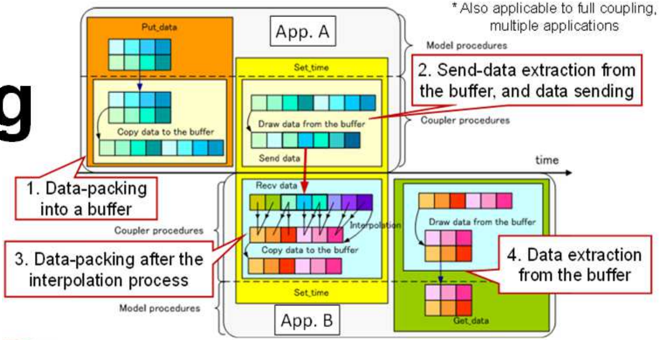
ppOpen-MATH/MP

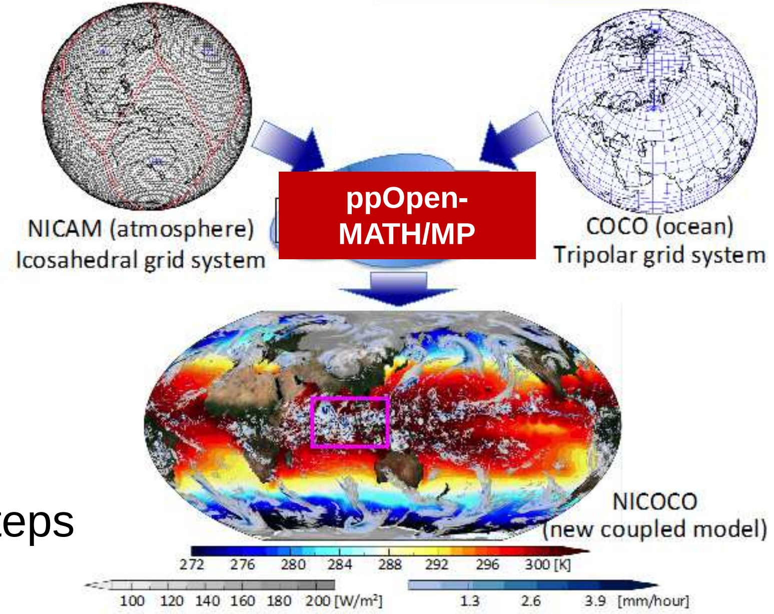


- Weak-Coupling of Multiple Applications
  - Each application does a single computation
  - Ocean-Atmosphere
  - Fluid-Structure
- Originally developed as a coupler for NICAM (atmosphere, semi-unstructured), and COCO (ocean, structured) in global climate simulations using K computer
- Non-Centralized/Fully Distributed Operations, No Master Process
- Various types of mesh/grid systems are supported
- Developed coupler is extended to more general use.

Global Climate Simulation by Atmosphere-Ocean Coupling NICAM-COCO [Sato et al. 2017]

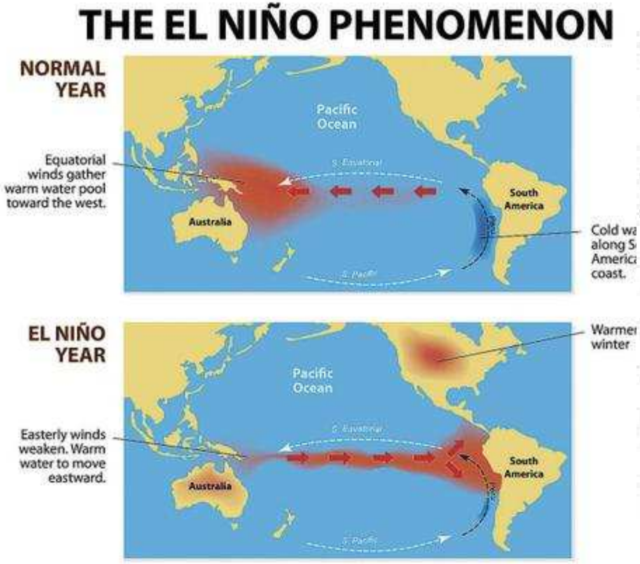
- Global Model
  - Atmosphere: NICAM: semi-unstructured
  - Ocean: COCO: stuructured
- ppOpen-MATH/MP
  - Coupler
  - Tool for Multiphysics Simulation
- Simulations for 2-Months
  - NICAM: 14km/110M meshes, 170K steps
  - COCO: 0.25-1.00 deg./5.8M meshes, 6K steps
  - (640+32) nodes of K computer, 28-hours

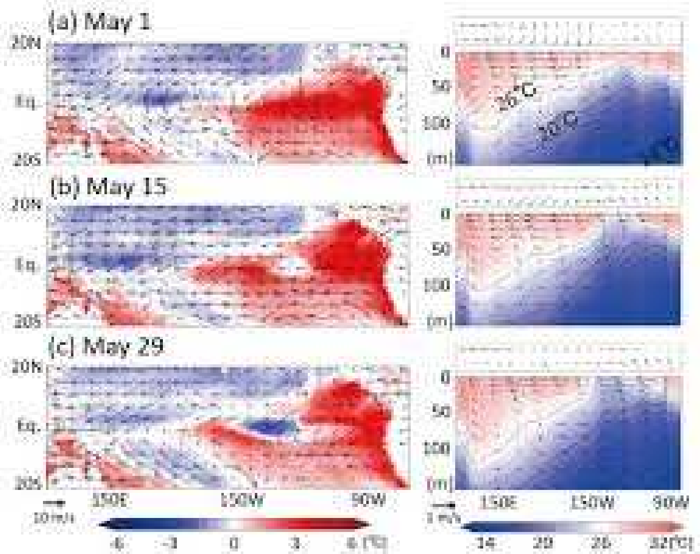




Simulations of Super El Niño

Mechanism of the Abrupt Terminate of Super El Niño in 1997/1998 has been revealed by Atmosphere-Ocean Coupling Simulations for the Entire Earth using ppOpen-HPC on the K computer [Sato et al. 2017]





2017.9.20
日経(夕)

「スーパーエルニーニョ」終息
巨大雲 引き金に
東大など解明

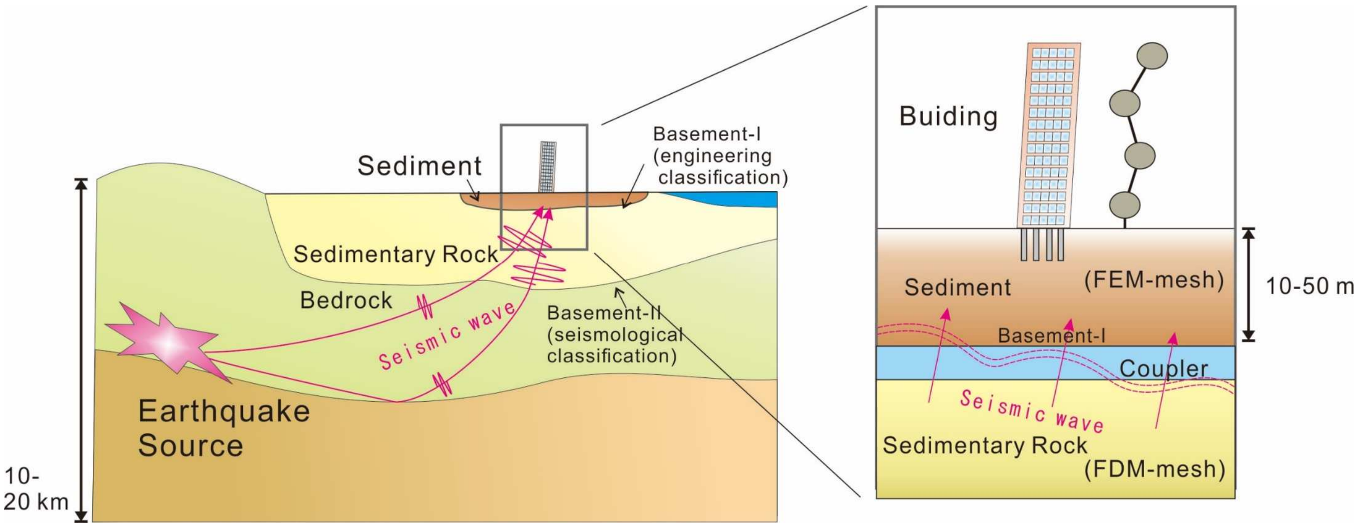
東京大学大気海洋研究所と理化学研究所、海洋研究開発機構は、1997～98年に発生して世界に異常気象をもたらした「スーパーエルニーニョ」現象が急に終息した原因を突き止めた。熱帯の巨大雲が引き金になっていた。台風の発生予測などの精度向上にもつながる成果という。

エルニーニョは太平洋の東側の海面温度が平年より高くなる現象。特に強まった状態はスーパーエルニーニョと呼ばれ、97～98年のほか2015～16年にも発生した。世界各地で豪雨など異常気象の一因になったと考えられている。

研究チームは雲の発生や動きを計算する手法を改良し、海洋との相互作用を精度良く予測する手法を開発した。理研のスーパーコンピューター「京」を使った数値計算で、赤道付近に巨大な雲ができて東に進む「マッデン・ジュリアン振動」と呼ぶ現象がスーパーエルニーニョ終息の引き金になったと突き止めた。

東西方向の数千キロに広がる巨大な雲から吹き出す東風が強まって海面付近の温かい海水が動かされ、入れ替わりに冷たい海水が下から上がってくるようになったという。

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).

[M. Matsumoto et al. 2015]

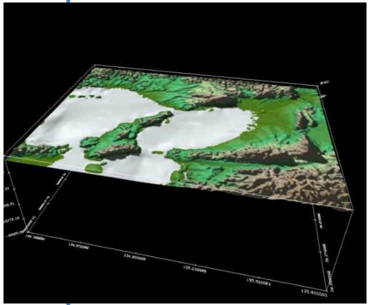
Weak-Coupled Simulation by the ppOpen-HPC Libraries

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

Seism3D+

ppOpen-APPL/FDM

Velocity →



ppOpen-MATH/MP

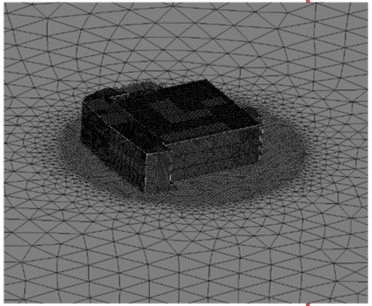
Principal Functions

- ✓ Make a mapping table
- ✓ Convert physical variables
- ✓ Choose a timing of data transmission
- ⋮

FrontISTR++

ppOpen-APPL/FEM

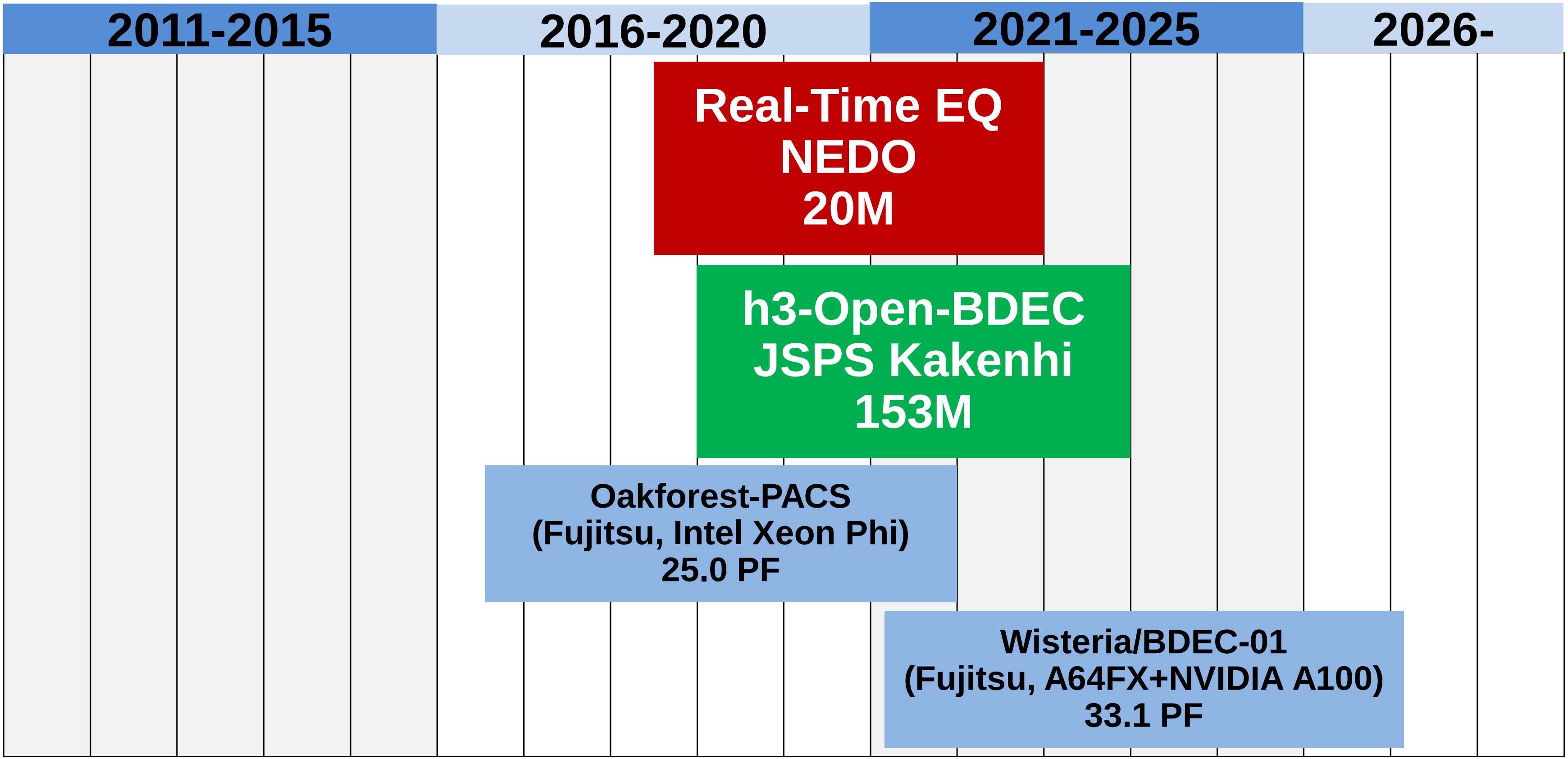
→ Displacement



[M. Matsumoto et al. 2015]

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

History of the Projects & Budgets (JPY)



Integration of (S+D+L) has been our main strategy in recent 10 years

- Various Types of Workloads on Supercomputers
  - Computational Science & Engineering: Simulations
  - Big Data Analytics +AI, Machine Learning …
- **Integration of (<u>Simulation+Data+ Learning</u>) (S+D+L) is important towards Society 5.0, <u>Human-Centered, Safe and Secure Society</u> proposed by Japanese Government**
  - **By Integration of Cyber & Physical Space**
- **Project BDEC (Big Data & Extreme Computing) since 2015**
  - **Development of Platform for Integration of (S+D+L)**
  - **Focusing on S (Simulation)**

BDEC (Big Data & Extreme Computing)

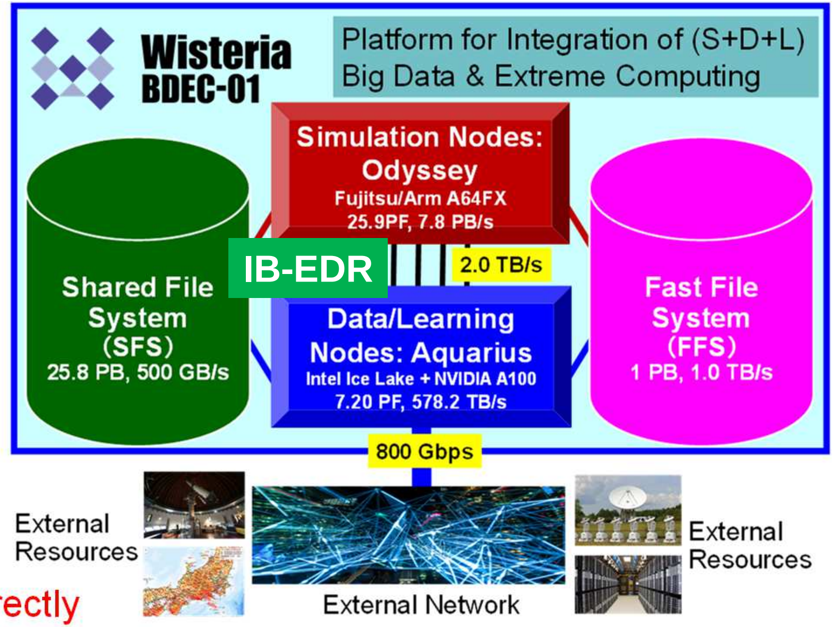
S + D + L

東京大学
THE UNIVERSITY OF TOKYO

Wisteria/BDEC-01

The 1st BDEC System (Big Data & Extreme Computing) HW Platform for Integration of (S+D+L)

- Operation started on May 14, 2021
- 33.1 PF, 8.38 PB/sec by **Fujitsu**
  - ~4.5 MVA with Cooling, ~360m^2
- 2 Types of Node Groups
  - Simulation Node Group: Odyssey
    - Fujitsu PRIMEHPC FX1000 (A64FX), 25.9 PF
      - 7,680 nodes (368,640 cores), Tofu-D
      - General Purpose CPU + HBM
      - Commercial Version of "Fugaku"
  - Data/Learning Node Group: Aquarius
    - Data Analytics & AI/Machine Learning
    - Intel Xeon Ice Lake + NVIDIA A100, 7.2PF
      - 45 nodes (90x Ice Lake, 360x A100), IB-HDR
    - DL nodes are connected to external resources directly
- 2 Node Groups are connected through:
  - IB-EDR (2.0TB/sec) & Fast File System (FFS, 1.0TB/sec)



Wisteria BDEC-01

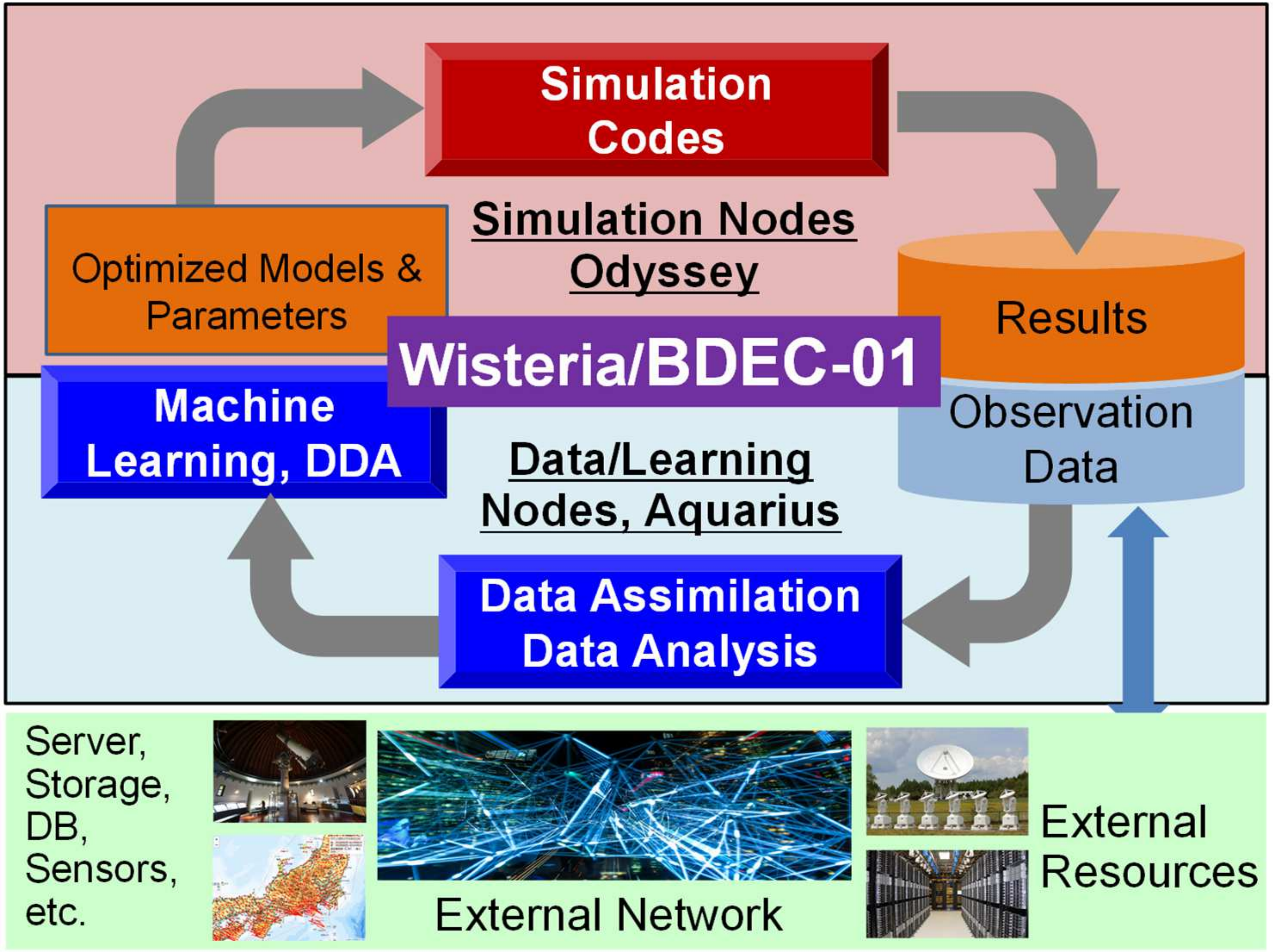
Simulation Nodes
Odyssey
25.9 PF, 7.8 PB/s

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

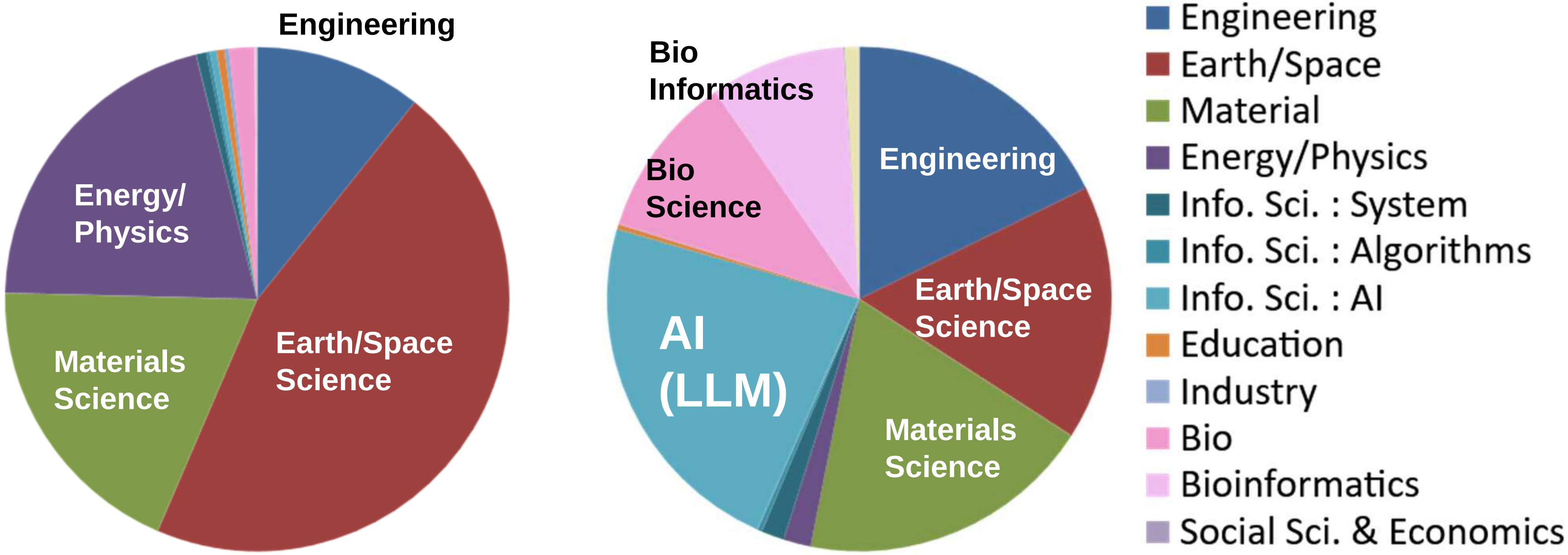
Shared File System (SFS)
25.8 PB, 0.50 TB/s

Data/Learning Nodes
Aquarius
7.20 PF, 578.2 TB/s

Wisteria BDEC-01



Research Area based on Machine Hours
CPU, GPU (April 2024-March 2025)

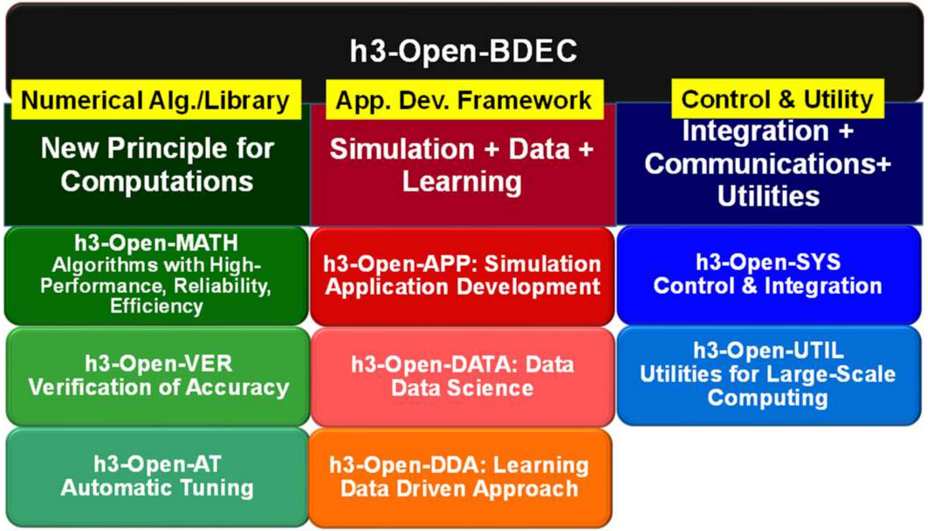


Odyssey A64FX

Aquarius A100

h3-Open-BDEC: Innovative Software Platform for Integration of (S+D+L) on the BDEC System, such as Wisteria/BDEC-01

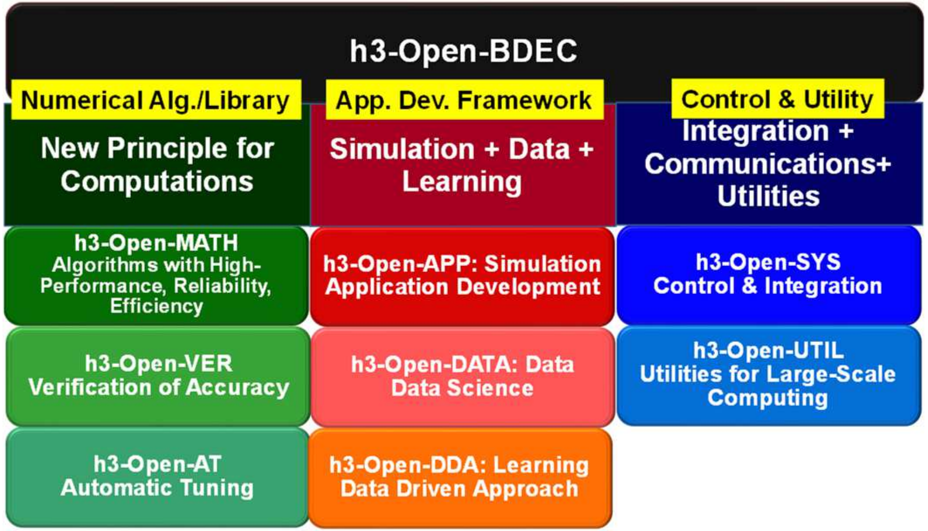
- 5-year project supported by Japanese Government (JSPS) since 2019
  - FY.2023 was the final year
    - Until the end of March 2024
- Leading-PI: Kengo Nakajima (The University of Tokyo)
- Total Budget: 1.40M USD



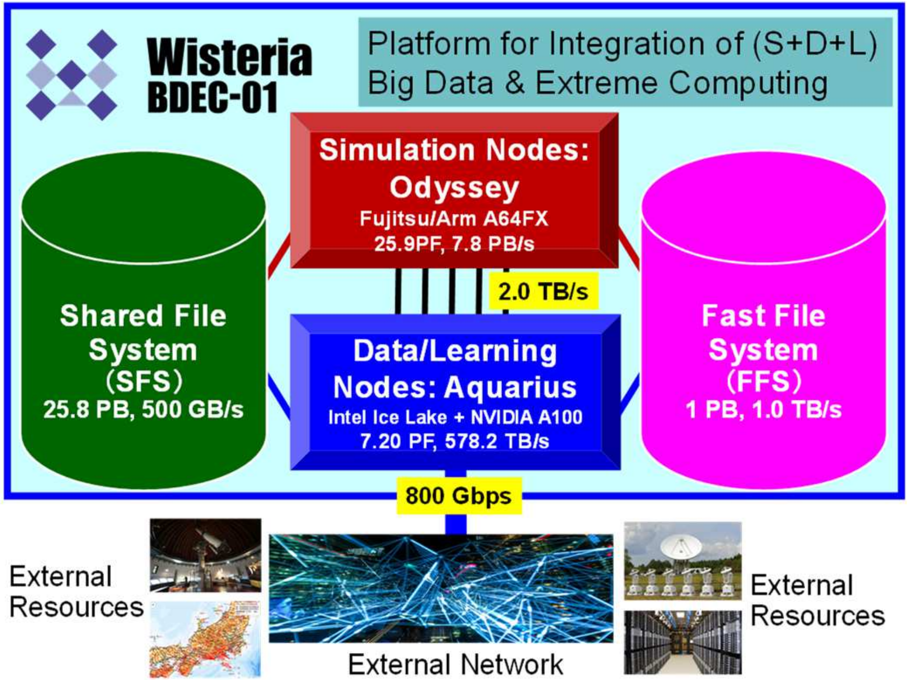
h3-Open-BDEC: Innovative Software Platform for Integration of (S+D+L)

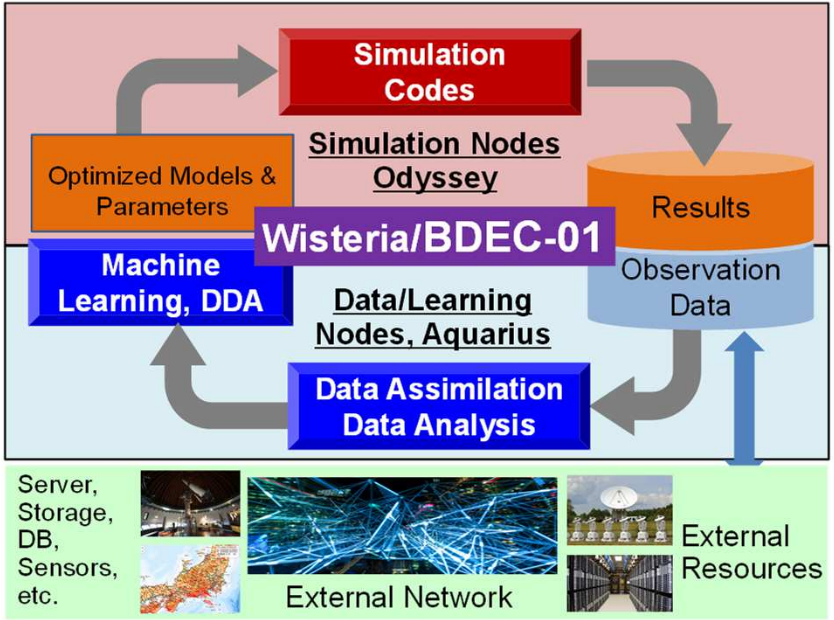
FY.2019-2023 supported by JSPS
Leading PI: Kengo Nakajima

- "Three" Innovations
  - New Principles for Numerical Analysis by Adaptive Precision, Automatic Tuning & Accuracy Verification
  - Integration of (S+D+L) by Hierarchical Data Driven Approach (*h*DDA)
  - Software & Utilities for Heterogenous Environment, such as Wisteria/BDEC-01



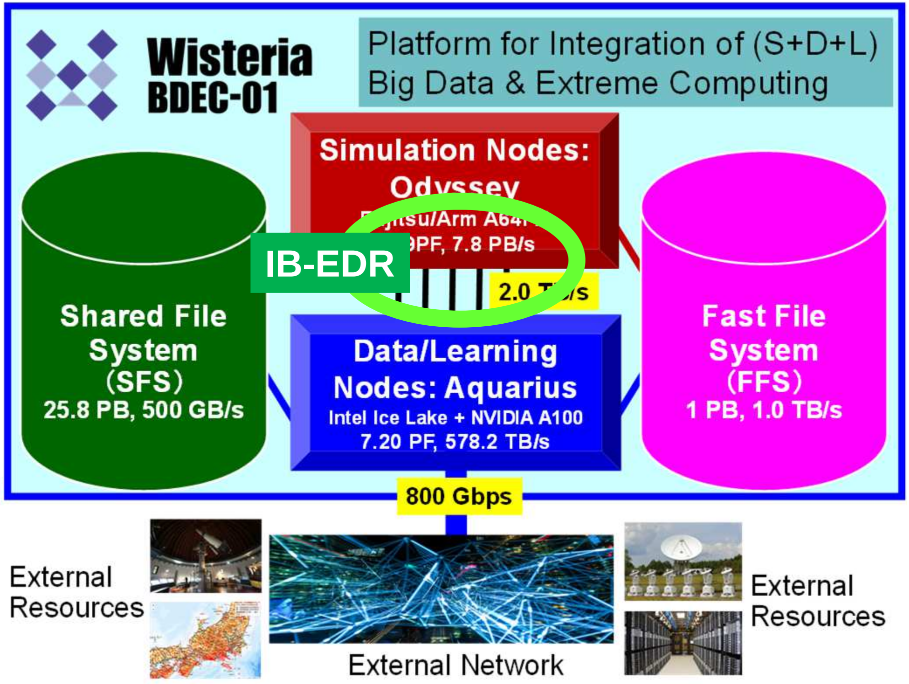
Wisteria/BDEC-01: The First "Really Heterogenous" System in the World





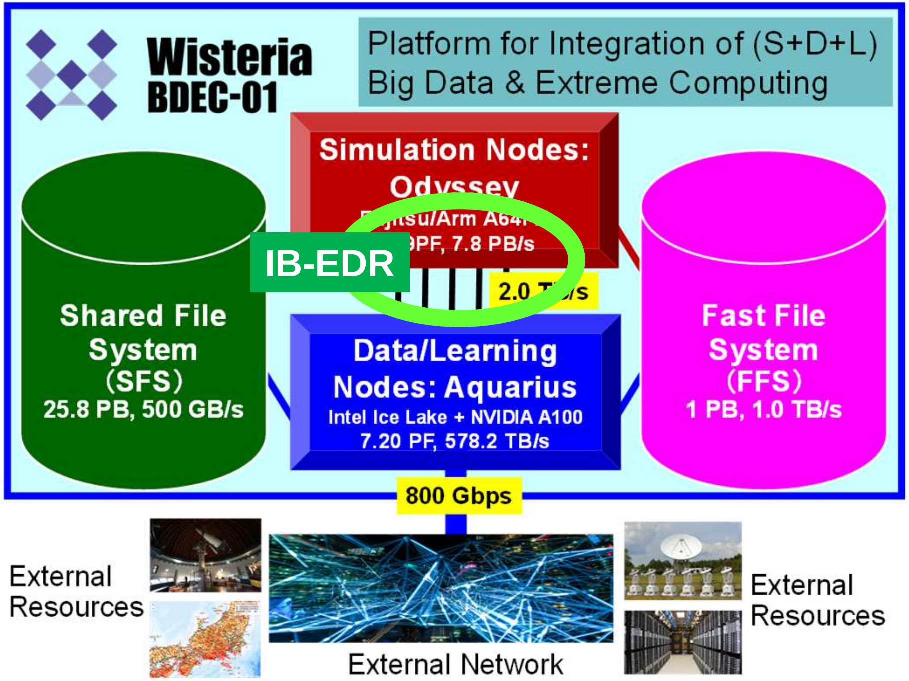
h3-Open-SYS/WaiIO (WaitIO)

- Wisteria/BDEC-01
  - Aquarius (GPU: NVIDIA A100)
  - Odyssey (CPU: A64FX)
- Combining Odyssey-Aquarius
  - Single MPI Job over O-A is impossible
- Connection between O-A
  - IB-EDR & Fast File System (FFS)
  - h3-Open-SYS/WaitIO (WaitIO)
    - Library for Inter-Process Communication over IB-EDR with MPI-like interface
    - Utilizing primitive IB library



h3-Open-SYS/WaiIO (WaitIO)

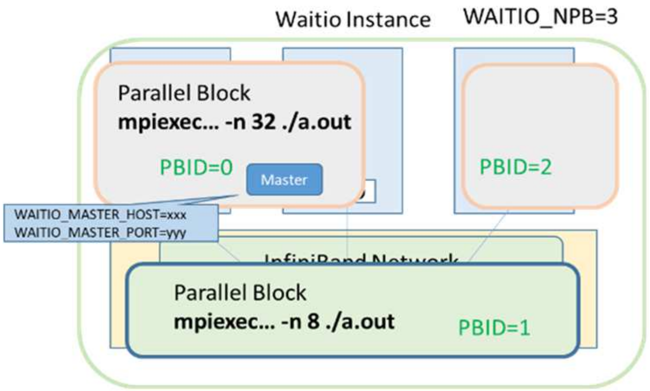
- Wisteria/BDEC-01
  - Aquarius (GPU: NVIDIA A100)
  - Odyssey (CPU: A64FX)
- Combining Odyssey-Aquarius
  - Single MPI Job over O-A is impossible
- **Connection between O-A**
  - **IB-EDR & Fast File System (FFS)**
  - **h3-Open-SYS/WaitIO (WaitIO)**
    - **Library for Inter-Process Communication over IB-EDR with MPI-like interface**
    - **Utilizing primitive IB library**
  - **WaitIO-Socket via IB-EDR**
  - **WaitIO-File via FFS**
  - **WaitIO-Hybrid: automatic switching between IB-EDR and FFS**



h3-Open-SYS/WaitIO（WaitIO）

Communication between Odyssey and Aquarius via IB-EDR (2.0TB/sec)/FFS (1.0TB/sec) is realized using an MPI-like interface. WaitIO can be called from Fortran/C/C++/Python/Julia

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



[Sumimoto et al. 2021]

Seism3D/OpenSWPC-DAF

Starting from (A+S: Data Assimilation+Simulation) to (Pure S: Pure Simulation)

Assim. Comp. Obs. Residual Comp.

$$x_n^a = x_n^f + W(y_n - Hx_n^f)$$

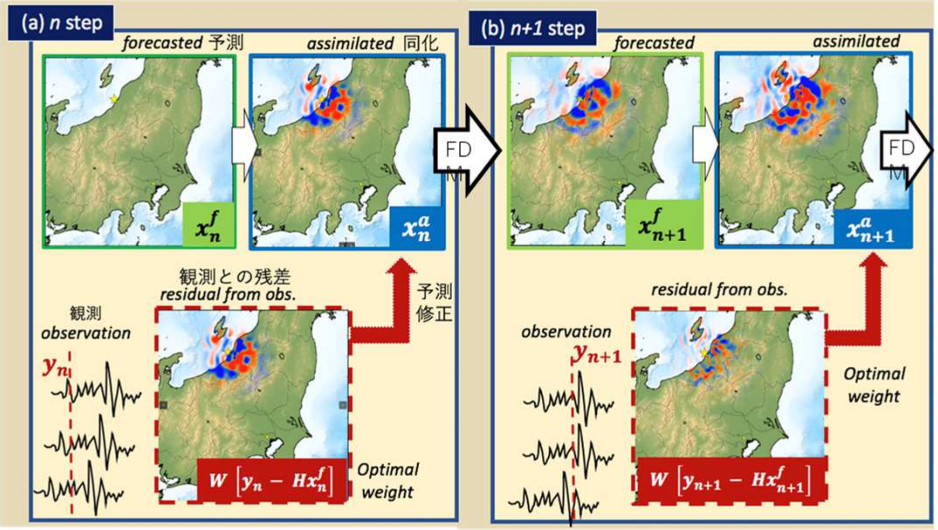
Comp. Assim.

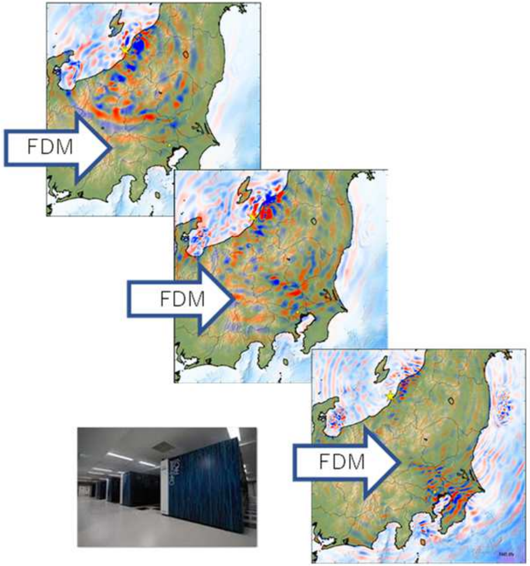
$$x_{n+1}^f = Fx_n^a$$

n: Time Step
W: **Weighting Matrix**
F: Wave Propagation simulation

(A+S) Assimilation+Simulation

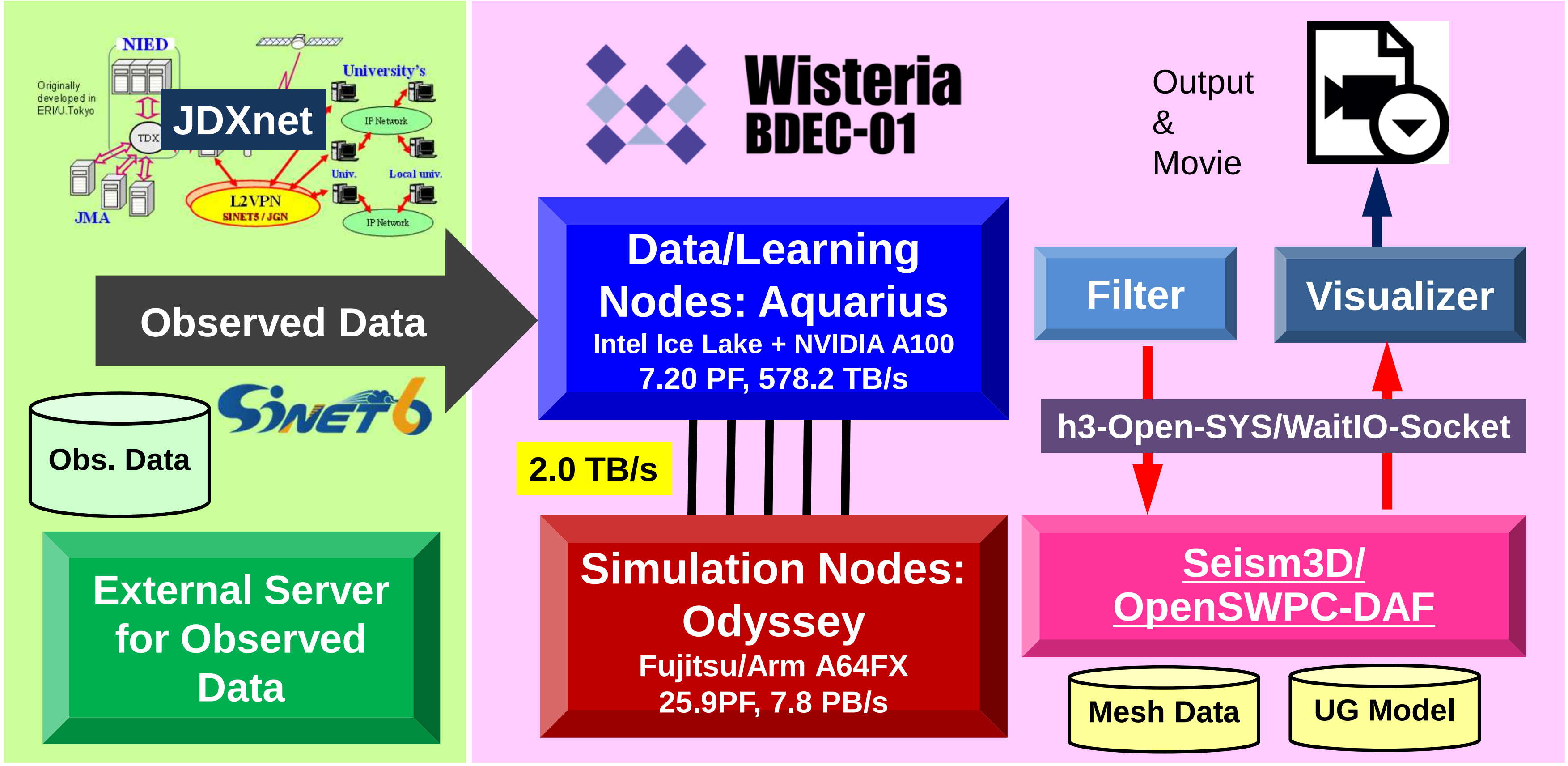
(Pure S) Pure Simulation/Forecast





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

3D Earthquake Simulation with Real-Time Data Observation/Assimilation on Wisteria/BDEC-01



Communications by WaitIO-Socket

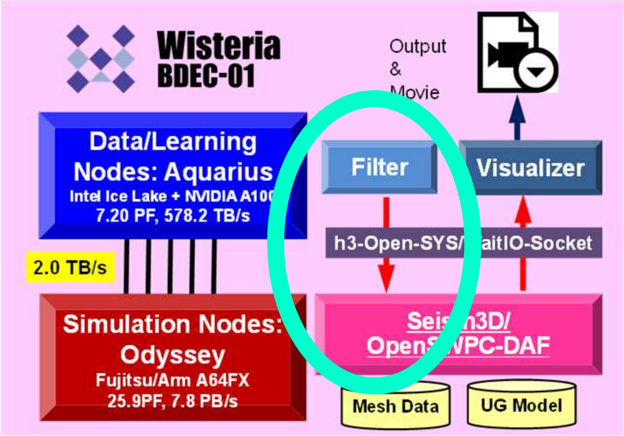
[Kasai et al. 2021]

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_FLOAT,    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_CHAR,     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
```

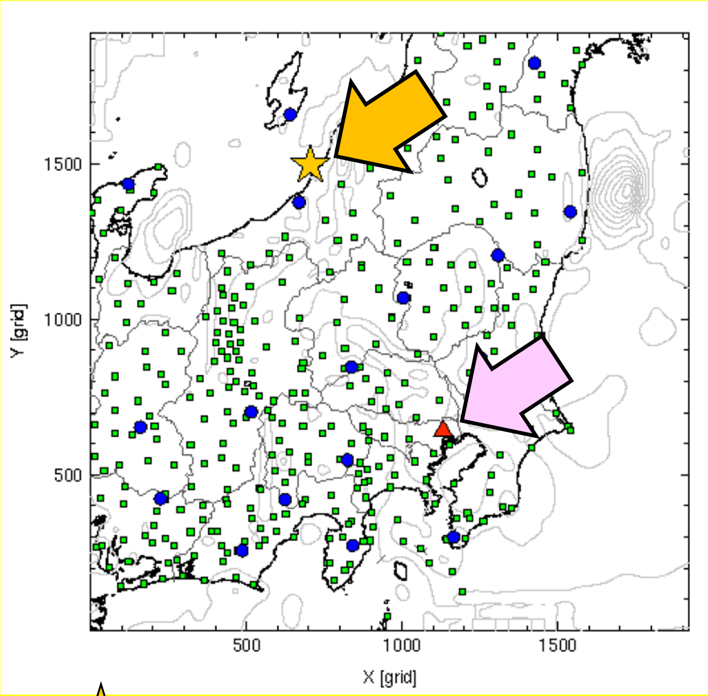
Odyssey: RECV

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



Off Niigata 2007 Mw6.6 Earthquake

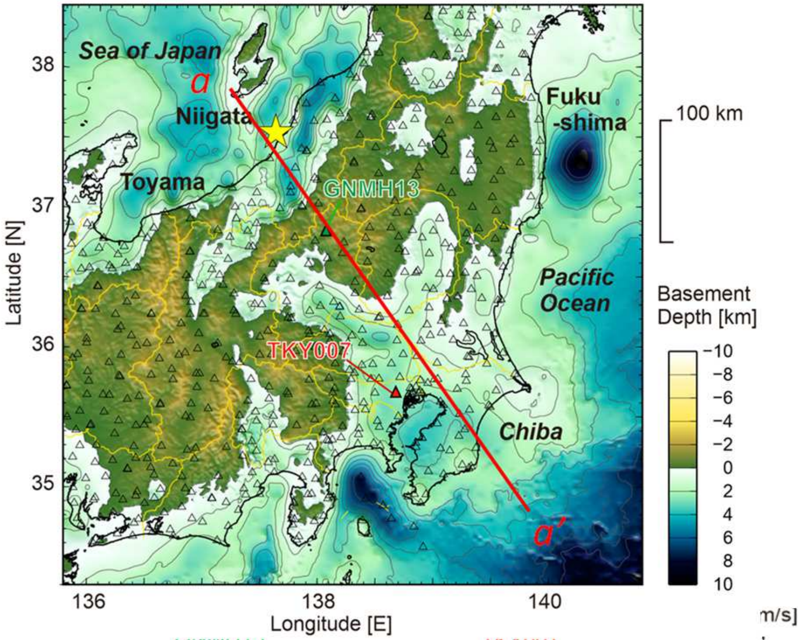
[c/o Prof. T. Furumura, ERI/U.Tokyo]

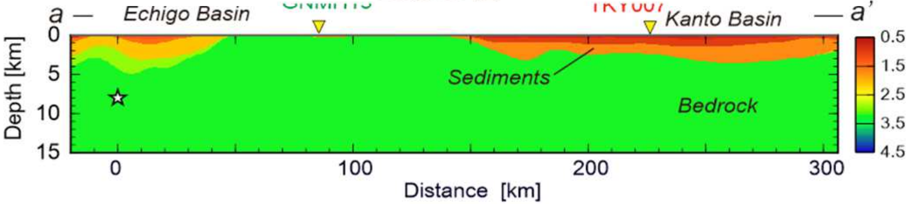


Epicenter

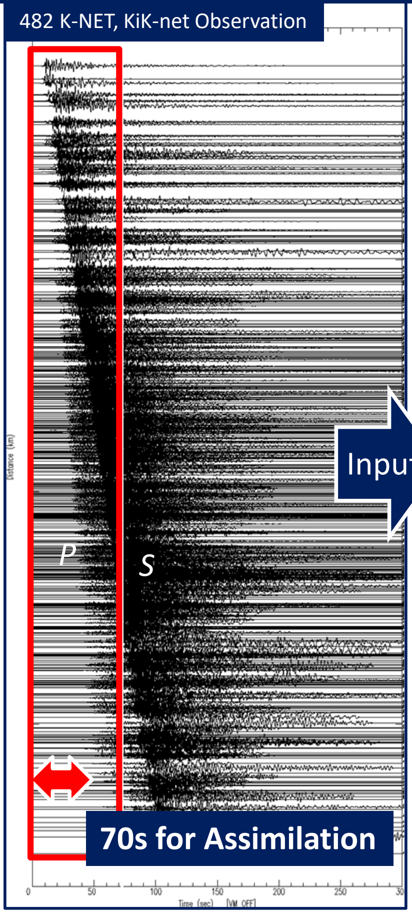
Hi-net (Short Period) 349 pts

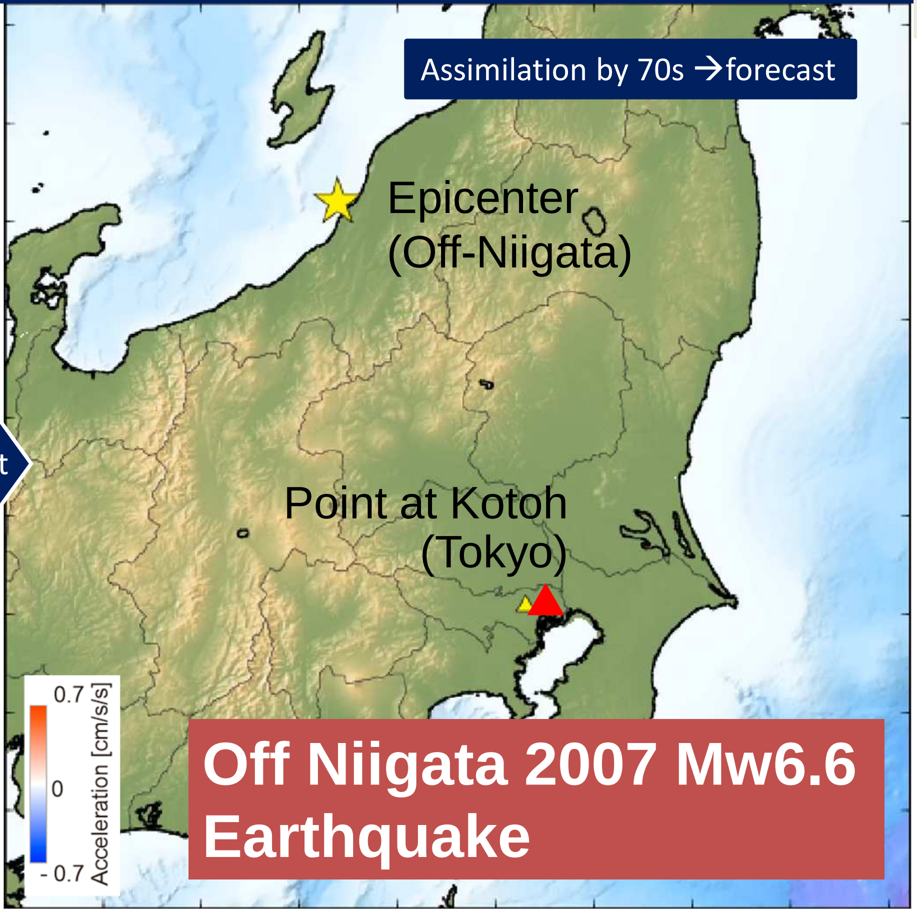
F-net (Broadband) 18 pts



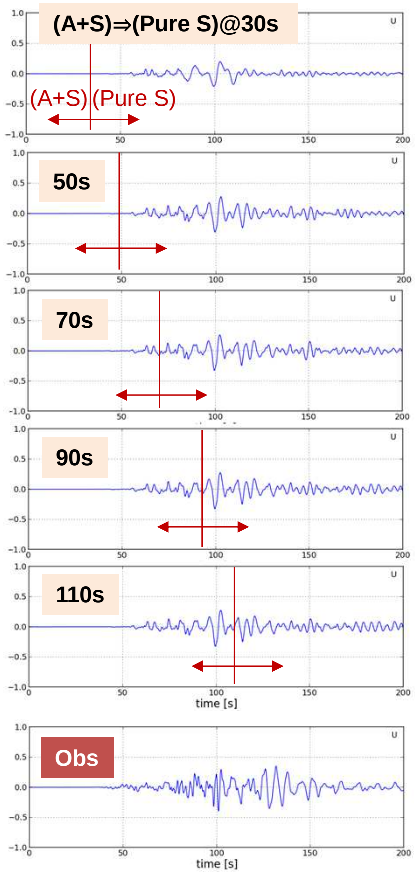


Data Assimilation + Pure Simulation/Forecast





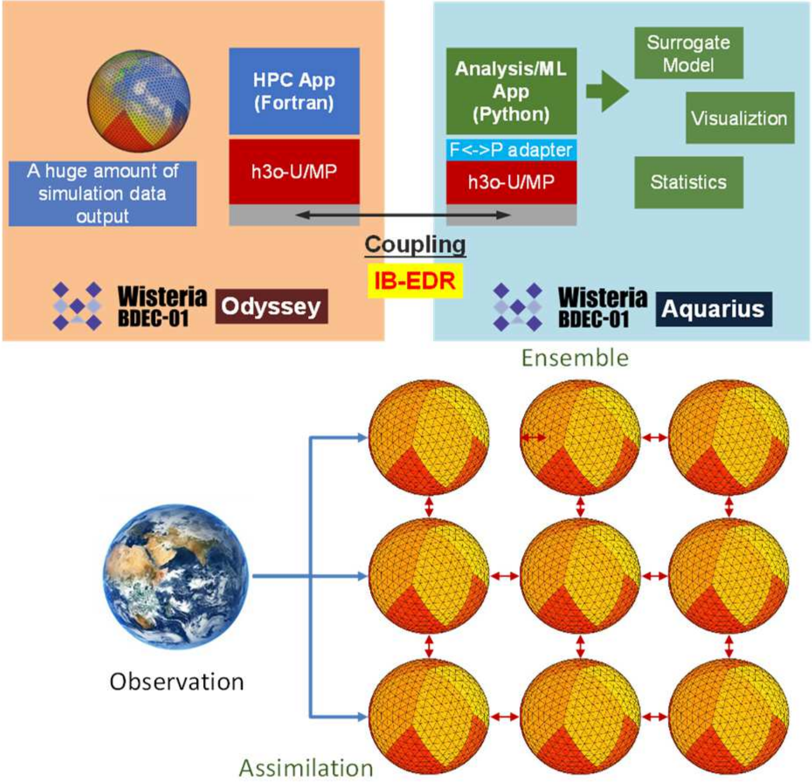
Results at Kotoh ▲ (N.KOTH)
N 35° 37.0'
E 139° 46.9'



h3-Open-UTIL/MP

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

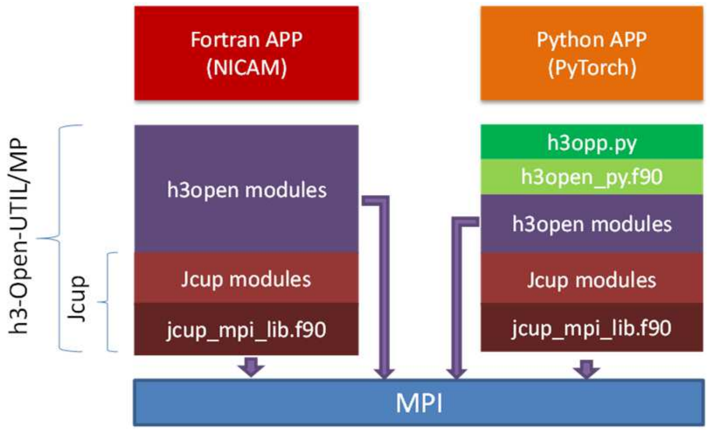
- Extended Version of Multi-Physics Coupler
- Coupling of Simulations on Odyssey and AI on Aquarius
  - Combined with WaitIO
- Special Capabilities for Data Assimilation and Ensemble Simulations
  - Data Assimilation by Coupled Codes
    - e.g. Atmosphere-Ocean
  - Ensemble Coupling



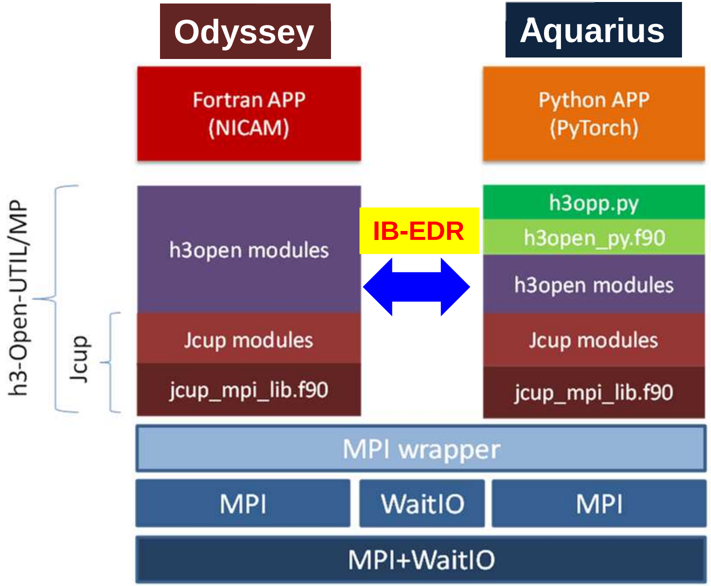
h3-Open-UTIL/MP + h3-Open-SYS/WaitIO-Socket

Available in June 2022

Wisteria BDEC-01



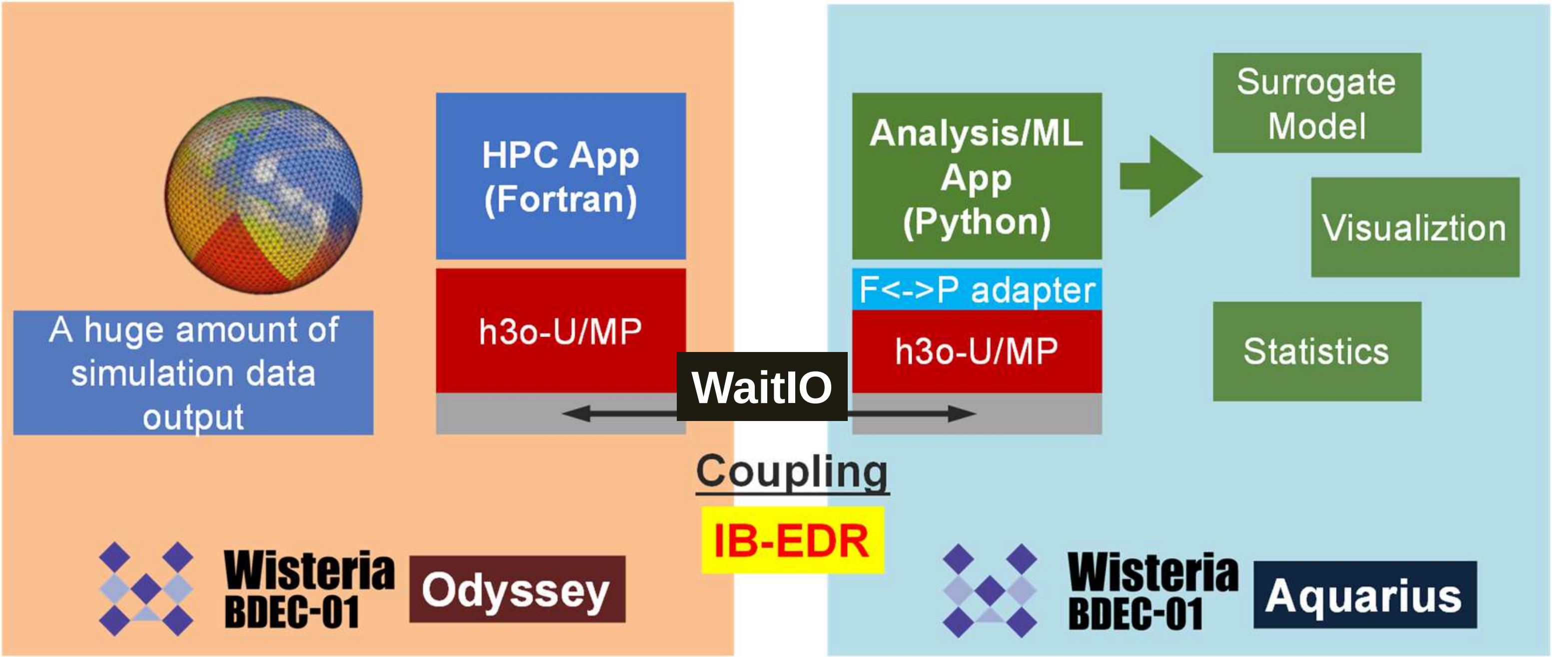
May 2021: MPI Only



June 2022: Coupler＋WaitIO

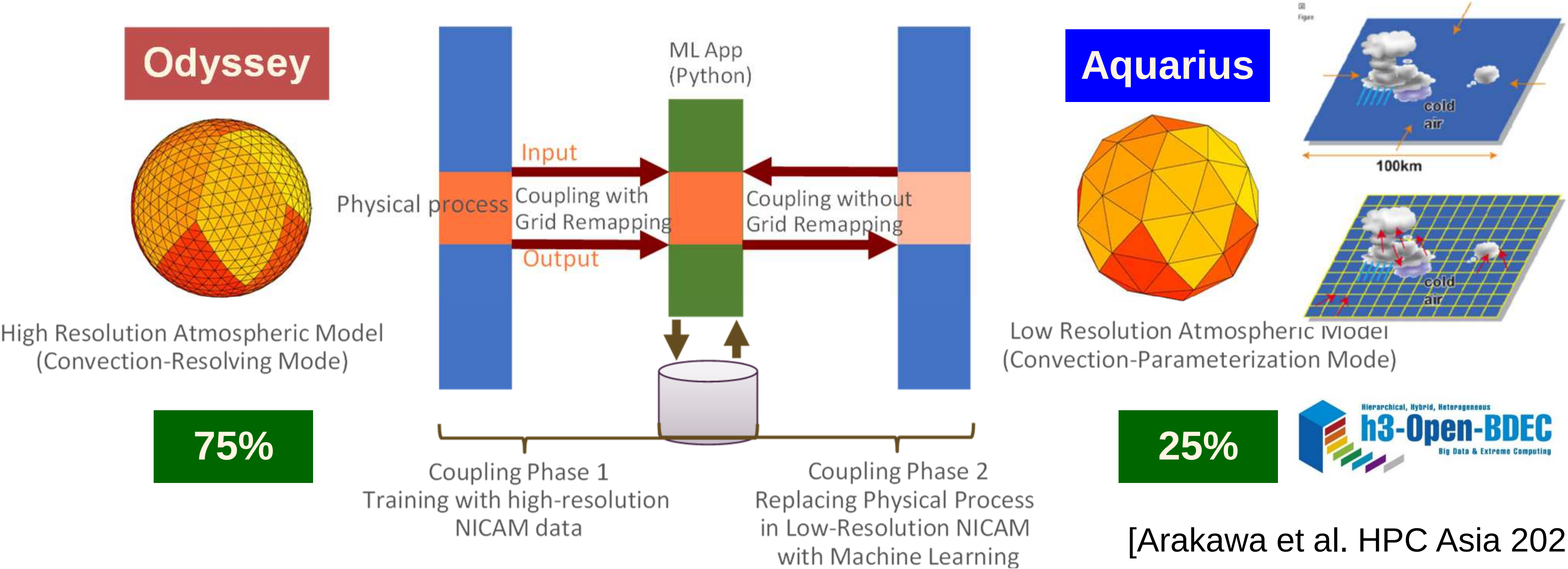
h3-Open-UTIL/MP

Extended Version of Multy-Physics Coupler
Coupling of Simulations on Odyssey and AI on Aquarius
Combined with WaitIO Library



Global Cloud Physics (1/2)

High-Resolution Model/Low-Resolution Model for Parameterization
Replace Low-Resolution Model Simulations (25%) with AI (trained by results of High-Resolution Model)



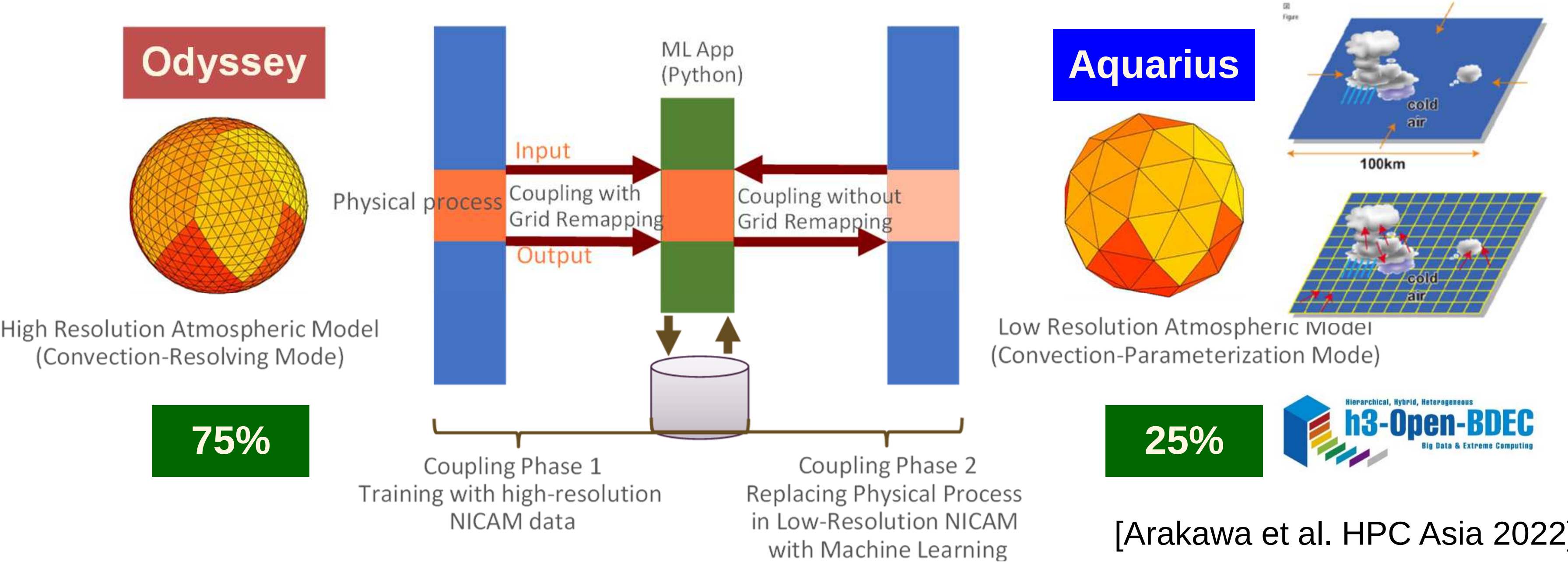
[Arakawa et al. HPC Asia 2022]

Global Cloud Physics (2/2)

High-Resolution Model: NICAM in Fortran on Odyssey

Training/Inferencing: (PyTorch: 3-Layer Perceptron) on Aquarius

Seamless Operations on Odyssey/Aquarius by h3-Open-BDEC: Training of AI by Simulation, and Integration of (Simulation+Inferencing)



[Arakawa et al. HPC Asia 2022]

Comparison: Simulations and AI

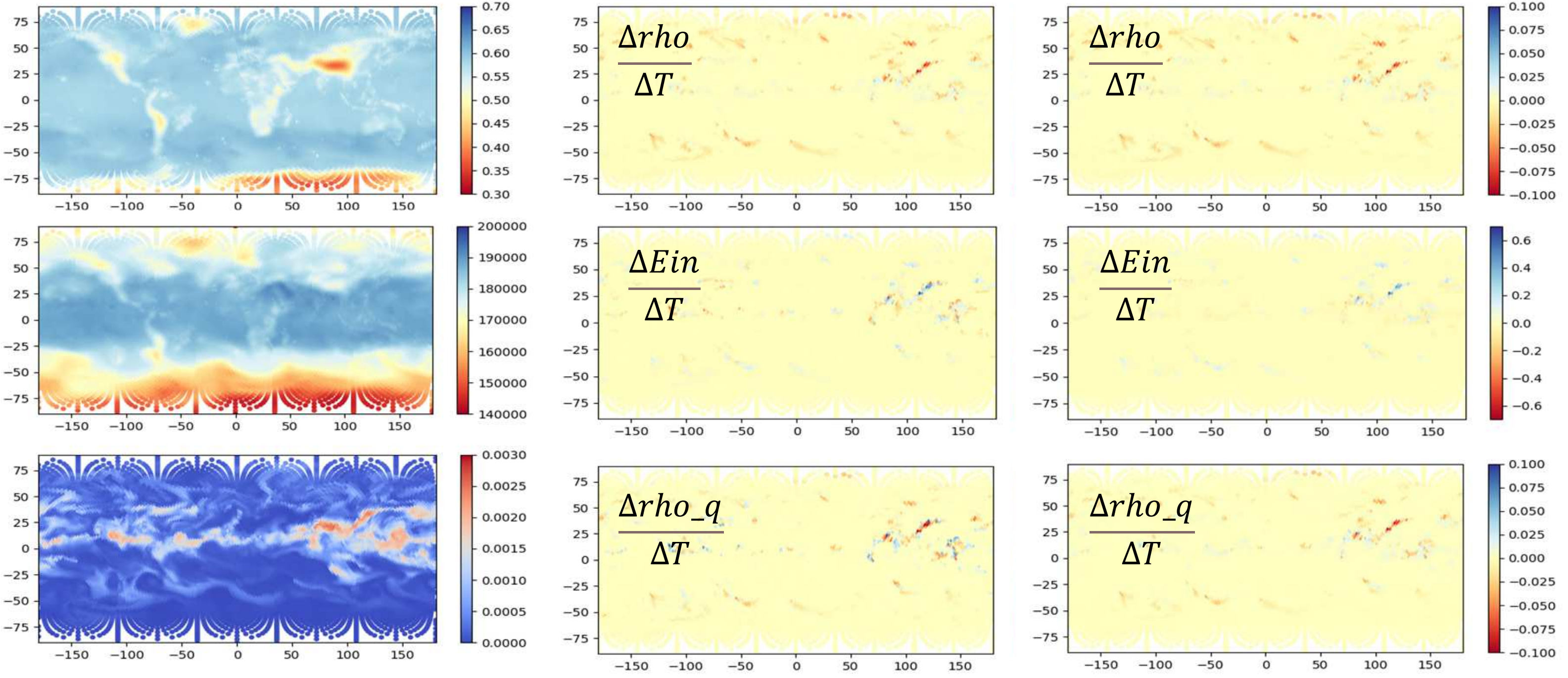
Input: Air Density, Internal Energy of Air, Vapor Density

Output: Gradient in Time Direction of 3 Parameters [Arakawa et al. HPC Asia 2022]

Air Density ***rho***

Internal Energy ***Ein***

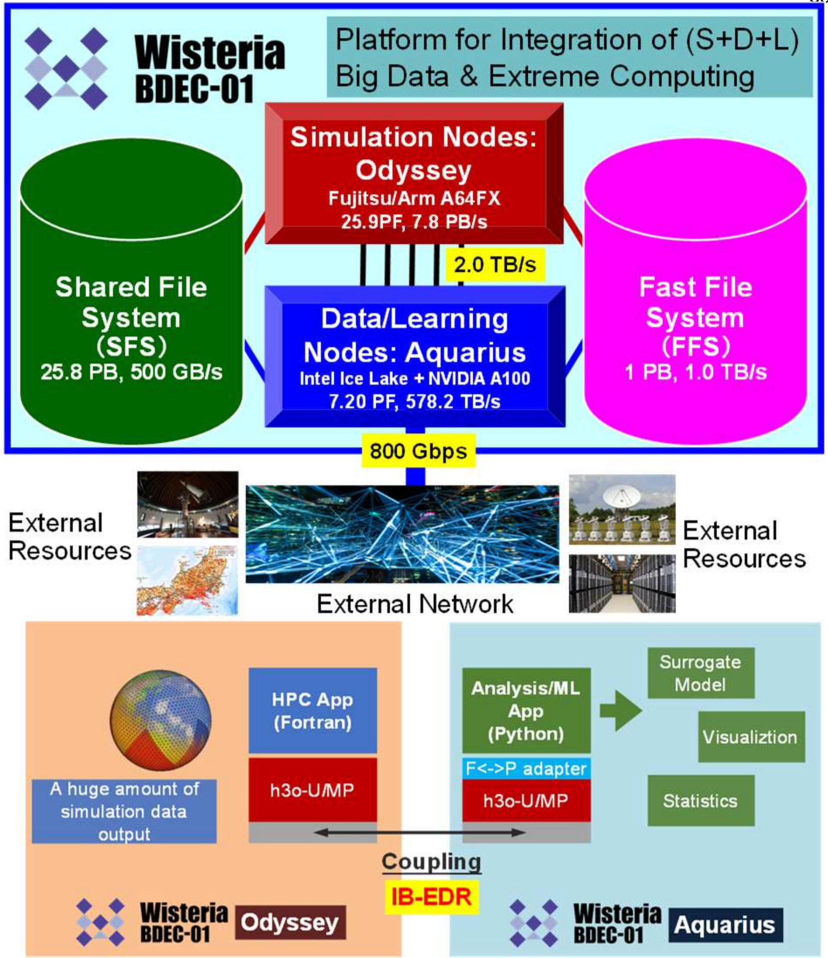
Vapor Density ***rho_q***



Input **Simulations** **Prediction by ML/NN**

How to run the workloads

- Total Number of Nodes
  - Odyssey: 7,680 nodes: not so crowded
  - Aquarius: 45 nodes, 360 GPUs, very crowded
- One node of Aquarius is reserved for this type of workload on the integration of (S+D+L)
- 2 separate jobs (Odyssey, Aquarius) should be submitted
- If both jobs “grab” resources, execution starts.
- More flexible (& complicated) policy needed



Examples of Scripts [Sumimoto, Arakawa]

Odyssey for Simulation

```
#!/bin/bash
#PJM -N "test_waitio"
#PJM -L rscgrp=coupler-lec-o
#PJM -L node=10:noncont
#PJM --mpi proc=80
#PJM -L elapse=00:10:00
#PJM -g gt00
#PJM -j
#PJM -e err

module load fj
module load fjmpi
module load waitio

export WAITIO_MASTER_HOST=`hostname`
export WAITIO_MASTER_PORT=7100
export WAITIO_PBID=0
export WAITIO_NPB=2

hostname
waitio-serv-a64fx -d -m $WAITIO_MASTER_HOST

#mpiexec -oferr-proc errnicam -np 160 ./nicam
mpiexec -np 80 ./nicam
```

Aquarius for AI

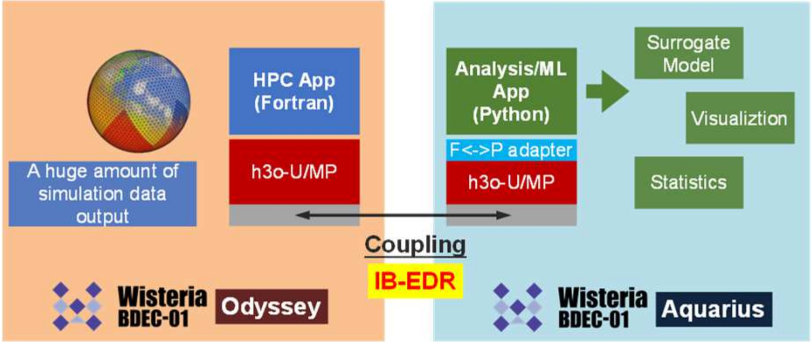
```
#!/bin/bash
#PJM -N "test_waitio"
#PJM -L rscgrp=coupler-lec-a
#PJM -L node=1
#PJM --mpi proc=10
#PJM -L elapse=00:10:00
#PJM -g gt00
#PJM -j
#PJM -e err

module unload aquarius
module unload gcc ompi
module load intel
module load impi
module load waitio

export WAITIO_MASTER_HOST=`waitio-serv -c`
export WAITIO_MASTER_PORT=7100
export WAITIO_PBID=1
export WAITIO_NPB=2

module unload intel
module unload impi
module load gcc ompi

mpiexec -n 10 ./ada
```

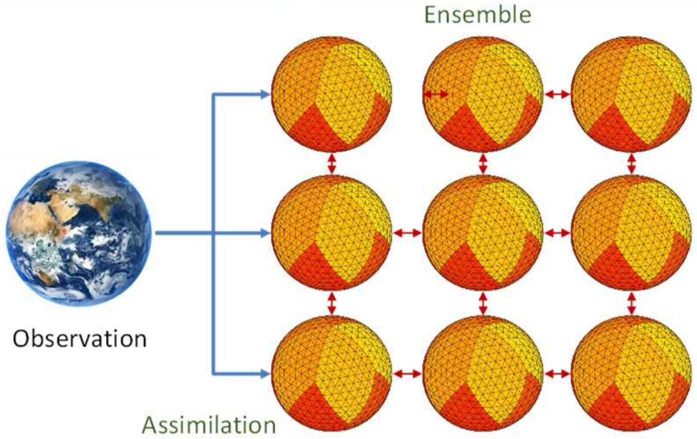
h3-Open-UTIL/MP

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

Hierarchical, Hybrid, Heterogeneous
h3-Open-BDEC
Big Data & Extreme Computing

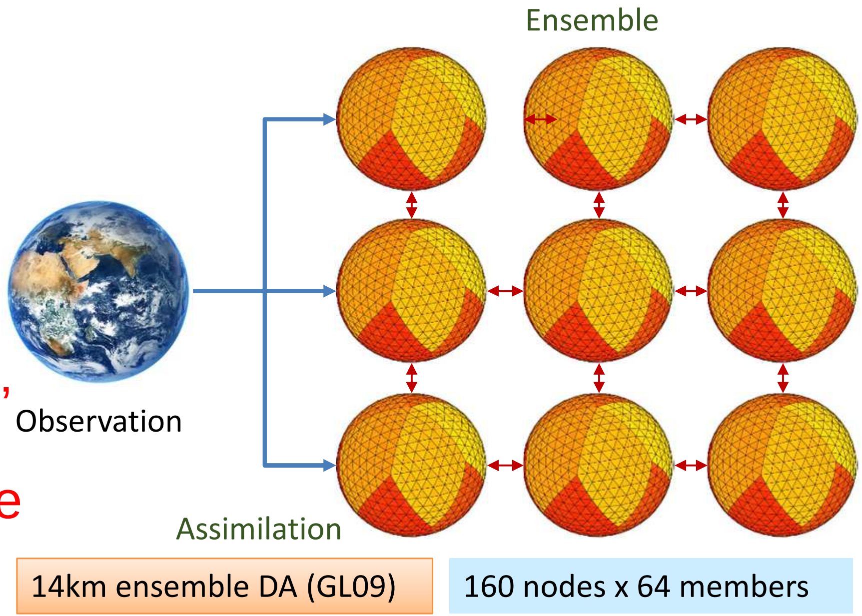
- Extended Version of Multi-Physics Coupler
- Coupling of Simulations on Odyssey and AI on Aquarius
  - Combined with WaitIO
- Special Capabilities for Data Assimilation and Ensemble Simulations
  - Data Assimilation by Coupled Codes
    - e.g. Atmosphere-Ocean
  - Ensemble Coupling





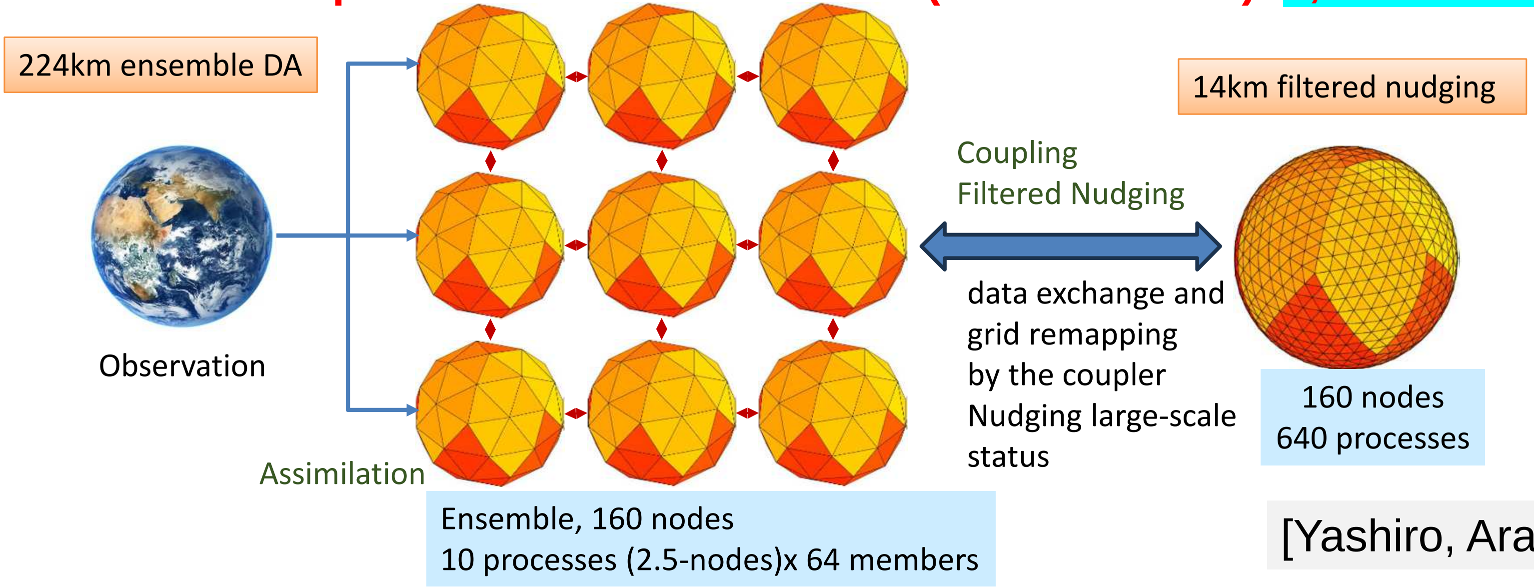
Ensemble-based Data Assimilation

- **Ensemble-based data assimilation** (in global atmospheric simulation for climate/weather prediction) combines data assimilation and ensemble simulations for accurate predictions, demanding significant computational resources for high-resolution simulations.
- 160-nodes of Odyssey are needed for running Global Atmospheric Simulation by NICAM with 14km meshes
- If we do 64 ensembles for 9-Hour Ensemble-based Data Assimilation, we need 2,240 Node-Hours (NH), using 160-nodes for each ensemble
  - ✓ 787.5 sec for each ensemble



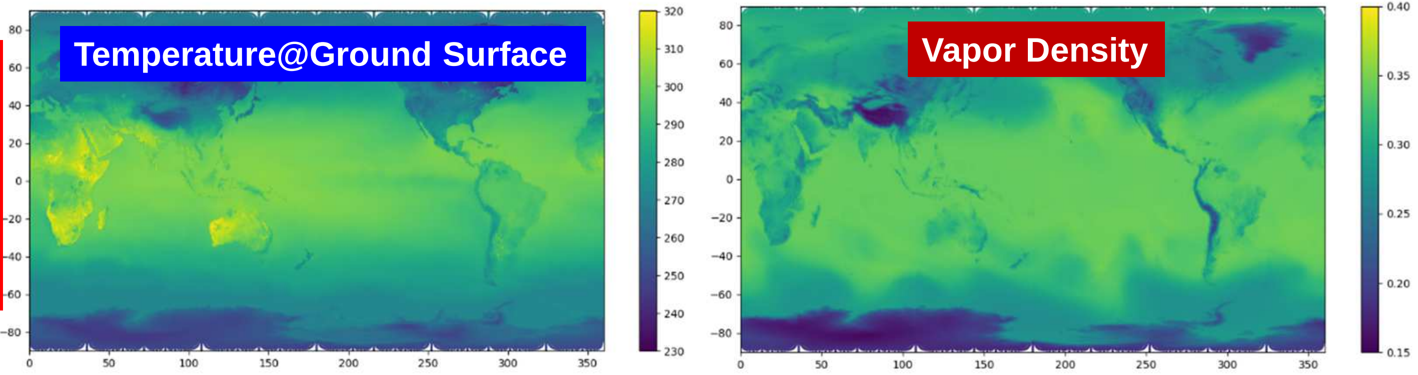
Ensemble Coupling: Ensemble+Coupling

- **Coupling low-resolution ensemble data assimilation with a high-resolution simulation ⇒ reducing resource requirements.**
- Ensemble Coupling: 9-Hour by NICAM on 320 nodes of Odyssey.
  - 160-nodes for low resolution (224km), 160-nodes for high-resolution (14km), coupled by "Filtered Nudging"
  - **Performance Improvement with 100+ times (with FP64⇒32): 2,240 NH⇒19.3 NH**

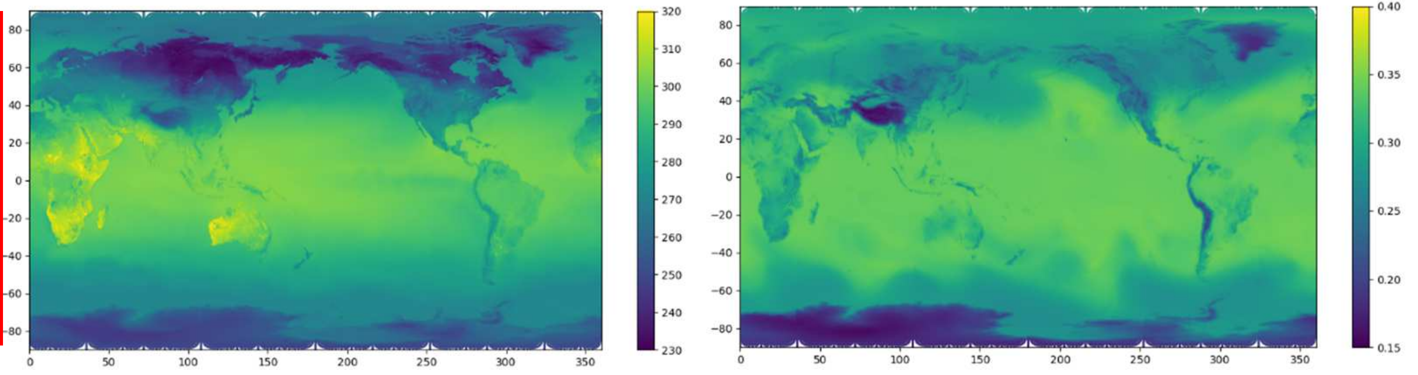


[Yashiro, Arakawa]

Data Assimilation 14km 2,240 NH



Ensemble-Coupling 224km+ 14km 19.3 NH



- Preliminary Results for 9-Hour Integration
- Detailed verification of reproducibility requires integration over at least 2 weeks and meteorological analyses.

[Yashiro, Arakawa]

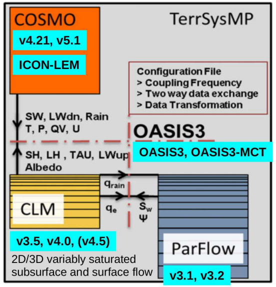
FY.2023-2025, JHPCN Project

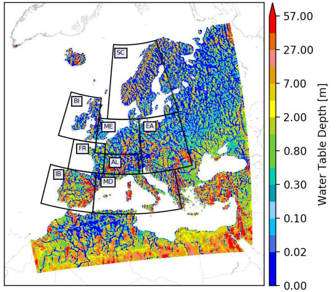
Innovative Computational Science by Integration of Simulation/Data/Learning on Heterogeneous Supercomputers

- ✓ 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)
- ✓ Technical University of Munich (TUM)
- ✓ University of Cologne

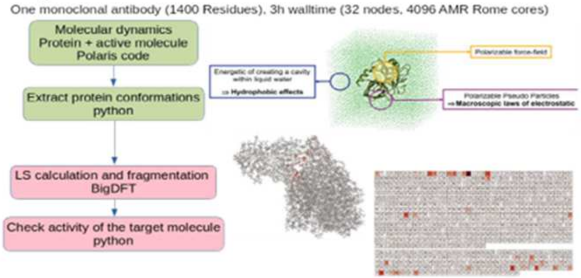
2023：35
2024：49
2025：57

Target Applications





- JSC
  - 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/
- CEA
  - 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 nex, approx and optim flags, vector matrix $\hat{V} \equiv [\hat{v}_1 \ldots \hat{v}_{nev+nex}]$ and estimates μ_1 and $\mu_{nev+nex}$.

Ensure: nev extremal eigenpairs $(\Lambda, \hat{Y})$, with $\Lambda = [\lambda_1 \ldots \lambda_{nev}]$ and $\hat{Y} \equiv [\hat{y}_1 \ldots \hat{y}_{nev}]$, and their residuals $[\mathrm{Res}(\hat{y}_1, \lambda_1) \ldots \mathrm{Res}(\hat{y}_{nev}, \lambda_{nev})]$

$m_{1:nev+nex} \leftarrow \deg$ ▷ INITIAL CONSTANT DEGREE
$(b_{sup}, \mu_1, \mu_{nev+nex}, \hat{V}) \leftarrow \mathrm{LANCZOS}(A, \text{approx})$ ▷ BOUND AND OPTIONAL INPUT

while $\mathrm{size}(\hat{Y}) < nev$ **do**
$\hat{V} \leftarrow \mathrm{FILTER}(A, b_{sup}, \mu_1, \mu_{nev+nex}, \hat{V}, m, \text{optim})$ ▷ USES ARRAY OF DEGREES
$\hat{Q} \leftarrow \mathrm{ORTHOGONALIZE}([\hat{Y}\ \hat{V}])$ ▷ QR FACTORIZATION
$\hat{Q} \leftarrow [\hat{Q}_{:,\mathrm{size}(\hat{Y})+1} \cdots \hat{Q}_{:,nev+nex}]$ ▷ REDUCE TO ACTIVE SUBSPACE
$(\hat{V}, \hat{\Lambda}) \leftarrow \mathrm{RAYLEIGH\text{-}RITZ}(A, \hat{Q})$
Compute the residuals $\mathrm{Res}(\hat{V}, \hat{\Lambda})$
$(\hat{V}, \Lambda, \hat{Y}) \leftarrow \mathrm{DEFLATION}\ \&\ \mathrm{LOCKING}(\hat{V}, \hat{\Lambda}, \mathrm{Res}(\hat{V}, \hat{\Lambda}), \hat{Y})$
$\mu_1 \leftarrow \min([\Lambda\ \hat{\Lambda}])$; $\mu_{nev+nex} \leftarrow \max([\Lambda\ \hat{\Lambda}])$
$c \leftarrow \frac{b_{sup}+\mu_{nev+nex}}{2}$; $e \leftarrow \frac{b_{sup}-\mu_{nev+nex}}{2}$
for $a = 1 \to \mathrm{size}(\hat{V})$ **do**
$m_a \leftarrow \mathrm{DEGREES}(\text{tol}, \mathrm{Res}(\hat{V}_{:,a}, \hat{\lambda}_a), \lambda_a, c, e)$ ▷ COMPUTE POLYNOMIAL DEGREE
end for
Sort $\mathrm{Res}(\hat{V}, \hat{\Lambda}), \hat{V}, \hat{\Lambda}, m$ according to m
end while

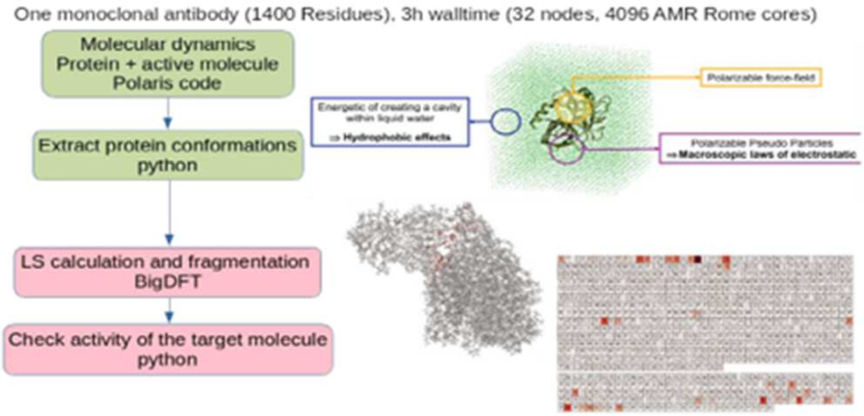
Big-DFT with GENESIS for SARS-CoV-2 Main Protease (CEA, RIKEN, U.Tokyo) (1/2)

- Developing medicines for viruses like SARS-CoV-2 faces challenges, including drug resistance (SARS-CoV-2: Virus, COVID-19: Infection)
  - In order to understand and predict drug resistance, we need to utilize long trajectories from classical molecular dynamics (MD/MM) to model structural changes from point mutations.
  - 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**

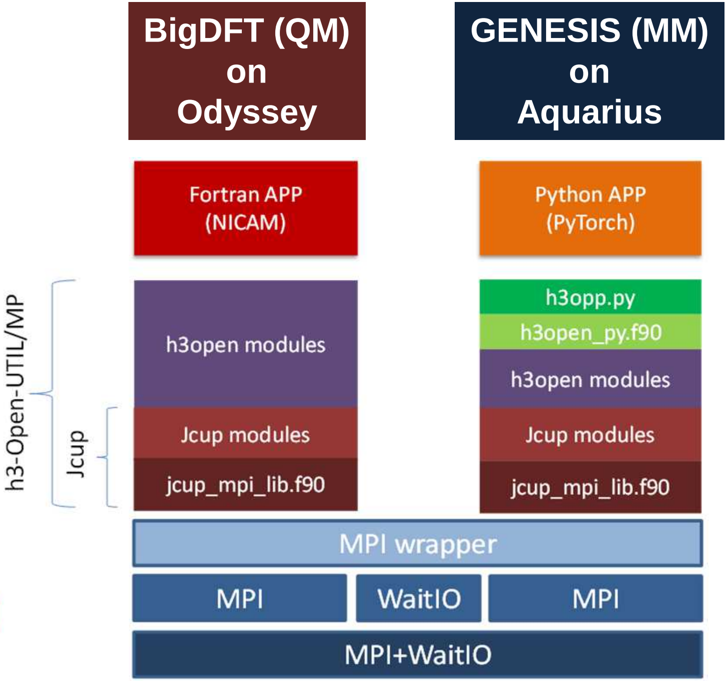
GENESIS (RIKEN)
Polaris (CEA)
(MM/MD)

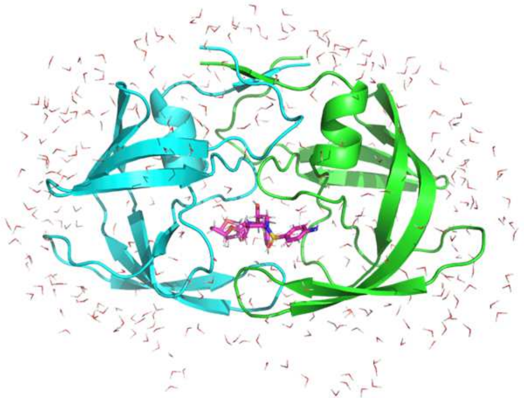
BigDFT (QM)



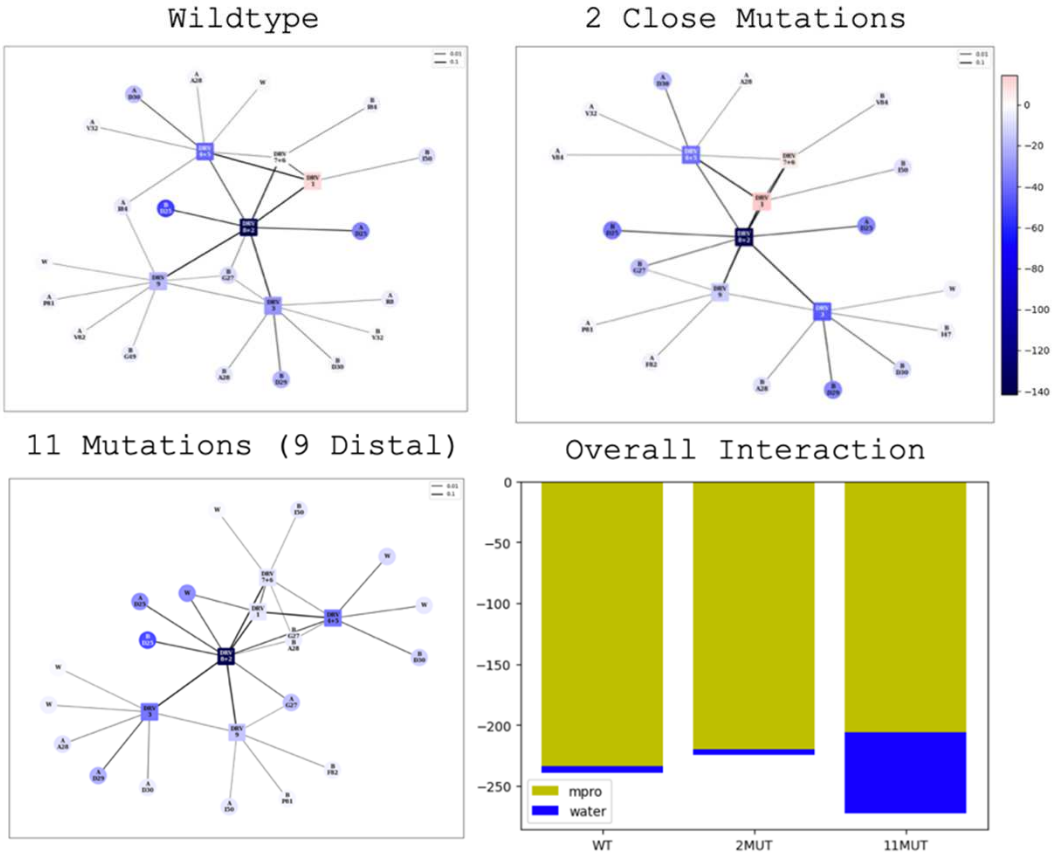
Big-DFT with GENESIS for SARS-CoV-2 Main Protease (CEA, RIKEN, U.Tokyo) (2/2)

- **We 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 constructed 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.**

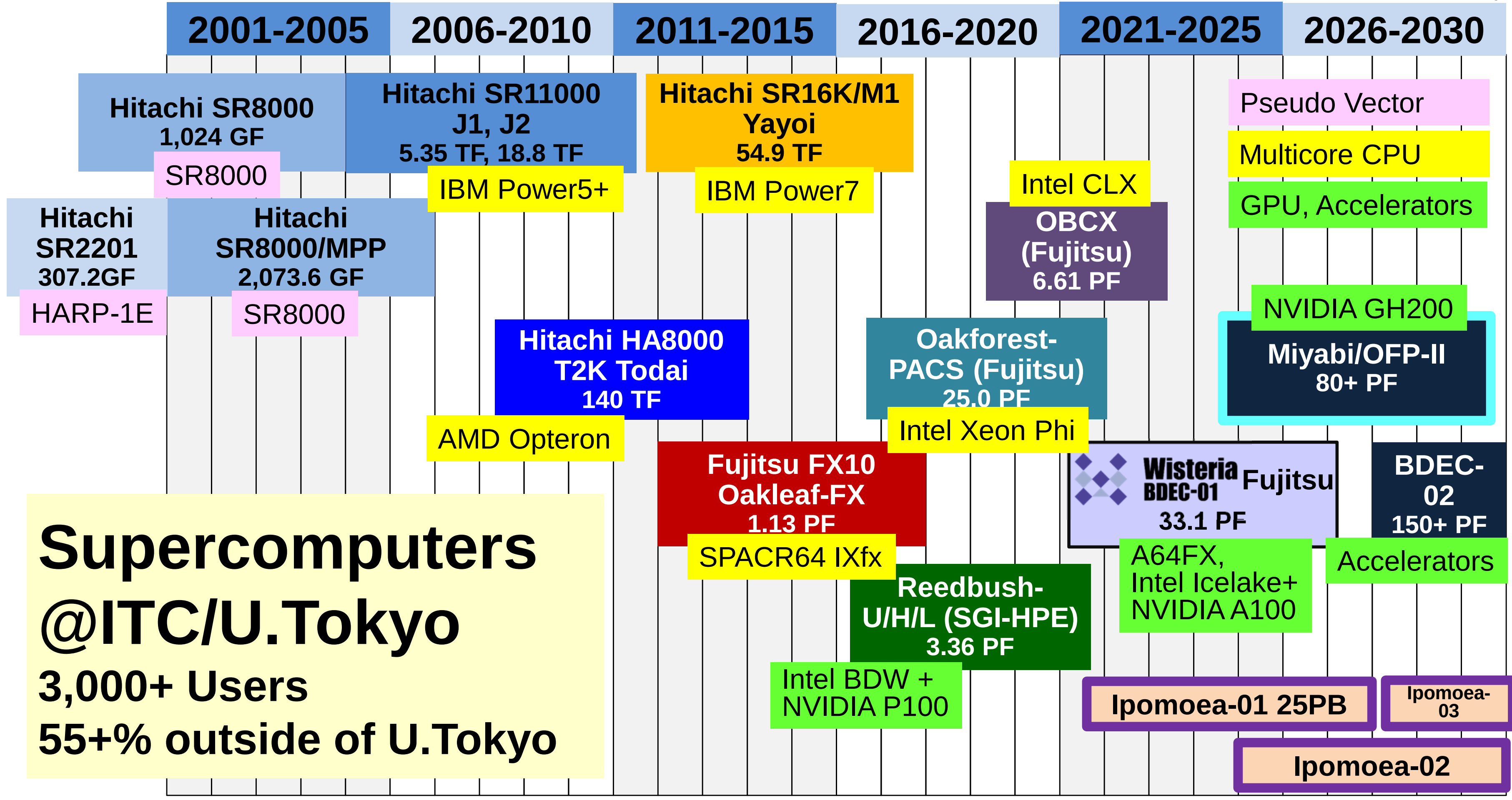




Darunavir bound to HIV-1 main protease



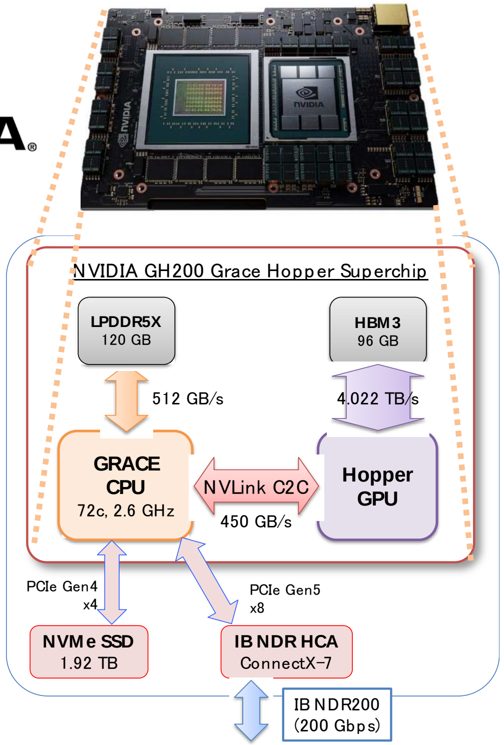
Interaction graphs and overall binding of Darunavir ($C_{27}H_{37}N_3O_7S$) to different variants of the HIV-1 main protease. The interaction network reveals a weakening of the binding as the number of mutations increase. Interaction descriptors are in the unit of kcal / mol.


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
Pseudo Vector
Multicore CPU
GPU, Accelerators
Intel CLX
OBCX
(Fujitsu)
6.61 PF
Hitachi
SR2201
307.2GF
HARP-1E
Hitachi
SR8000/MPP
2,073.6 GF
SR8000
Hitachi HA8000
T2K Todai
140 TF
AMD Opteron
Oakforest-
PACS (Fujitsu)
25.0 PF
Intel Xeon Phi
NVIDIA GH200
Miyabi/OFP-II
80+ PF
Fujitsu FX10
Oakleaf-FX
1.13 PF
SPACR64 IXfx
Wisteria
BDEC-01
Fujitsu
33.1 PF
A64FX,
Intel Icelake+
NVIDIA A100
BDEC-
02
150+ PF
Accelerators
Reedbush-
U/H/L (SGI-HPE)
3.36 PF
Intel BDW +
NVIDIA P100
Ipomoea-01 25PB
Ipomoea-
03
Ipomoea-02
Supercomputers
@ITC/U.Tokyo
3,000+ Users
55+% outside of U.Tokyo

Miyabi (1/2)

Operation starts in January 2025

- **Miyabi-G: CPU+GPU: NVIDIA GH200**
  - Node: NVIDIA GH200 Grace-Hopper Superchip
    - Grace: 72c, 3.456 TF, 120 GB, 512 GB/sec (LPDDR5X)
    - H100: 66.9 TF DP-Tensor Core, 96 GB, 4,022 GB/sec (HBM3)
      - Cache Coherent between CPU-GPU
    - NVMe SSD for each GPU: 1.9TB, 8.0GB/sec, GPUDirect Storage
  - **Total (Aggregated Performance: CPU+GPU)**
    - **1,120 nodes, 78.8 PF, 5.07 PB/sec, IB-NDR 200**
- **Miyabi-C: CPU Only: Intel Xeon Max 9480 (SPR)**
  - Node: Intel Xeon Max 9480 (1.9 GHz, 56c) x 2
    - 6.8 TF, 128 GiB, 3,200 GB/sec (HBM2e only)
  - **Total**
    - **190 nodes, 1.3 PF, IB-NDR 200**
    - **372 TB/sec for STREAM Triad (Peak: 608 TB/sec)**



Miyabi (2/2)

Operation starts in January 2025

- <u>**File System: DDN EXA Scalar, Lustre FS**</u>
  - 11.3 PB (NVMe SSD) 1.0TB/sec, "Ipomoea-01" with 26 PB is also available
- **All nodes are connected with Full Bisection Bandwidth**
  - **(400Gbps/8)×(32×20＋16×1) = 32.8 TB/sec**
- **Operation starts in January 2025, h3-Open-SYS/WaitoIO will be adopted for communication between Miyabi-G and Miyabi-C**

IB-NDR（400Gbps）

IB-NDR200（200）

IB-HDR（200）

Miyabi-G
NVIDIA GH200 1,120
78.2 PF, 5.07 PB/sec

Miyabi-C
Intel Xeon Max
(HBM2e) 2 x 190
1.3 PF, 608 TB/sec

File System
DDN EXA Scaler
11.3 PB, 1.0TB/sec
All Flash Storage

Ipomoea-01
Common Shared Storage
26 PB

miyabi

“Miyabi（雅）” means “Elegance” in Japanese

65th TOP500 List (June, 2025)

R_{max}: Performance of Linpack (TFLOPS)
R_{peak}: Peak Performance (TFLOPS), Power: kW

http://www.top500.org/

	Site	Computer/Year Vendor	Cores	R_{max} (PFLOPS)	R_{peak} (PFLOPS)	GFLOPS/W	Power (kW)
1	**El Capitan, 2024, USA** DOE/NNSA/LLNL	HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS	11,039,616	1,742.00 (=1.742 EF)	2,746.38 63.4 %	58.99	**29,581**
2	**Frontier, 2021, USA** DOE/SC/Oak Ridge National Laboratory	HPE Cray EX235a, AMD Optimized 3rd Gen. EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11	9,066,176	1.353.00	2,055.72 65.8 %	54.98	**24,607**
3	**Aurora, 2023, USA** DOE/SC/Argonne National Laboratory	HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel	9,264,128	1,012.00	1,980.01 51.1 %	26.15	**38,698**
4	**JUPITER Booster, 2025, USA** EuroHPC/FZJ	EVIDEN, BullSequana XH3000, GH Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, RedHat Enterprise Linux	4,801,344	794,40	930.00 85.3 %	60.62	**13,088**
5	**Eagle, 2023, USA** Microsoft	Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR	2,073,600	561.20	846.84 66.3 %		
6	**HPC 6, 2024, Italy** Eni S.p.A.	HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, RHEL 8.9	3,143,520	477.90	606.97 66.3 %	56.48	**8,461**
7	**Fugaku, 2020, Japan** R-CCS, RIKEN	Fujitsu PRIMEHPC FX1000, Fujitsu A64FX 48C 2.2GHz, Tofu-D	7,630,848	442.01	537.21 82.3 %	14.78	**29,899**
8	**Alps, 2024, Switzerland** Swiss Natl. SC Centre (CSCS)	HPE Cray EX254n, NVIDIA Grace 72C 3.1GHz, NVIDIA GH200 Superchip, Slingshot-11	2,121,600	434.90	574.84 75.7 %	61.05	**7,124**
9	**LUMI, 2023, Finland** EuroHPC/CSC	HPE Cray EX235a, AMD Optimized 3rd Gen. EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11	2,752,704	379.70	531.51 71.4 %	53.43	**7,107**
10	**Leonard, 2023, Italy** EuroHPC/Cineca	EVIDEN, BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64GB, Quad-rail NVIDIA HDR100	1,824,768	241.20	181.49 78.7 %	32.19	**7,494**
15	**ABCI 3.0, 2025, Japan** AIST	HPE Cray XD670, Xeon Platinum 8558 48C 2.1GHz, NVIDIA H200 SXM5 141 GB, Infiniband NDR200, Rocky Linux 9	479,232	145.10	130.44 80.0 %	59.29	**3,596**
22	**CHIE-3, 2024, Japan** SoftBank, Corp.	NVIDIA DGX H100, Xeon Platinum 8480C 56C 2GHz, NVIDIA H100, Infiniband NDR400, Ubuntu 22.04.4 LTS	297,840	91.94	138.32 66.5 %		
24	**CHIE-2, 2024, Japan** SoftBank, Corp.	NVIDIA DGX H100, Xeon Platinum 8480C 56C 2GHz, NVIDIA H100, Infiniband NDR400, Ubuntu 22.04.4 LTS	297,840	89.78	138.32 64.9 %		
27	**ABCI-Q, 2025, Japan** AIST	Fujitsu, Supermicro SYS-221GE-TNHT-LCC, Intel Xeon Platinum 8558 48C 2.1GHz, NVIDIA H100 SXM5 80GB, Infiniband NDR, Rocky Linux 9.4	315,210	74.58	99.35 75.1 %	40.67	**1,834**
36	**FPT, 2025, JAPAN** FPT AI Factory	HPE, HGX H200, Xeon Platinum 8558 48C 2.1GHz, NVIDIA H200 SXM5 141 GB, Infiniband NDR400, Ubuntu 22.04.5 LTS	146,304	49.85	67.44 73.9 %		
37	**Miyabi-G, 2024, Japan** JCAHPC	Fujitsu, Supermicro ARS 111GL DNHR LCC, Grace Hopper Superchip 72C 3GHz, Infiniband NDR200, Rocky Linux	80,640	46.80	72.80 64.3 %	47.59	**983**
46	**TSUBAME 4.0, 2024, Japan** Institute of Science Tokyo	HPE Cray XD665, AMD EPYC 9654 96C 2.4GHz, NVIDIA H100 SXM5 94 GB, Infiniband NDR200	172,800	39.62	61.60 64.3 %	48.55	**816**
73	**Wisteria/BDEC-01 (Odyssey), 2021, Japan** U.Tokyo	Fujitsu PRIMEHPC FX1000, A64FX 48C 2.2GHz, Tofu D	368,640	22.12	25.95 85.2 %	15.07	**1,468**

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

Anything is possible with h3-Open-BDEC (WaitIO and Coupler)

CPU		GPU			Others	
A64FX Arm	X86	NVIDIA Intel AMD Arm	AMD	Intel	Sambanova Cerebras Graphcore etc.	Quantum

Hierarchical, Hybrid, Heterogeneous
h3-Open-BDEC
Big Data & Extreme Computing

Q. to MS CoPilot: Are all of supercomputers replaced by quantum computers ?

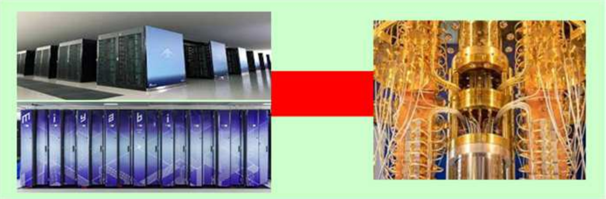
Copilot

- Not quite! While quantum computers are incredibly powerful and excellent at certain types of computations, they haven't replaced supercomputers.
- Instead, they're often seen as complementary technologies.
- Supercomputers are still essential for tasks requiring massive computational capacity and data management, like climate modeling and large-scale data analysis.
- Quantum computers, on the other hand, are better suited for specialized computations that classical supercomputers struggle with, such as optimization problems and crypt-analysis.
- The future of computing is likely to involve a **hybrid approach**, where quantum computers work alongside classical supercomputers to tackle different types of problems more efficiently.

QC-HPC Hybrid Computing

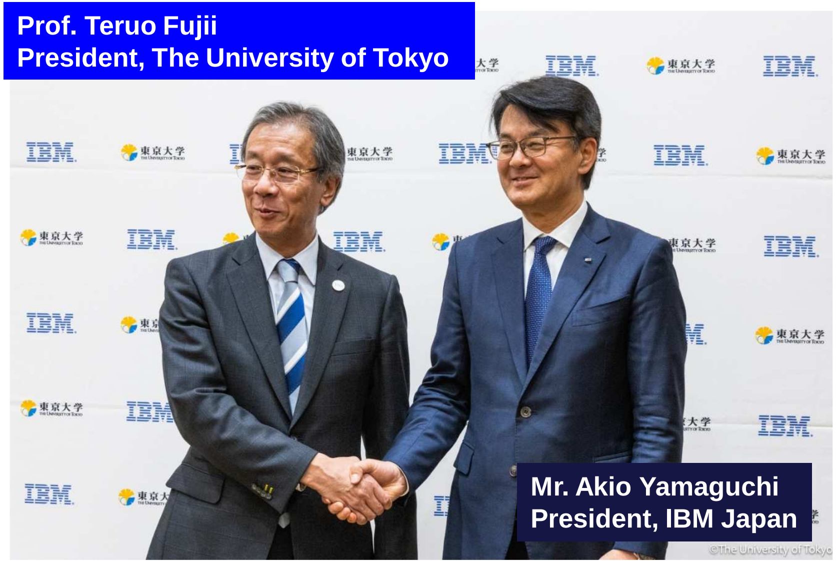
- **JHPC-quantum (FY.2023-FY.2028)**
  - https://jhpc-quantum.org/
  - RIKEN, Softbank, U.Tokyo, Osaka U.
    - supported by Japanese Government (METI/NEDO)
    - This project has a strong focus on industrial applications
  - 2 Real QC's to be installed in FY.2025
    - IBM's Superconducting QC at RIKEN-Kobe (156 Qbit)
    - Quantinuum's Ion-Trap QC at RIKEN-Wako (20+Qbit)
      - Op. started in Feb., 2025: 黎明(Reimei)= "Dawn"
  - Target Applications
    - Quantum Sciences, Quantum Machine Learning etc.
- Role of U.Tokyo
  - R&D on System SW for QC-HPC Hybrid Environment (QC as Accelerators)
    - Extension of h3-Open-BDEC
- **Fugaku (RIKEN) & Miyabi/Wistereia-BDEC01 (U.Tokyo) will be connected to QCs in Fall 2025**

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



University of Tokyo to Equip IBM Quantum System One with Latest IBM Quantum Heron Processor (May 16, 2025) (1/2)

IBM Quantum System One also to be linked with Miyabi supercomputer at the Joint Center for Advanced High Performance Computing (JCAHPC)



- Today, the University of Tokyo (UTokyo) and IBM announced plans to deploy the latest 156-qubit IBM Quantum Heron processor, which will be operational in the IBM Quantum System One, administered by UTokyo for the members of the Quantum Innovation Initiative (QII) Consortium.
- The IBM Heron processor is significantly more powerful than the 127-qubit IBM Quantum Eagle processor previously installed in 2023, with an order of magnitude lower 100-qubit layered error rates.

University of Tokyo to Equip IBM Quantum System One with Latest IBM Quantum Heron Processor (May 16, 2025) (2/2)

IBM Quantum System One also to be linked with Miyabi supercomputer at the Joint Center for Advanced High Performance Computing (JCAHPC)

- UTokyo is planning to link the IBM Quantum System One to the Miyabi supercomputer later this year, through “cloud bursting”, a long-distance networking connection.
- The combined quantum and HPC systems will give the industry and academic users of QII access to new computational capabilities and increasing performance in the domain areas such as chemistry, bioinformatics, high energy physics, materials science, finance and among many other disciplines.
- Miyabi will be connected to three quantum systems this year: IBM Kawasaki, IBM Kobe, and Quantinuum Wako (Reimei).

History of the Projects & Budgets (JPY)

2011-2015	2016-2020	2021-2025	2026-

JHPC-quantum
NEDO
139M (for U.Tokyo)

QC's
IBM
QUANTINUUM

Miyabi
(Fujitsu, GH200/SPR)
80.1 PF

Wisteria/BDEC-01
(Fujitsu, A64FX/NVIDIA A100)
33.1 PF

Development for QC-HPC Hybrid Environment

- Extension of WaitIO Library
  - WaitIO4Py
    - Python Interface
  - WaitIO-Router
    - Communication Library among Supercomputer Centers over SINET

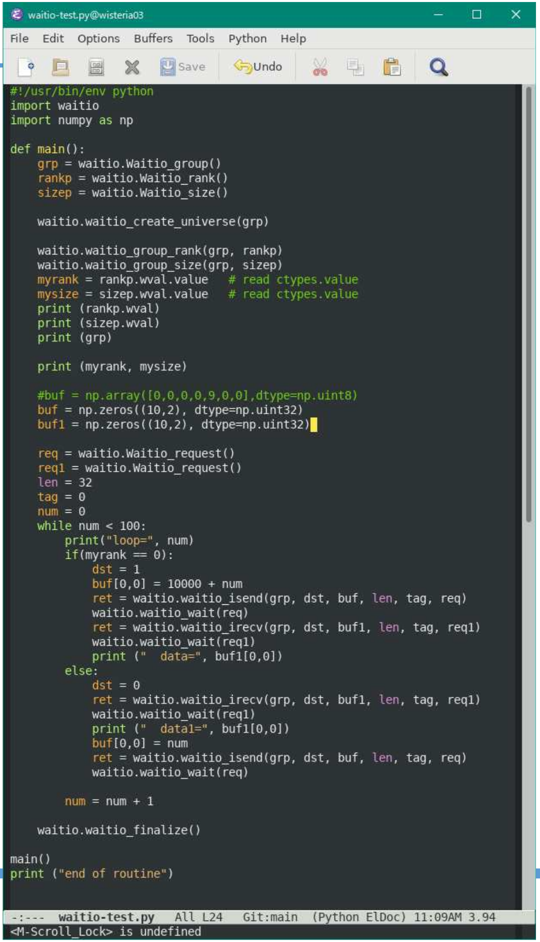
- FY.2024: Simulators on Wisteria/BDEC-01
- FY.2025: Real QC-HPC Environment
  - WaitIO4Julia

WaitIO4Py: Sample Code

- PingPong Program

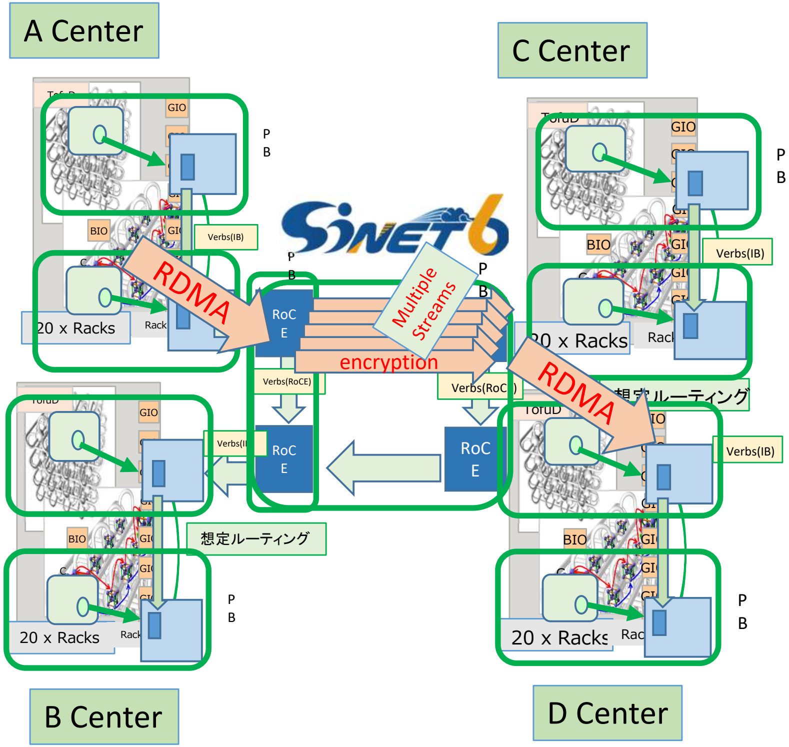
```
import waitio
import numpy as np

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()
```

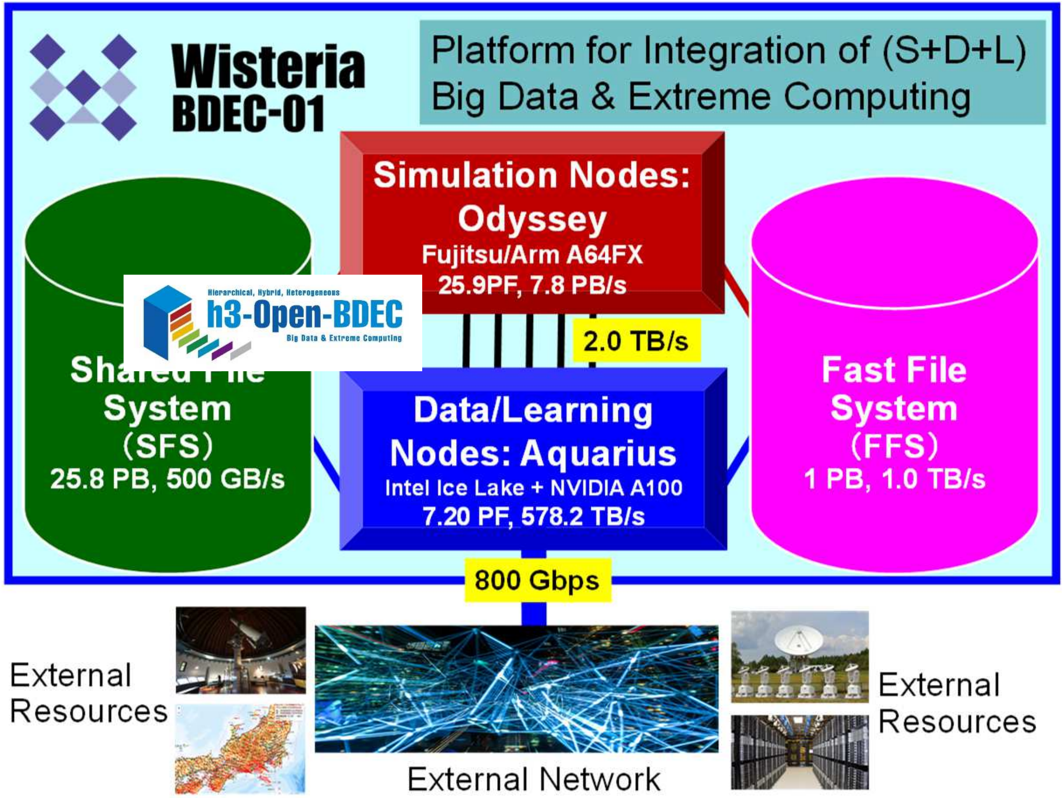


WaitIO-Router for Center-Wide Communication

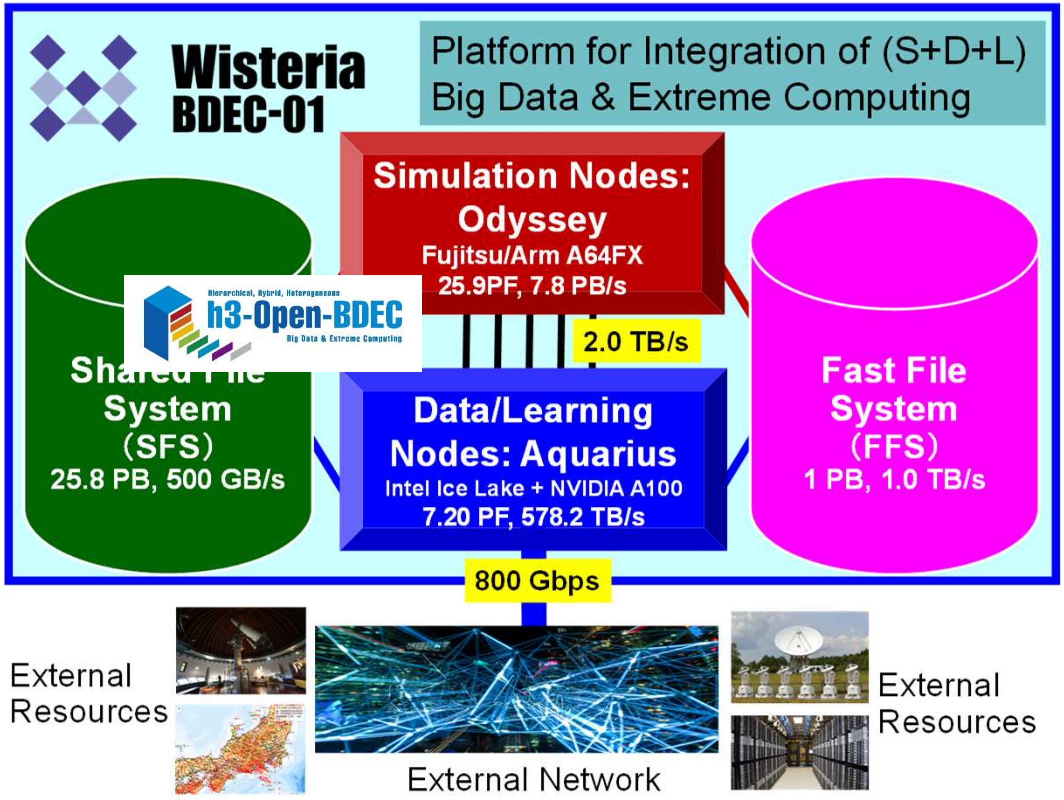
- In a Center: High performance communication using RDMA and Zero-Copy Routing.
  - WaitIO-Verbs and WaitIO-Tofu
  - WaitIO-Router for Zero-Copy Routing
- Among Centers: Converting Single Stream to Multiple Streams
  - For Performance: Multiple RDMA(RoCE) and Socket streams
  - For Security: with Encryption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

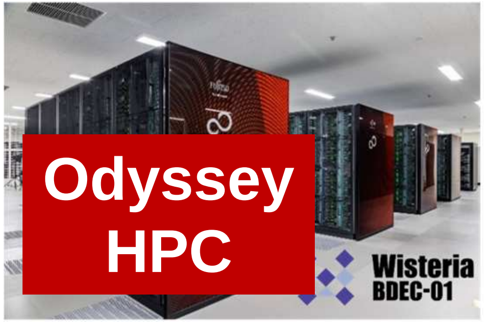


Wisteria/BDEC-01 with Odyssey-Aquarius Simulator of (QC-HPC) Hybrid Environment



Wisteria/BDEC-01 with Odyssey-Aquarius Simulator of (QC-HPC) Hybrid Environment





Wisteria/BDEC-01 with Odyssey-Aquarius Simulator of (QC-HPC) Hybrid Environment

- CUDA-Q
  - Quantum Simulator developed by NVIDIA
    - available on Aquarius
- 1-Day Tutorial on December 18, 2024
  - “QC-HPC Hybrid" with "CUDA-Q+Wisteria/BDEC-01+ h3-Open-BDEC"!
    - VQE (Variational Quantum Eigensolver) written in Python
    - Aquarius (GPU) with CUDA-Q: QC
    - Odyssey (CPU): HPC
    - h3-Open-BDEC: QC-HPC Hybrid by WaitIO for Python
  - World’s 1st Tutorial on “QC-HPC” with hands-on

Wisteria/BDEC-01 with Odyssey-Aquarius Simulator of (QC-HPC) Hybrid Environment

- CUDA-Q
  - Quantum Simulator developed by NVIDIA
    - available on Aquarius
- 1-Day Tutorial on December 18, 2024

10:00-10:10 10:10-10:30	Background Introduction to NVIDIA Quantum
10:30 - 12:00	Introduction to CUDA-Q Login to Wisteria/BDEC-01 Very Simple Examples
13:00 - 14:30	More Advanced Topics VQE
15:00 - 17:00	QC-HPC Hybrid

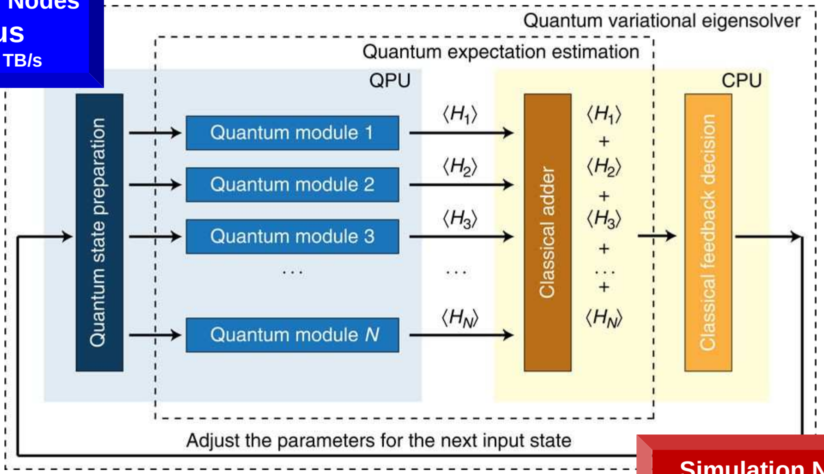
Variational Quantum Eigensolver (VQE)
Ground State (Minimum Energy) = Minimum Eigenvalue

- Compute the minimum eigenvalue and its eigenvector for a matrix $H = \sum_i H_i = \sum_i a_i P_i$

P_i: Pauli matrix (sparse matrix)

Data/Learning Nodes
Aquarius
7.20 PF, 578.2 TB/s

CUDA-Q: QPU-agnostic platform for accelerated quantum supercomputing



Expectation value $\langle H \rangle_{\psi(\theta)}$

Simulation Nodes
Odyssey
25.9 PF, 7.8 PB/s

SciPy: scipy.optimize for adjust the parameters

Figure from Peruzzo, A., McClean, J., Shadbolt, P. et al. A variational eigenvalue solver on a photonic quantum processor. Nat Commun 5, 4213 (2014).

Variational Quantum Eigensolver (VQE)

1. Generating quantum state $|\psi(\theta)\rangle$ on QC
2. Measuring $\langle H(\theta)\rangle = \langle\psi(\theta)|H|\psi(\theta)\rangle$
3. Based on the measured results, HPC determines θ that minimizes $\langle\psi(\theta)|H|\psi(\theta)\rangle$

By repeating these procedures until $\langle\psi(\theta)|H|\psi(\theta)\rangle$ converges, an approximate ground state is obtained.

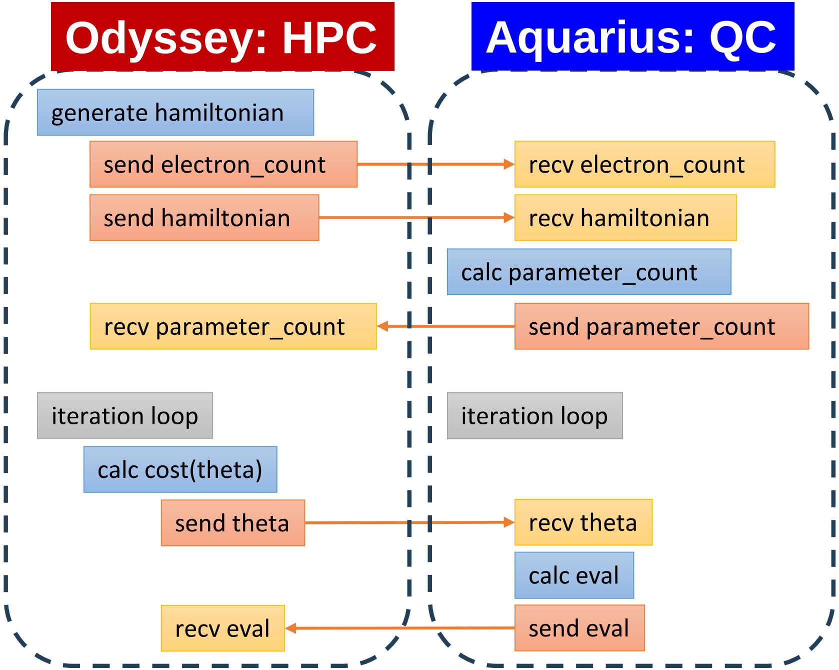
Variational Quantum Eigensolver (VQE)

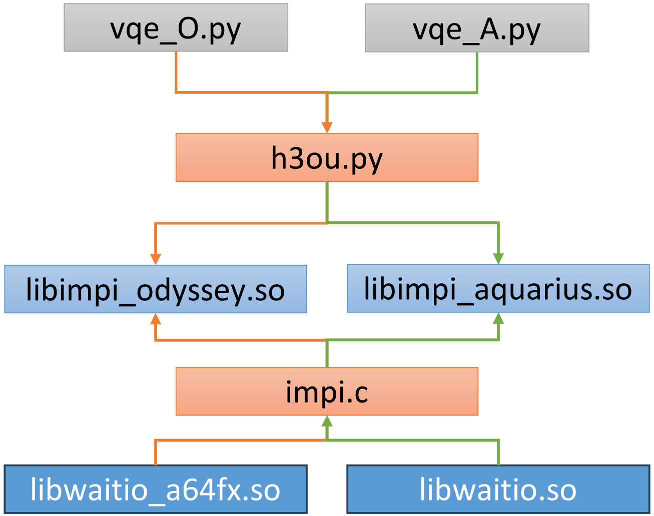
Aquarius QC	Generating quantum state $\vert\psi(\theta)\rangle$ on QC
	Measuring $\langle H(\theta)\rangle = \langle\psi(\theta)\vert H\vert\psi(\theta)\rangle$
Odyssey HPC	Based on the measured results, HPC determines θ that minimizes $\langle\psi(\theta)\vert H\vert\psi(\theta)\rangle$

By repeating these procedures until $\langle\psi(\theta)\vert H\vert\psi(\theta)\rangle$ converges, an approximate ground state is obtained.

QC-HPC Hybrid on Wisteria/BDEC-01

- Odyssey: HPC + Aquarius: QC
- Codes are written in Python, WaitIO with Python Interface





Python Scripts

Odyssey: HPC

```
def cost(theta):
    h3ou.h3ou_send(theta, 1)
    exp_val = h3ou.h3ou_recv(1)
    return exp_val

exp_vals = []

def callback(xk):
    exp_vals.append(cost(xk))

# Initial variational parameters.
np.random.seed(15)
x0 = np.random.normal(0, np.pi, parameter_count)

# Use the scipy optimizer to minimize the function of interest
result = minimize(
    cost, x0, method="COBYLA", callback=callback,
options={"maxiter": 100}
)
# Aquarius側で無限ループを出るために0を送信する
finish_array = np.zeros(3, dtype = np.float32)
h3ou.h3ou_send(finish_array, 1)
print(result)

print("FCI:", -1.1371757102406854) ## Classical solution
print("VQE:", result.fun + 0.7080240981000804)
h3ou.h3ou_finalize()
```

Aquarius: QC

```
@cudaq.kernel
def kernel(thetas: list[float]):
    qubits = cudaq.qvector(qubit_count)
    for i in range(electron_count):
        x(qubits[i])
    cudaq.kernels.uccsd(qubits, thetas, electron_count, qubit_count)

def compute_expval(theta: list[float]) -> float:
    return cudaq.observe(kernel, spin_operator, theta).expectation()

# Odysseyからthetaを受信しcomput_expvalで計算し値を返す
# 収束判定の結果を受信し、収束していたらループを抜ける
while True:
    theta_recv = h3ou.h3ou_recv(0)
    #print("theta_recv = ", theta_recv)
    if (np.all(theta_recv == 0)):
        break
    exp_val = compute_expval(theta_recv)
    #print("exp_val = ", exp_val)
    h3ou.h3ou_send(exp_val, 0)

print("vqe_A end")
h3ou.h3ou_finalize()
```

"h3ou_send", "h3ou_recv" for p-to-p communications

Odyssey: Loops for Convergence Evaluation

"theta" is sent from Odyssey to Aquarius

Cuda-Q functions are called on Aquarius, and results ("expval") is sent to Odyssey

Job Scripts

Odyssey HPC 1-core

```
#!/bin/bash
#PJM -L rscgrp=coupler-o
#PJM -L node=1x1x1:torus
#PJM --mpi proc=1
#PJM -L elapse=00:10:00
#PJM -g gt45
#PJM -N VQE_WAITIO

module load fj
module load fjmpi
module load waitio

export COUPLER_JOB_ID=1

###export WAITIO_COMM_TYPE=hybrid
###export WAITIO_COMM_TYPE=file
export WAITIO_COMM_TYPE=socket

export WAITIO_MASTER_HOST=`hostname`
export WAITIO_MASTER_PORT=7100
export WAITIO_PBID=0
export WAITIO_NPB=2

echo "waitio-serv START"
/work/share/waitio/bin/waitio-serv-a64fx -d -m $WAITIO_MASTER_HOST
echo "waitio-serv END"

export SYSTEM=odyssey

mpiexec -np 1 python3  ./vqe_O.py
```

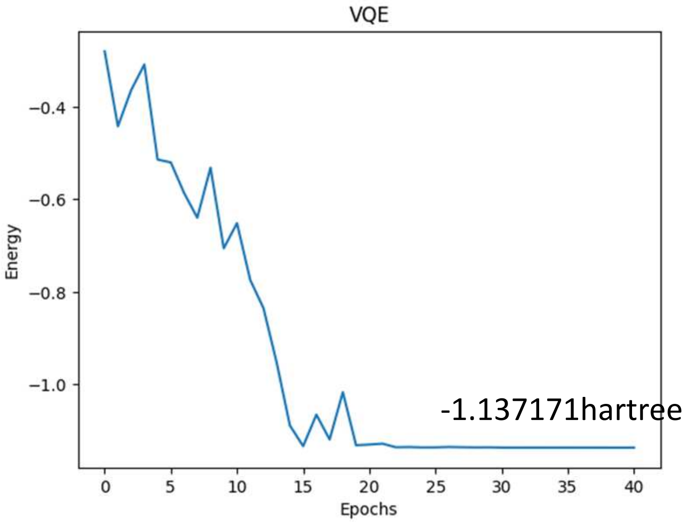
Aquarius QC 1-GPU

```
#!/bin/bash
#PJM -L rscgrp=coupler-a
#PJM -L node=1
#PJM --mpi proc=1
#PJM -L elapse=00:10:00
#PJM -g gt45
#PJM -N VQE_WAITIO

module purge
module load intel impi
module load waitio

export COUPLER_JOB_ID=1

###export WAITIO_COMM_TYPE=hybrid
###export WAITIO_COMM_TYPE=file
export WAITIO_COMM_TYPE=socket

echo "waitio-serv START"
export WAITIO_MASTER_HOST=`/work/share/waitio/bin/waitio-serv -c`
echo "waitio-serv END"
export WAITIO_MASTER_PORT=7100
export WAITIO_PBID=1
export WAITIO_NPB=2

module unload intel impi
module load gcc ompi

source /work/share/QC-HPC/miniconda3/bin/activate
conda activate cudaq-env

export SYSTEM=aquarius

python3 ./vqe_A.py
```

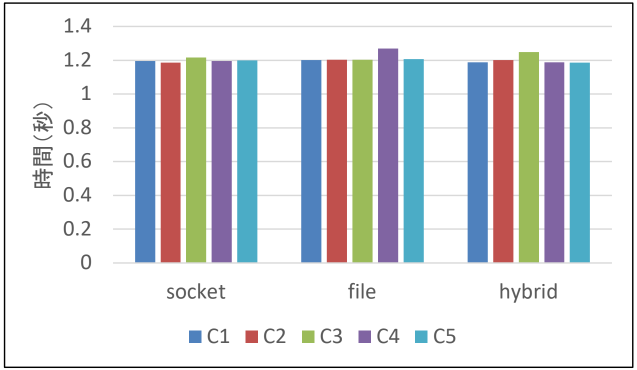
Demonstration of VQE: Hydrogen

- H_2

0.7474Å



- Small-scale molecule, well-known → good for verification
- Succeeded in finding the optimal parameters



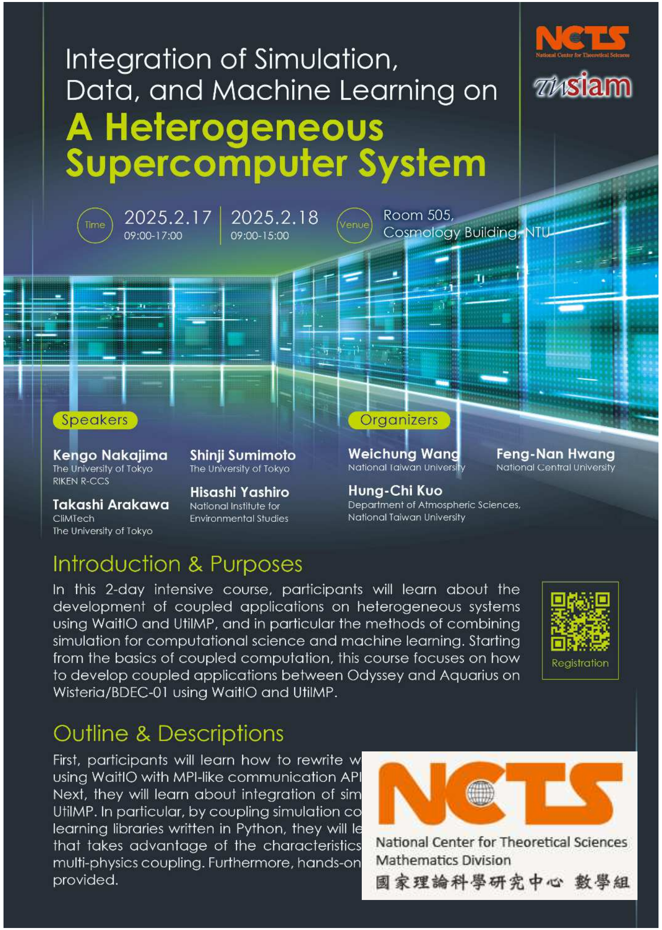
Computation Time on Odyssey for various types of communication methods

2025 NCTS Winter School

Feb.17-18, 2025@National Taiwan Univ.

https://ncts.ntu.edu.tw/events_3_detail.php?nid=358
http://nkl.cc.u-tokyo.ac.jp/NCTS2025W/

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 13:10-14:00 14:10-15:00 15:10-16:00	Overview of h3-Open-SYS/WaitIO, Parallel Computation on Heterogeneous Environment, Hands-on	Sumimoto
	16:10-17:00	Introduction to Global Atmospheric Simulation Overview of h3-Open-UTIL/MP	Yashiro Arakawa
Day-2	09:10-10:00 10:10-11:00 11:10-12:00 13:10-14:00	**Coupling of Global Atmospheric Simulation and AI, Hands-on**	
	14:10-15:00	**QC (Quantum Computing)-HPC Hybrid**	Nakajima Arakawa
	15:10-16:00	Roundtable Discussion	All



2025 NCTS Winter School

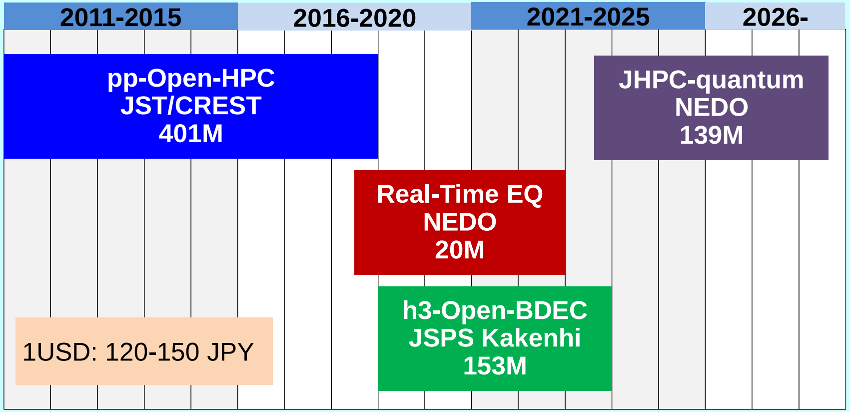
Feb.17-18, 2025@National Taiwan Univ.: 30+ On-Site Participants

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

http://nkl.cc.u-tokyo.ac.jp/NCTS2025W/

Summary: Sustainable Coupler

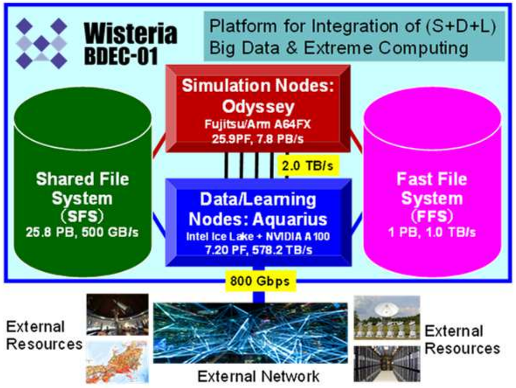
- MPI-based
  - Atmosphere (NICAM) - Ocean (COCO)
  - Other Applications
- Heterogeneous Environment
  - Combined with WaitIO
  - Real-Time Integration of (S+D+L)
  - Ensemble Coupling
- QC-HPC Hybrid Environment
  - Further development of WaitIO (e.g. WaitIO4Py, WatIO4Julia, WaitIO-Router)

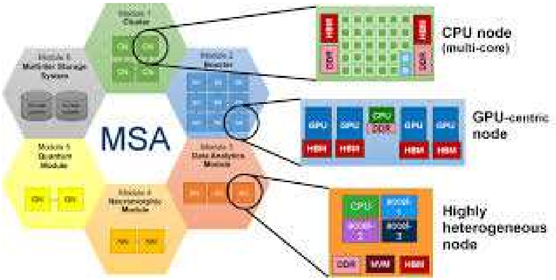


- **Lessons Learned: Sustainability of Software, such as Coupler**
  - **(Strongly depends on continuous funding)**
  - **Extensions of target appl’s and systems with additional developments for adaptation (e.g. WaitIO) are important.**
    - **Coupler is a very important workflow tool towards “AI for Science”**
  - **We must always keep our eyes open, and pay attention to whether it can be applied somewhere.**

Future Issues

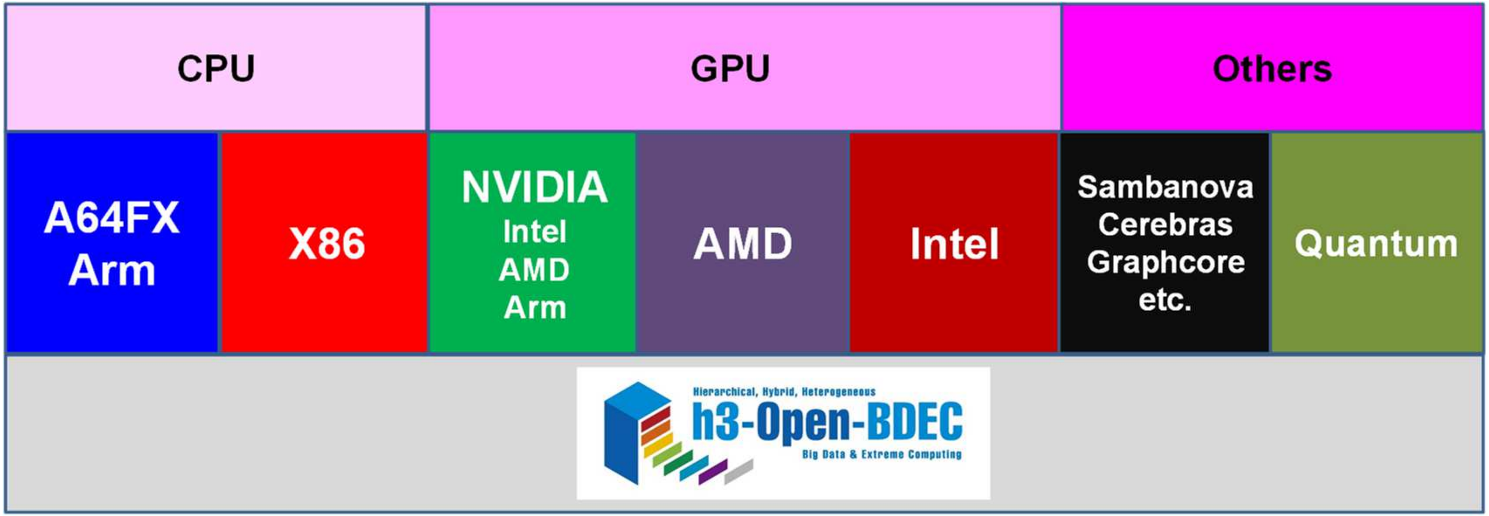
- Heterogeneous Computing, Modular Supercomputing Architecture (MSA) proposed by JSC (Jülich Supercomputing Center, Germany)
  - The idea is to run different workloads (Simulation, Data, Learning, Inferencing) on computers that are suitable for each workload.
  - Very important in “AI-for-Science” Era
  - QC-HPC Hybrid is an example of Heterogenous Computing, and MSA





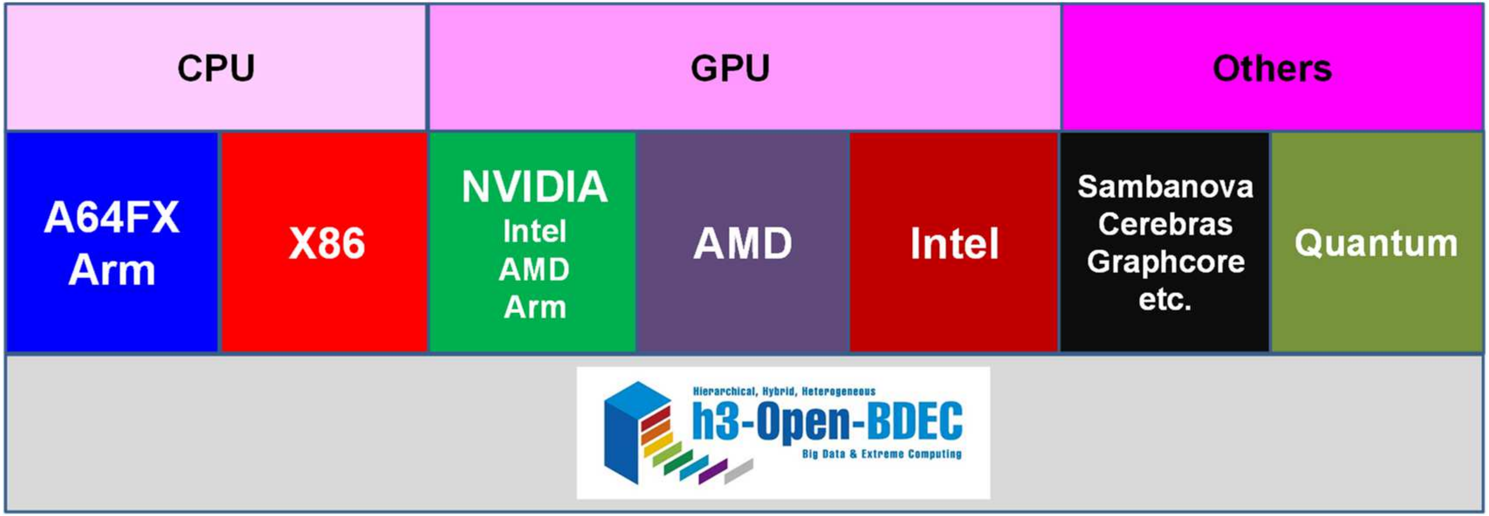
- h3-Open-BDEC: WaitIO, Coupler: Data Transfer
  - Originally developed for connecting Odyssey-Aquarius
  - Applied to QC-HPC Hybrid via SINET6 (nation-wide network)(under development)
  - This idea can be applied to “connecting” multiple supercomputer systems
    - e.g. Fugaku-Miyabi, AMD GPUs for HPC-NVIDIA GPUs for AI

- Q1 : What missing interoperability layers (software, standard, or abstraction) would most accelerate convergence between traditional HPC linear algebra workflows and today's extreme-scale AI workloads ?
  - Standard Interface between HPC and AI Workloads under Heterogeneous Environments.
  - "Coupler" is really important as a Workflow Tool



- Q2 :Looking ahead to 2030, do you expect the principal bottleneck for extreme-scale AI to be data, algorithms, resilience or energy, and how does that prediction shape your research priorities today ?
  - High-Quality and Reliable Data
    - Generation of Data
    - Simulation/HPC is still important
  - Energy

- Q3 : Given the different developments in architecture processors for AI and “computational science”, do you think we'll see a convergence or divergence of roadmaps?
  - “Divergence” is more important for future of “AI-for-Science”
  - If we have a good “Coupler”, we do not need to worry about programming.
  - Bridging multiple systems, AMD-for-HPC and NVIDIA/Sambanova-for-AI



SCA/HPCAsia 2026

Everything with HPC - AI, Cloud, QC and Future Society

Date January 26-29, 2026

Venue Osaka International Convention Center (Osaka, Japan)

- Jointly hosted b
  conference on H
- (Planned) Co-lo
  - Trillion Parame
  - Accelerated Da
  - HANAMI Works
  - ACM Asian Inte
  - R-CCS International Symposium
  - Fugaku/R-CCS Tour in Kobe
  - DOE-MEXT Meeting (Jan.30/31@Kobe)
- Osaka: 2nd Largest City in Japan
  - Expo 2025 Osaka Kansai
  - 30-min by train to Kyoto and Kobe (Home of Fugaku)
  - 150-min by High-Speed Train from Tokyo

✓ Technical Papers/Posters
✓ Workshops
✓ BoF
✓ Tutorials

SCAsia Supercomputing Asia — Gathering the Best of HPC in Asia

HPC Asia 2026

富岳

SCA/HPCAsia 2026: Call for Submissions

- Event Overview:
  - Date: January 26-29, 2026
  - Venue: Osaka International Convention Center (Osaka, Japan)
  - Theme: “Everything with HPC –AI, Cloud, QC, and Future Society”
- Call for Submissions: Papers, Posters, Workshops, BoFs, and Tutorials

Papers	Posters	Workshops	Birds of a Feather	Tutorials
Paper abstracts: 29 Aug 2025 Submissions close: 5 Sep 2025 Result notification: 20 Oct 2025	Submissions close: 27 Oct 2025 Result notification: 14 Nov 2025	Submissions close: 30 Jun 2025 Result notification: 31 Jul 2025	Submissions close: 1 Sep 2025 Result notification: 1 Oct 2025	Submissions close: 11 Jul 2025 Result notification: 15 Aug 2025

For more details, please visit our website:
https://www.sca-hpcasia2026.jp/

65th TOP500 List (June, 2025)

R_{max}: Performance of Linpack (TFLOPS)
R_{peak}: Peak Performance (TFLOPS), Power: kW

http://www.top500.org/

	Site	Computer/Year Vendor	Cores	R_{max} (PFLOPS)	R_{peak} (PFLOPS)	GFLOPS/W	Power (kW)
1	**El Capitan, 2024, USA** DOE/NNSA/LLNL	HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS	11,039,616	1,742.00 (=1.742 EF)	2,746.38 63.4 %	58.99	29,581
2	**Frontier, 2021, USA** DOE/SC/Oak Ridge National Laboratory	HPE Cray EX235a, AMD Optimized 3rd Gen. EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11	9,066,176	1.353.00	2,055.72 65.8 %	54.98	24,607
3	**Aurora, 2023, USA** DOE/SC/Argonne National Laboratory	HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel	9,264,128	1,012.00	1,980.01 51.1 %	26.15	38,698
4	**JUPITER Booster, 2025, USA** EuroHPC/FZJ	EVIDEN, BullSequana XH3000, GH Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, RedHat Enterprise Linux	4,801,344	794,40	930.00 85.3 %	60.62	13,088
5	**Eagle, 2023, USA** Microsoft	Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR	2,073,600	561.20	846.84 66.3 %		
6	**HPC 6, 2024, Italy** Eni S.p.A.	HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, RHEL 8.9	3,143,520	477.90	606.97 66.3 %	56.48	8,461
7	**Fugaku, 2020, Japan** R-CCS, RIKEN	Fujitsu PRIMEHPC FX1000, Fujitsu A64FX 48C 2.2GHz, Tofu-D	7,630,848	442.01	537.21 82.3 %	14.78	29,899
8	**Alps, 2024, Switzerland** Swiss Natl. SC Centre (CSCS)	HPE Cray EX254n, NVIDIA Grace 72C 3.1GHz, NVIDIA GH200 Superchip, Slingshot-11	2,121,600	434.90	574.84 75.7 %	61.05	7,124
9	**LUMI, 2023, Finland** EuroHPC/CSC	HPE Cray EX235a, AMD Optimized 3rd Gen. EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11	2,752,704	379.70	531.51 71.4 %	53.43	7,107
10	**Leonard, 2023, Italy** EuroHPC/Cineca	EVIDEN, BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64GB, Quad-rail NVIDIA HDR100	1,824,768	241.20	181.49 78.7 %	32.19	7,494
15	**ABCI 3.0, 2025, Japan** AIST	HPE Cray XD670, Xeon Platinum 8558 48C 2.1GHz, NVIDIA H200 SXM5 141 GB, Infiniband NDR200, Rocky Linux 9	479,232	145.10	130.44 80.0 %	59.29	3,596
22	**CHIE-3, 2024, Japan** SoftBank, Corp.	NVIDIA DGX H100, Xeon Platinum 8480C 56C 2GHz, NVIDIA H100, Infiniband NDR400, Ubuntu 22.04.4 LTS	297,840	91.94	138.32 66.5 %		
24	**CHIE-2, 2024, Japan** SoftBank, Corp.	NVIDIA DGX H100, Xeon Platinum 8480C 56C 2GHz, NVIDIA H100, Infiniband NDR400, Ubuntu 22.04.4 LTS	297,840	89.78	138.32 64.9 %		
27	**ABCI-Q, 2025, Japan** AIST	Fujitsu, Supermicro SYS-221GE-TNHT-LCC, Intel Xeon Platinum 8558 48C 2.1GHz, NVIDIA H100 SXM5 80GB, Infiniband NDR, Rocky Linux 9.4	315,210	74.58	99.35 75.1 %	40.67	1,834
36	**FPT, 2025, JAPAN** FPT AI Factory	HPE, HGX H200, Xeon Platinum 8558 48C 2.1GHz, NVIDIA H200 SXM5 141 GB, Infiniband NDR400, Ubuntu 22.04.5 LTS	146,304	49.85	67.44 73.9 %		
37	**Miyabi-G, 2024, Japan** JCAHPC	Fujitsu, Supermicro ARS 111GL DNHR LCC, Grace Hopper Superchip 72C 3GHz, Infiniband NDR200, Rocky Linux	80,640	46.80	72.80 64.3 %	47.59	983
46	**TSUBAME 4.0, 2024, Japan** Institute of Science Tokyo	HPE Cray XD665, AMD EPYC 9654 96C 2.4GHz, NVIDIA H100 SXM5 94 GB, Infiniband NDR200	172,800	39.62	61.60 64.3 %	48.55	816
73	**Wisteria/BDEC-01 (Odyssey), 2021, Japan** U.Tokyo	Fujitsu PRIMEHPC FX1000, A64FX 48C 2.2GHz, Tofu D	368,640	22.12	25.95 85.2 %	15.07	1,468