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

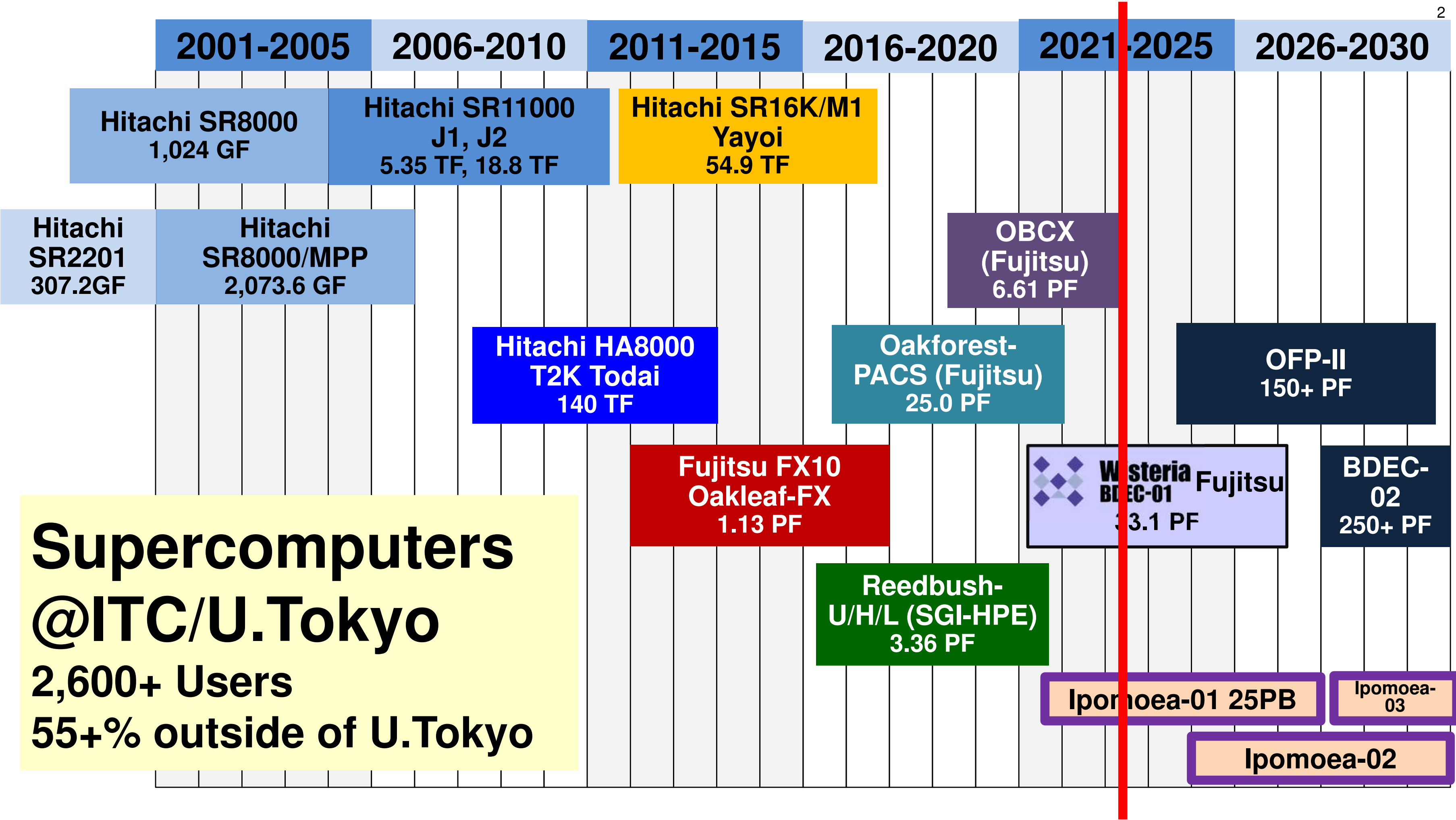
東京大学
THE UNIVERSITY OF TOKYO

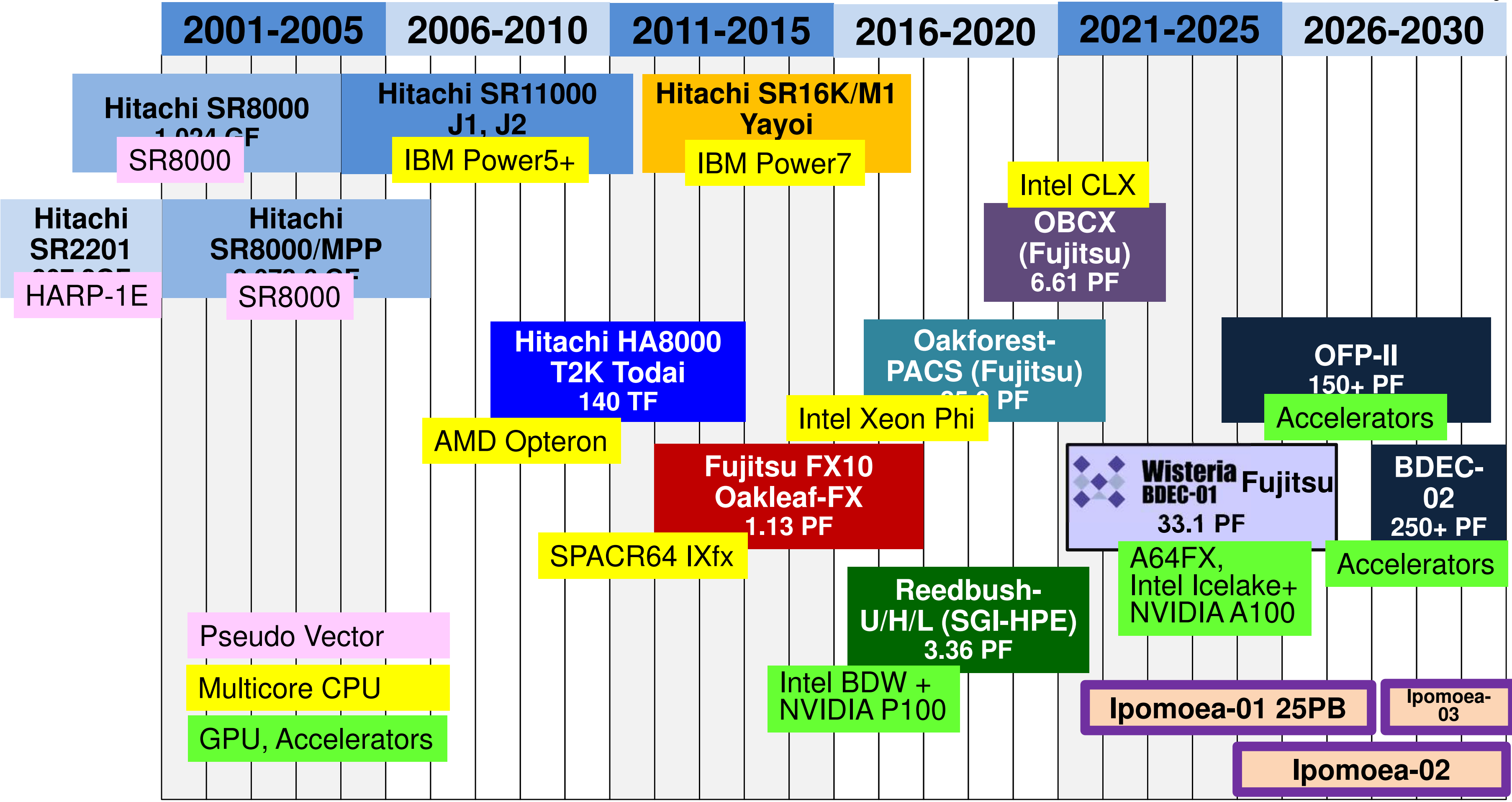
Innovative Scientific Computing by Integration of (Simulation+Data+Learning)

Wisteria BDEC-01

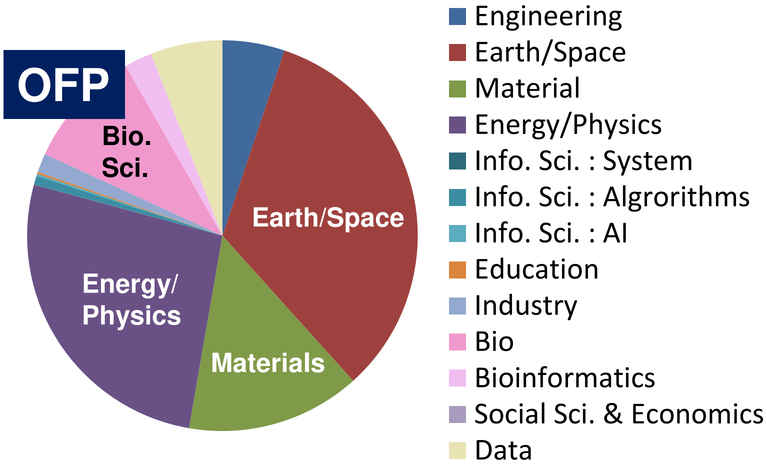
Hierarchical, Hybrid, Heterogeneous
h3-Open-BDEC
Big Data & Extreme Computing

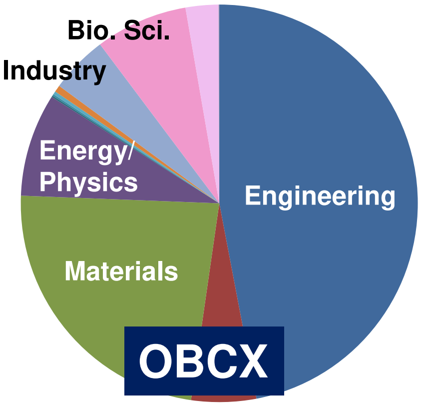
Kengo Nakajima
Information Technology Center
The University of Tokyo


2001-2005
2006-2010
2011-2015
2016-2020
2021-2025
2026-2030
Hitachi SR8000
1,024 GF
Hitachi SR11000
J1, J2
5.35 TF, 18.8 TF
Hitachi SR16K/M1
Yayoi
54.9 TF
Hitachi
SR2201
307.2GF
Hitachi
SR8000/MPP
2,073.6 GF
OBCX
(Fujitsu)
6.61 PF
Hitachi HA8000
T2K Todai
140 TF
Oakforest-
PACS (Fujitsu)
25.0 PF
OFP-II
150+ PF
Fujitsu FX10
Oakleaf-FX
1.13 PF
Wisteria
BDEC-01
Fujitsu
33.1 PF
BDEC-
02
250+ PF
Reedbush-
U/H/L (SGI-HPE)
3.36 PF
Ipomoea-01 25PB
Ipomoea-
03
Ipomoea-02
Supercomputers
@ITC/U.Tokyo
2,600+ Users
55+% outside of U.Tokyo


2001-2005
2006-2010
2011-2015
2016-2020
2021-2025
2026-2030
Hitachi SR8000
SR8000
Hitachi SR11000
J1, J2
IBM Power5+
Hitachi SR16K/M1
Yayoi
IBM Power7
Intel CLX
OBCX
(Fujitsu)
6.61 PF
Hitachi
SR2201
HARP-1E
Hitachi
SR8000/MPP
SR8000
Hitachi HA8000
T2K Todai
140 TF
AMD Opteron
Oakforest-
PACS (Fujitsu)
Intel Xeon Phi
OFP-II
150+ PF
Accelerators
Fujitsu FX10
Oakleaf-FX
1.13 PF
SPACR64 IXfx
Wisteria
BDEC-01
Fujitsu
33.1 PF
A64FX,
Intel Icelake+
NVIDIA A100
BDEC-
02
250+ PF
Accelerators
Reedbush-
U/H/L (SGI-HPE)
3.36 PF
Intel BDW +
NVIDIA P100
Ipomoea-01 25PB
Ipomoea-
03
Ipomoea-02
Pseudo Vector
Multicore CPU
GPU, Accelerators

Research Area based on Machine Hours (FY.2020)

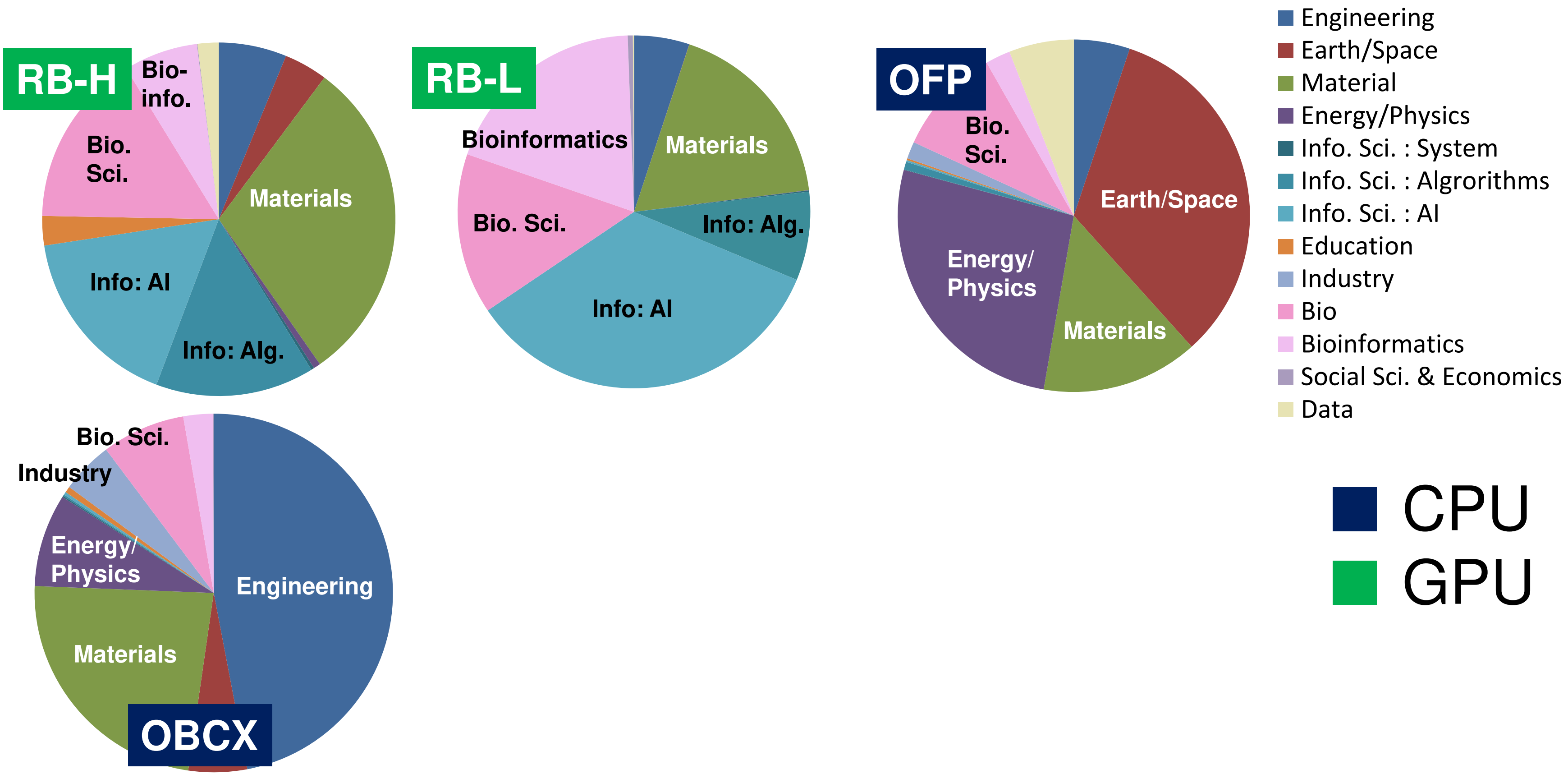




CPU

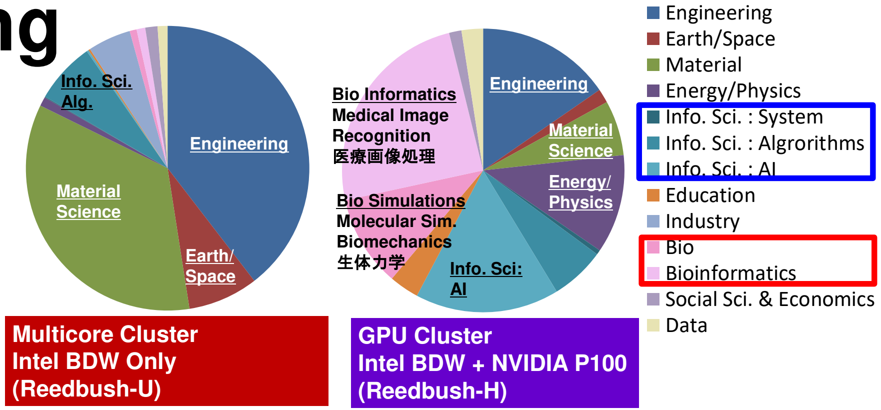
GPU

Research Area based on Machine Hours (FY.2020)



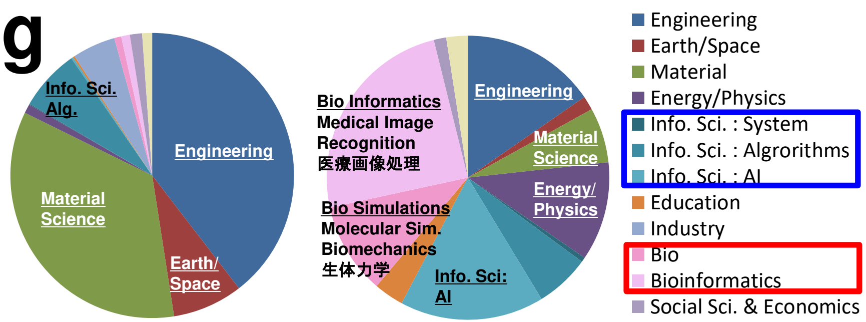
Future of Supercomputing

- Various Types of Workloads
  - Computational Science & Engineering: Simulations
  - Big Data Analytics
  - AI, Machine Learning …



Future of Supercomputing

- Various Types of Workloads
  - Computational Science & Engineering: Simulations
  - Big Data Analytics
  - AI, Machine Learning …
- Integration/Convergence of (Simulation + Data + Learning) (S+D+L) is important towards Society 5.0
  - Super Smart & Human-centered Society by Digital Innovation (IoT, Big Data, AI etc.) and by Integration of Cyber Space & Physical Space

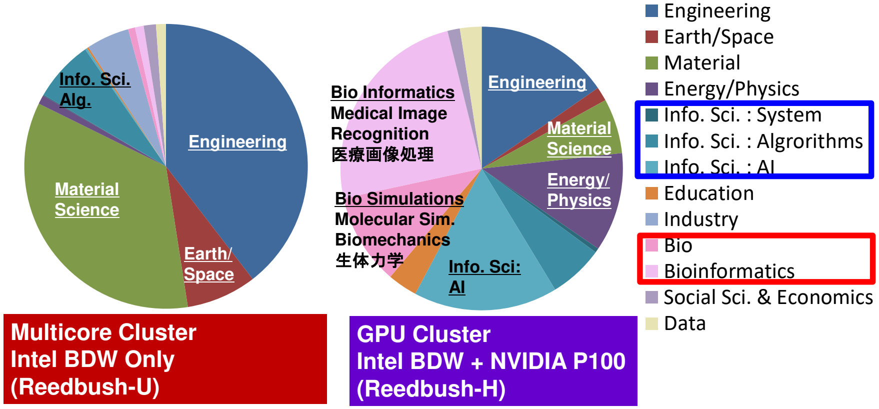


Multicore Cluster Intel BDW Only (Reedbush-U)

GPU Cluster Intel BDW + NVIDIA P100 (Reedbush-H)

Future of Supercomputing

- Various Types of Workloads
  - Computational Science & Engineering: Simulations
  - Big Data Analytics
  - AI, Machine Learning …
- **Integration/Convergence of (Simulation + Data + Learning) (S+D+L) is important towards Society 5.0**



- **BDEC (Big Data & Extreme Computing)**
  - **Platform for Integration of (S+D+L)**
  - **Focusing on S (Simulation)**
    - **AI for HPC, AI for Science, Digital Twins**
  - **Planning started in 2015**

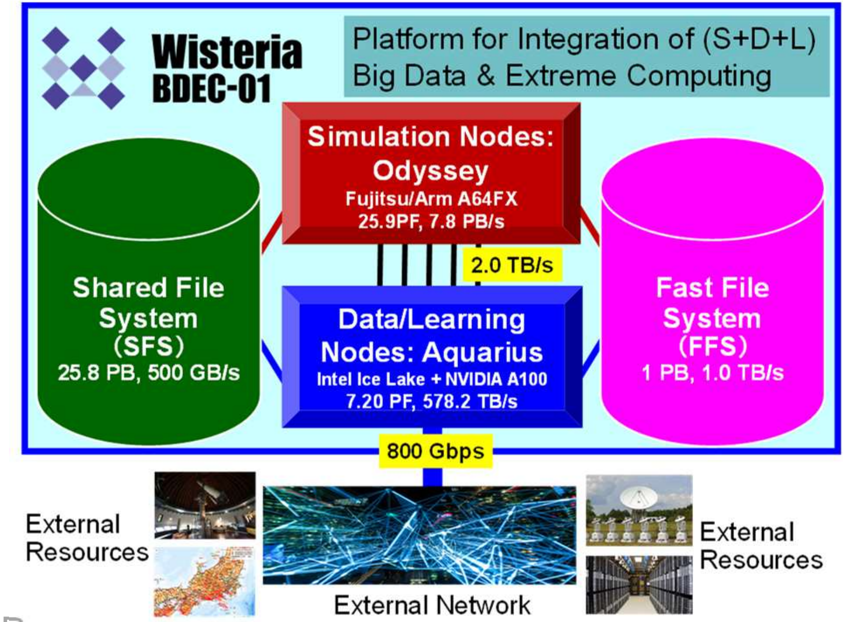
BDEC (Big Data & Extreme Computing)

S + D + L

Wisteria/BDEC-01

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

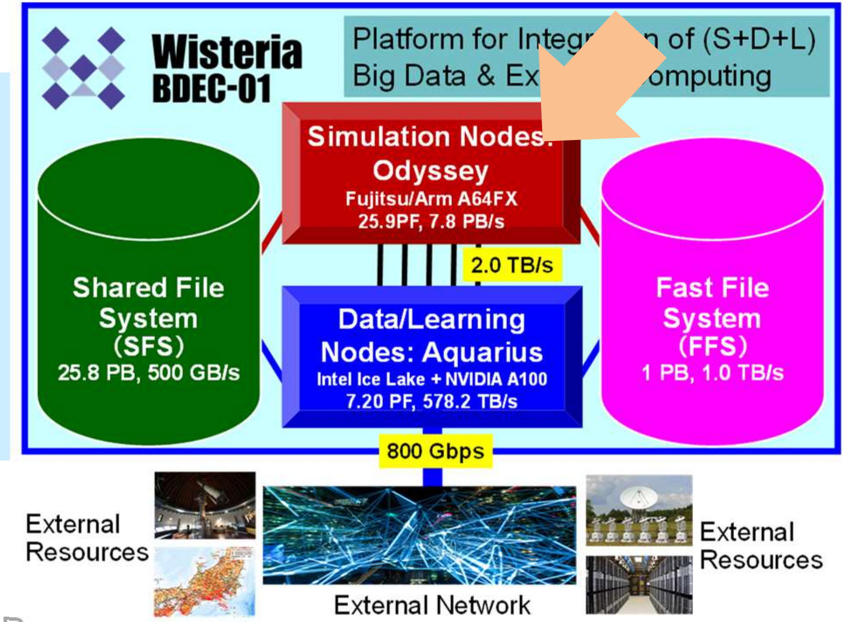
- Operation starts on May 14, 2021
- 33.1 PF, 8.38 PB/sec by **Fujitsu**
  - ~4.5 MVA with Cooling, ~360m²
- 2 Types of Node Groups
  - Hierarchical, Hybrid, Heterogeneous (h3)
  - Simulation Nodes: Odyssey
    - Fujitsu PRIMEHPC FX1000 (A64FX), 25.9 PF
      - 7,680 nodes (368,640 cores), Tofu-D
      - General Purpose CPU + HBM
      - Commercial Version of "Fugaku"
  - Data/Learning Nodes: Aquarius
    - Data Analytics & AI/Machine Learning
    - Intel Xeon Ice Lake + NVIDIA A100, 7.2PF
      - 45 nodes (90x Ice Lake, 360x A100), IB-HDR
    - Some of the DL nodes are connected to external resources directly
- File Systems: SFS (Shared/Large) + FFS (Fast/Small)



Wisteria/BDEC-01

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

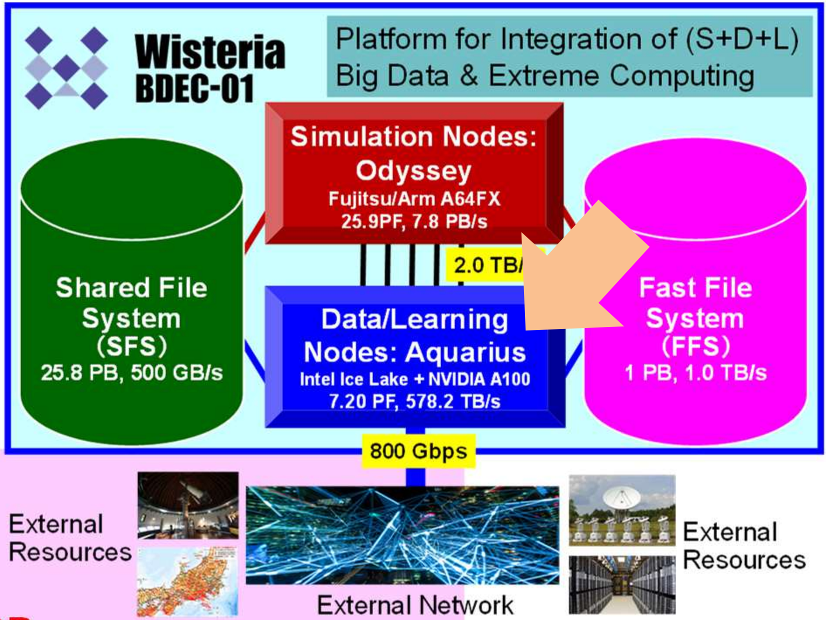
- Operation starts on May 14, 2021
- 33.1 PF, 8.38 PB/sec by **Fujitsu**
  - ~4.5 MVA with Cooling, ~360m^2
- **2 Types of Node Groups**
  - Hierarchical, Hybrid, Heterogeneous (h3)
  - **Simulation Nodes: Odyssey**
    - **Fujitsu PRIMEHPC FX1000 (A64FX), 25.9 PF**
      - **7,680 nodes (368,640 cores), Tofu-D**
      - **General Purpose CPU + HBM**
      - **Commercial Version of "Fugaku"**
  - Data/Learning Nodes: Aquarius
    - Data Analytics & AI/Machine Learning
    - Intel Xeon Ice Lake + NVIDIA A100, 7.2PF
      - 45 nodes (90x Ice Lake, 360x A100), IB-HDR
    - Some of the DL nodes are connected to external resources directly
- File Systems: SFS (Shared/Large) + FFS (Fast/Small)



Wisteria/BDEC-01

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

- Operation starts on May 14, 2021
- 33.1 PF, 8.38 PB/sec by **Fujitsu**
  - ~4.5 MVA with Cooling, ~360m²
- **2 Types of Node Groups**
  - Hierarchical, Hybrid, Heterogeneous (h3)
  - **Simulation Nodes: 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 Nodes: Aquarius**
    - **Data Analytics & AI/Machine Learning**
    - **Intel Xeon Ice Lake + NVIDIA A100, 7.2PF**
      - **45 nodes (90x Ice Lake, 360x A100), IB-HDR**
    - **Some of the DL nodes are connected to external resources directly**
- File Systems: SFS (Shared/Large) + FFS (Fast/Small)



61st TOP500 List (June, 2023)

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

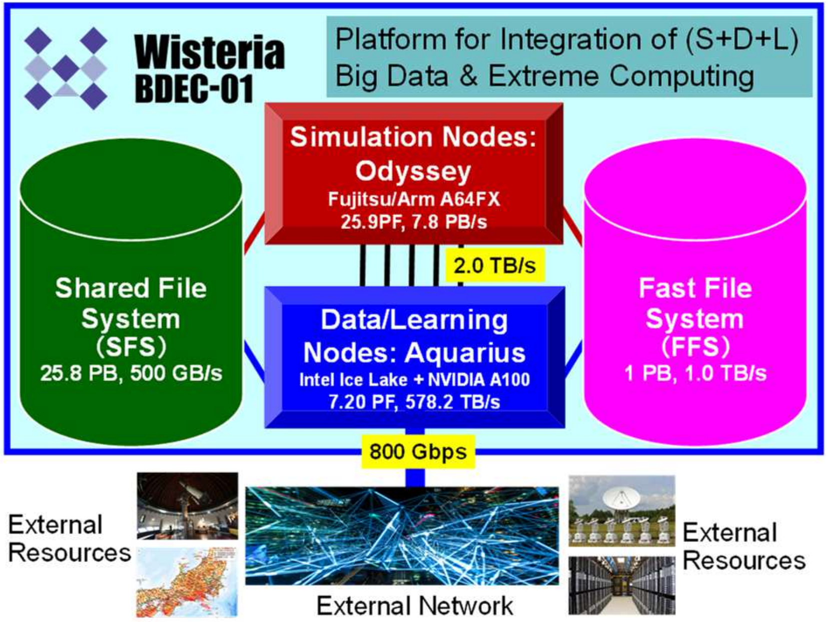
http://www.top500.org/

	Site	Computer/Year Vendor	Cores	R_{max} (PFLOPS)	R_{peak} (PFLOPS)	Power (kW)
1	**Frontier, 2022, USA** DOE/SC/Oak Ridge National Laboratory	HPE Cray EX235a, AMD Optimized 3rd Gen. EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11	8,699,904	1,194.00 (=1.194 EF)	1,679.82	**22,703**
2	**Fugaku, 2020, Japan** R-CCS, RIKEN	Fujitsu PRIMEHPC FX1000, Fujitsu A64FX 48C 2.2GHz, Tofu-D	7,630,848	442,010 (= 442.0 PF)	537,212.0	**29,899**
3	**LUMI, 2022, Finland** EuroHPC/CSC	HPE Cray EX235a, AMD Optimized 3rd Gen. EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11	2,220,288	309.10	428.70	**6,016**
4	**Leonard, 2022, Italy** EuroHPC/Cineca	BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64GB, Quad-rail NVIDIA HDR100	1,824,768	238.70	304.47	**7,404**
5	**Summit, 2018, USA** DOE/SC/Oak Ridge National Laboratory	IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR InfiniBand	2,414,592	148.60	200.79	**10,096**
6	**Sierra, 2018, USA** DOE/NNSA/LLNL	IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR InfiniBand	1,572,480	94.64	125.71	**7,438**
7	**Sunway TaihuLight, 2016, China** National Supercomputing Center in Wuxi	Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway	10,649,600	93.01	125.44	**15,371**
8	**Perlmutter, 2021, USA** DOE/NERSC/LBNL	HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10	761,856	70.87	93.75	**2,528**
9	**Selene, 2020, USA** NVIDIA	NVIDIA DGX A100 SuperPOD, AMD EPYC 7742 64C 2.25GHz, NVIDIA GA100, Mellanox Infiniband HDR	555,520	63.46	79.22	**2,646**
10	**Tianhe-2A, 2018, China** National Super Computer Center in Guangzhou	TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000	4,981,760	61.44	100.68	**18,482**
24	**ABCI 2.0, 2021, Japan** AIST	PRIMERGY GX2570 M6, Xeon Platinum 8360Y 36C 2.4GHz, NVIDIA A100 SXM4 40 GB, InfiniBand HDR	504,000	22.21	54.34	**1,600**
25	**Wisteria/BDEC-01 (Odyssey), 2021, Japan** ITC, University of Tokyo	PRIMEHPC FX1000, A64FX 48C 2.2GHz, Tofu interconnect D	368,640	22.12	25.95	**1,468**

Rankings@ISC-HPC 23
June 2023

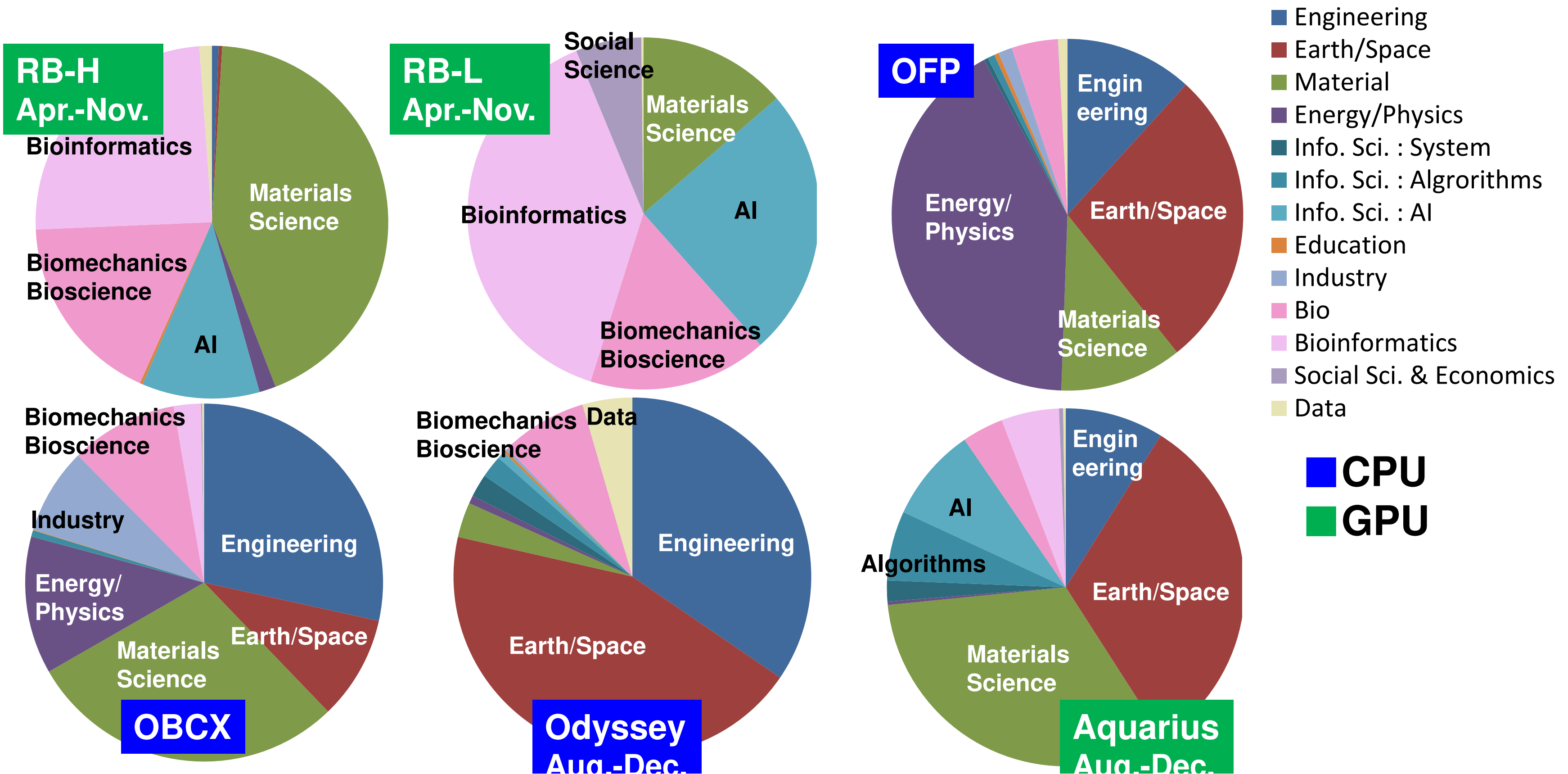
IMAGINE TOMORROW
ISC High Performance
MAY 21 – MAY 25, 2023 | HAMBURG, GERMANY

	Odyssey	Aquarius
TOP 500	25	136
Green 500	50	32
HPCG	12	68
Graph 500 BFS	6	-
HPL-MxP (HPL-AI)	14 (Nov.2022)	-



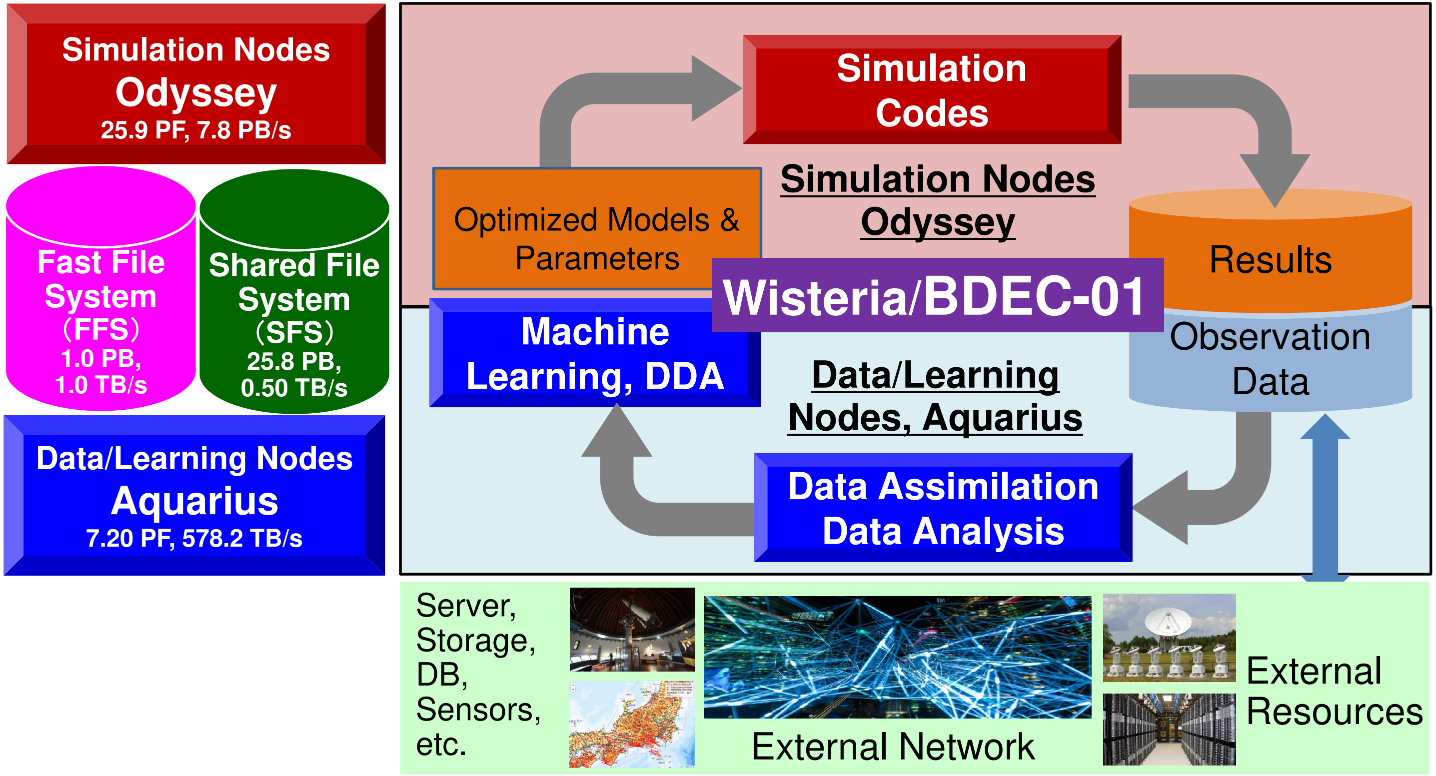
Research Area based on CPU Hours (FY.2021)

Apr.2021-Dec.2021, until Nov.: RB-H/L, from Aug.: Odyssey & Aquarius



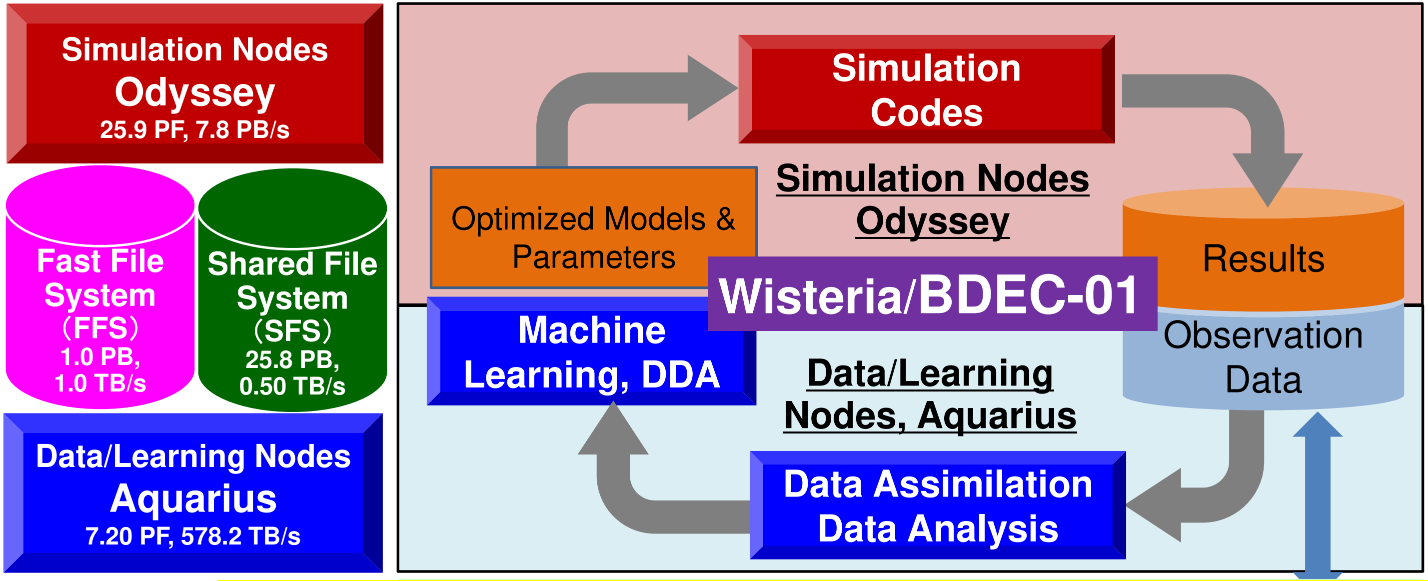
Wisteria/BDEC-01

Platform for Integration of (Simulation+Data+Learning) (S+D+L)



Wisteria/BDEC-01

Platform for Integration of (Simulation+Data+Learning) (S+D+L)

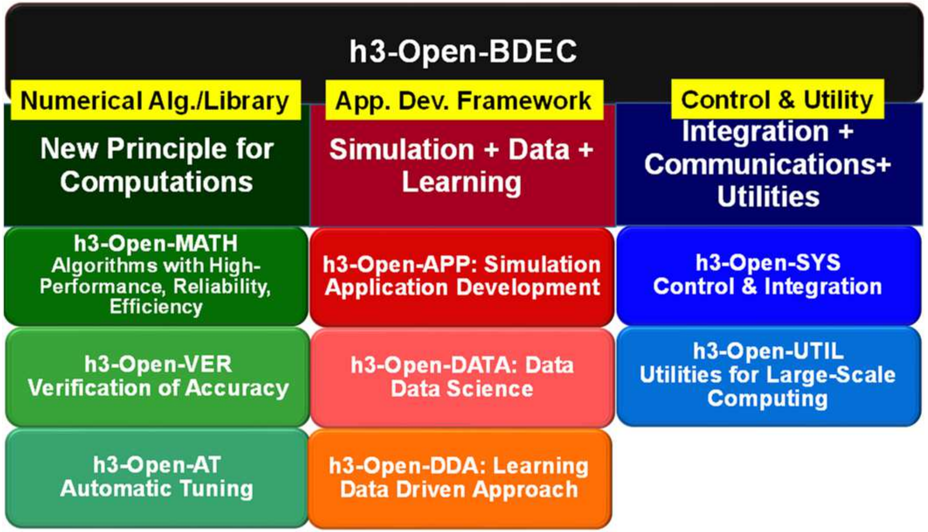


Optimization of Models/Parameters for Simulations by Data Analytics & Machine Learning (S+D+L)

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
- Leading-PI: Kengo Nakajima (The University of Tokyo)
- Total Budget: 1.41M USD

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

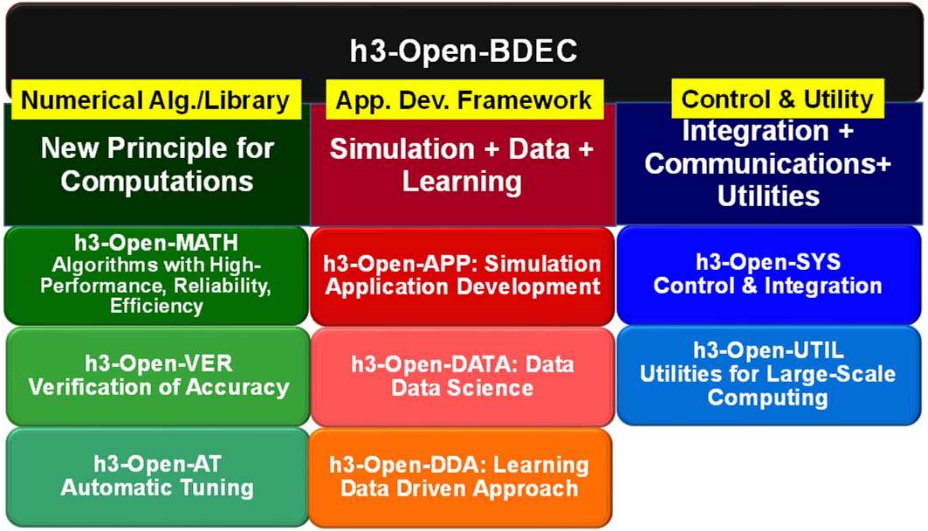


東京大学 THE UNIVERSITY OF TOKYO
RIKEN
R-CCS
国立研究開発法人 国立環境研究所 National Institute for Environmental Studies
北海道大学 HOKKAIDO UNIVERSITY
Tokyo Woman's Christian University 東京女子大学
名古屋大学 NAGOYA UNIVERSITY

JAPAN SOCIETY FOR THE PROMOTION OF SCIENCE · FOUNDED 1932 ·

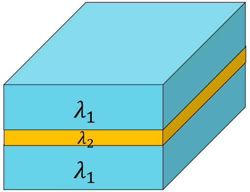
h3-Open-BDEC: Innovative Software Platform for Integration of (S+D+L) on the BDEC System, such as Wisteria/BDEC-01

- "Three" Innovations
  - New Principles for Numerical Analysis by Adaptive Precision, Automatic Tuning & Accuracy Verification
  - Hierarchical Data Driven Approach (*h*DDA) based on Machine Learning
  - Software & Utilities for Heterogenous Environment, such as Wisteria/BDEC-01

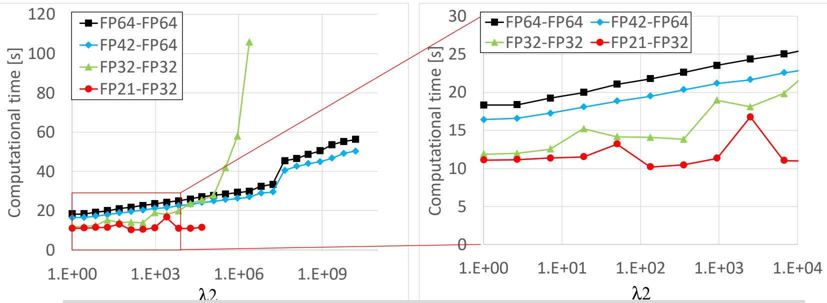


Adaptive Precision Computing with FP42/FP21

Masatoshi Kawai (kawai@cc.u-tokyo.ac.jp)



Heat Conduction with
Heterogeneous Material Property

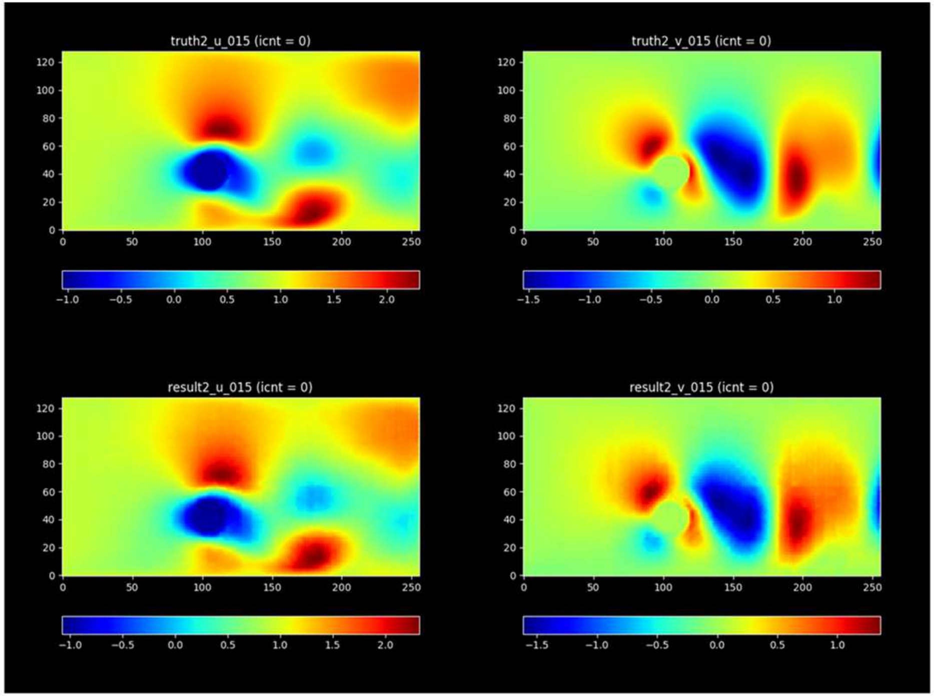


Computation Time for ICCG Solver
Various Types of Precisions on Intel Xeon Cascadelake

In recent years, the usefulness of low-precision floating-point representation has been studied in various fields such as machine learning. Low accuracy can be expected to have effects such as shortening calculation time and reducing power consumption. For example, in an application with a memory bandwidth bottleneck, the effect of reducing the calculation time by reducing the amount of memory transfer is significant. However, in fields such as iterative methods, it is common to use FP64 because the calculation accuracy strongly affects the convergence, and there are few application examples of low-precision arithmetic. This study investigates the applicability of low-precision representation to the Krylov subspace and stationary iterative methods. In this research, we focus on the FP32, FP16, and FP42, FP21, which are not standardized by IEEE754. Developed method has been evaluated for ICCG solver, which solves linear equations derived from 3D FVM code for steady-state head conduction with heterogeneous material property ($\lambda_1=10^0$, $\lambda_2=10^0\sim10^9$). Generally, computation with lower precision (e.g. FP32-FP32, FP21-FP32) becomes unstable, if condition number of the coefficient matrix is larger (λ_2 is larger), FP21-FP32 provides the best performance if λ_2 is up to 10^4. (“FP21-FP32” means “matrices are in FP21, and vectors are in FP32)

Prediction of CFD Simulation by Deep Learning

Takashi Shimokawabe (shimokawabe@cc.u-tokyo.ac.jp)

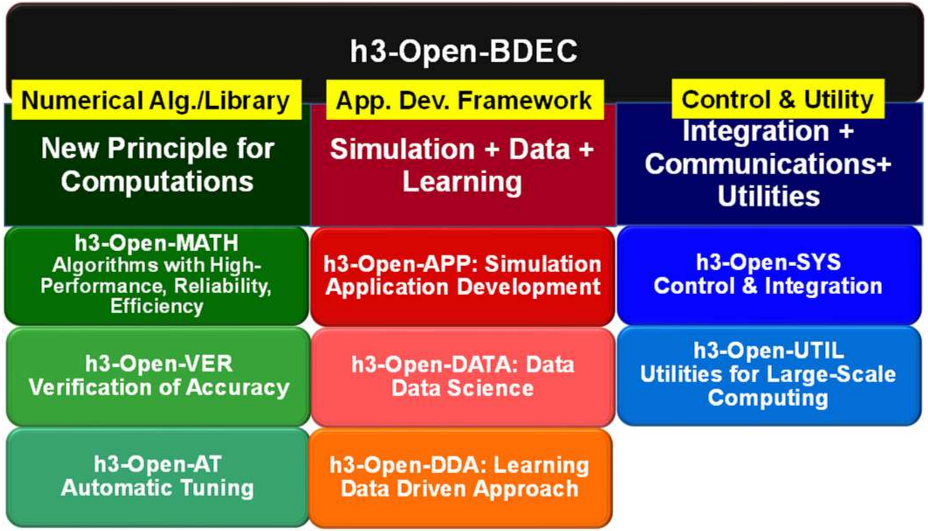


Comparison of the flow velocity results obtained by the conventional simulation (upper) and the prediction of these results by deep learning (lower)

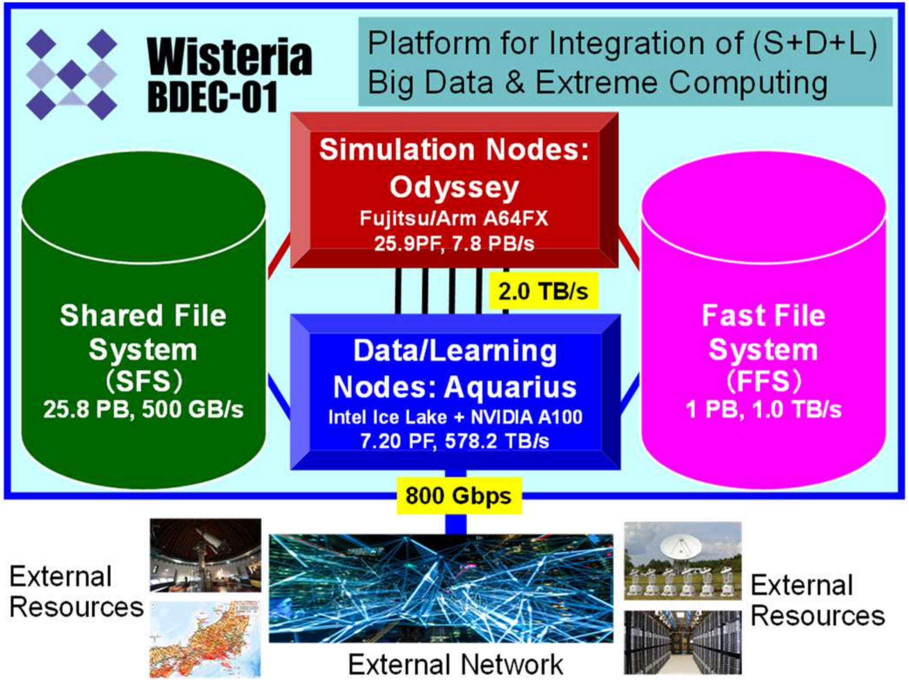
Computational fluid dynamics (CFD) is widely used in science and engineering. However, since CFD simulations requires a large number of grid points and particles for these calculations, these kinds of simulations demand a large amount of computational resources such as supercomputers. Recently, deep learning has attracted attention as a surrogate method for obtaining calculation results by CFD simulation approximately at high speed. We are working on a project to develop a parallelization method to make it possible to apply the surrogate method based on the deep learning to large scale geometry. Unlike the model parallel computing, the method we are currently developing predicts large-scale steady flow simulation results by dividing the input geometry into multiple parts and applying a single small neural network to each part in parallel. This method is developed based on considering the characteristics of CFD simulation and the consistency of the boundary condition of each divided subdomain. By using the physical values on the adjacent subdomains as boundary conditions, applying deep learning to each subdomain can predict simulation results consistently in the entire computational domain. It is possible to predict the simulation results in about 36.9 seconds by the developed method, compared to about 286.4 seconds by the conventional numerical method. In addition to this, we are also attempting to develop a method for fast prediction of time evolution calculations using deep learning.

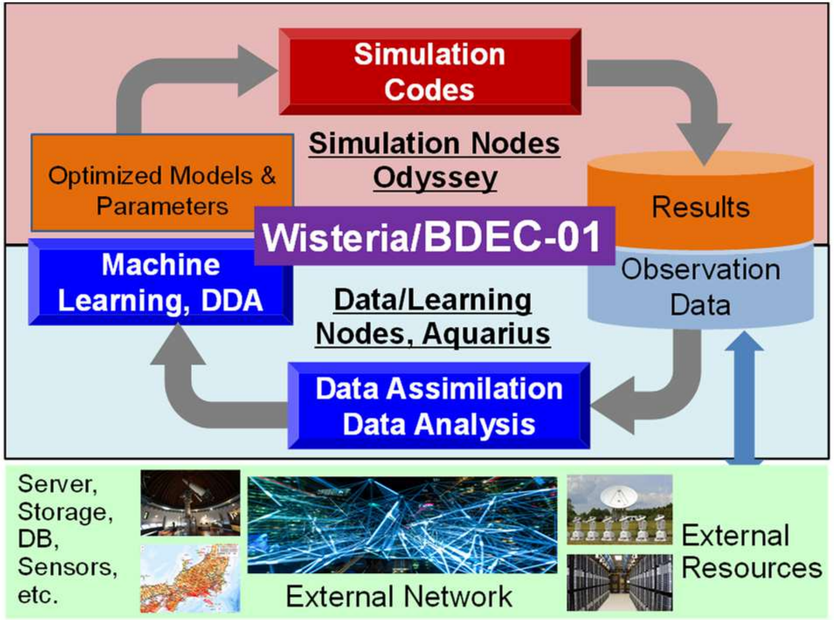
h3-Open-BDEC: Innovative Software Platform for Integration of (S+D+L) on the BDEC System, such as Wisteria/BDEC-01

- “Three” Innovations
  - New Principles for Numerical Analysis by Adaptive Precision, Automatic Tuning & Accuracy Verification
  - Hierarchical Data Driven Approach (*h*DDA) based on Machine Learning
  - Software & Utilities for Heterogenous Environment, such as Wisteria/BDEC-01



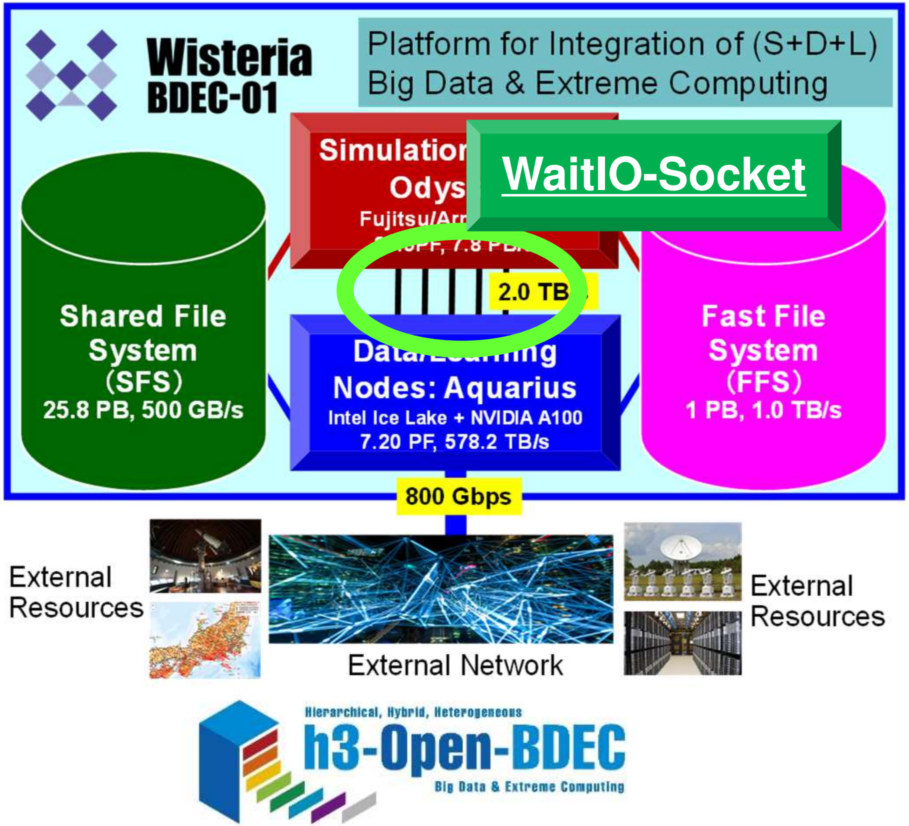
Wisteria/BDEC-01: The First "Really Heterogenous" System in the World





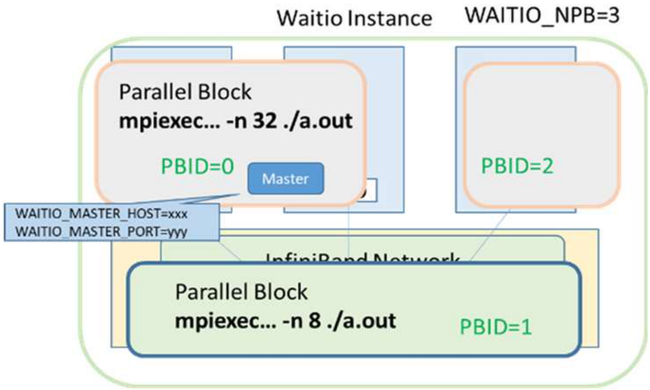
h3-Open-SYS/WaitIO-Socket

- Wisteria/BDEC-01
  - Aquarius (GPU: NVIDIA A100)
  - Odyssey (CPU: A64FX)
- Combining Odyssey-Aquarius
  - Single MPI Job over O-A is impossible
- **Connection between Odyssey-Aquarius**
  - **IB-EDR with 2TB/sec.**
  - **Fast File System**
  - **h3-Open-SYS/WaitIO-Socket**
    - **Library for Inter-Process Communication through IB-EDR with MPI-like interface**



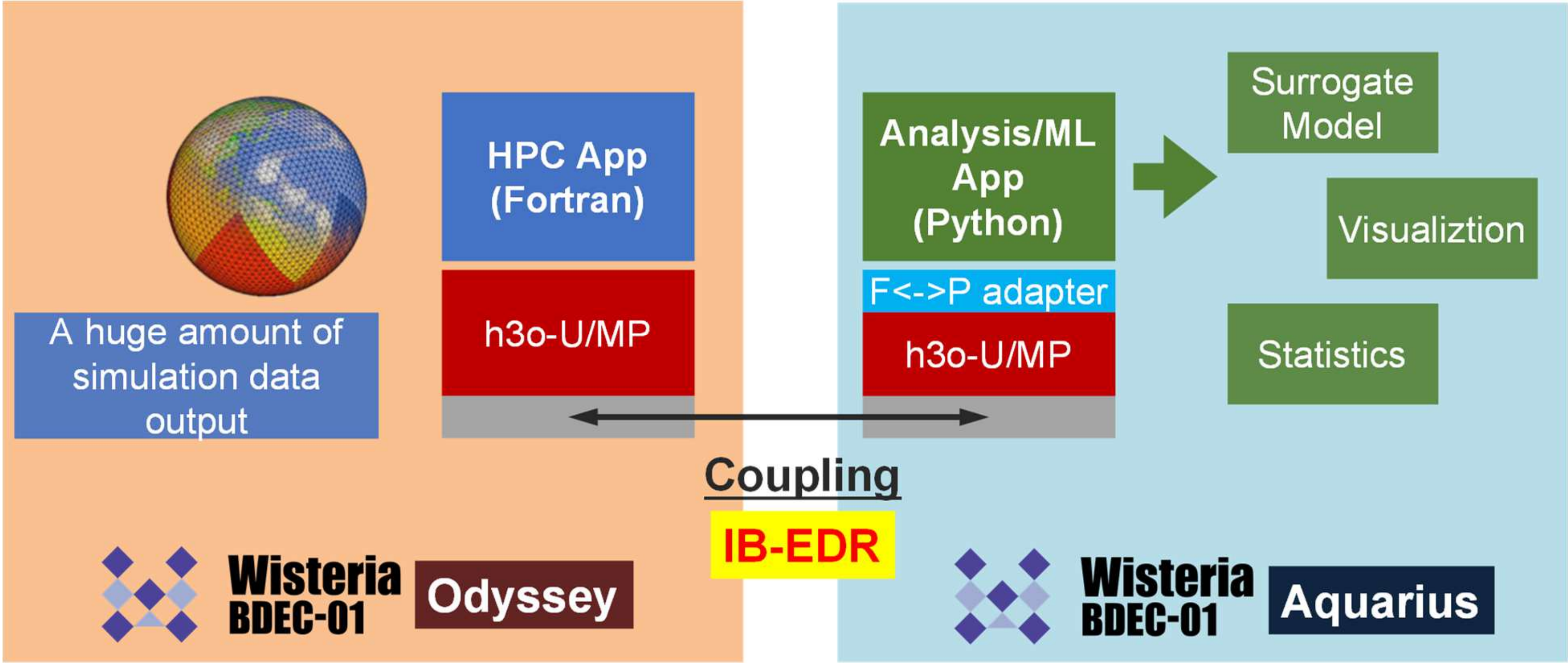
API of h3-Open-SYS/WaitIO-Socket PB (Parallel Block): Each Application

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



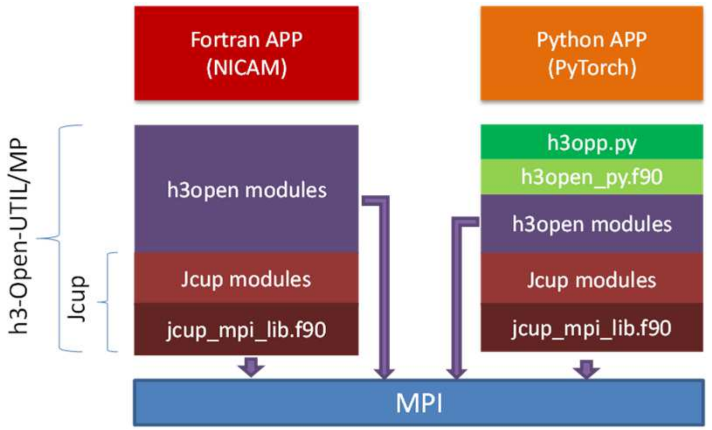
[Sumimoto et al. 2021]

h3-Open-UTIL/MP (h3o-U/MP) + h3-Open-SYS/WaitIO-Socket



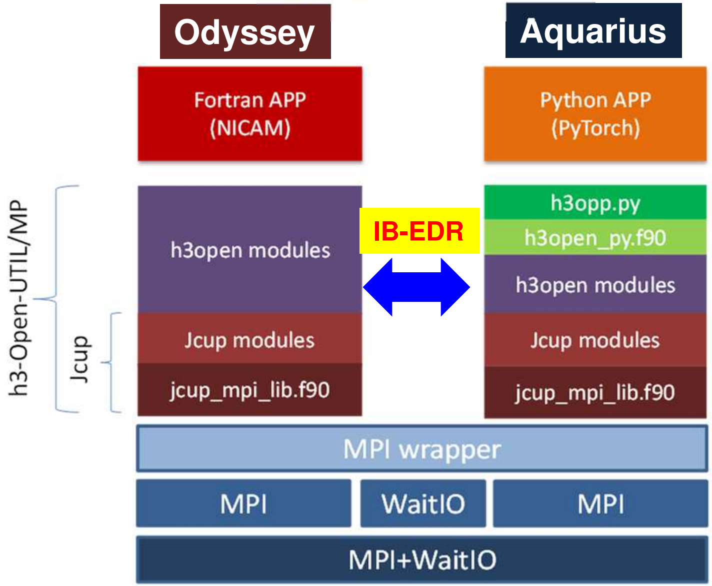
h3-Open-UTIL/MP + h3-Open-SYS/WaitIO-Socket

Available in June 2022



May 2021: MPI Only

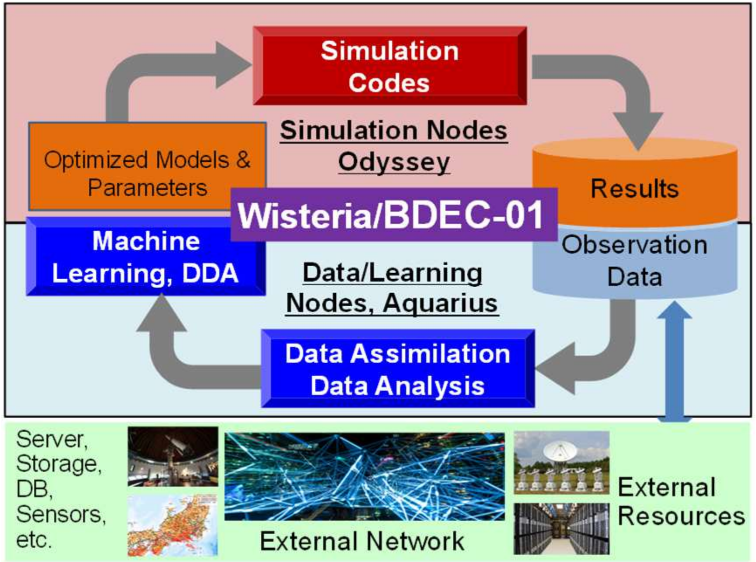
Wisteria BDEC-01

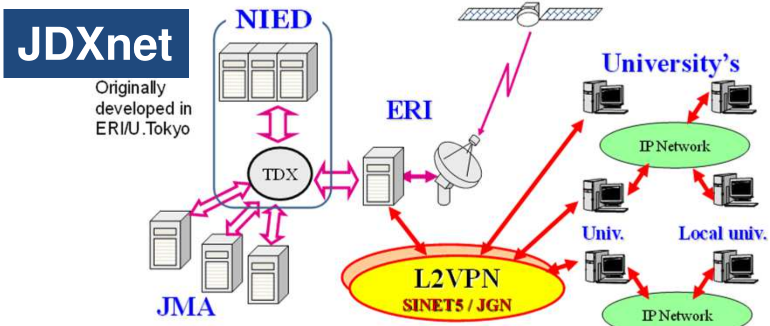


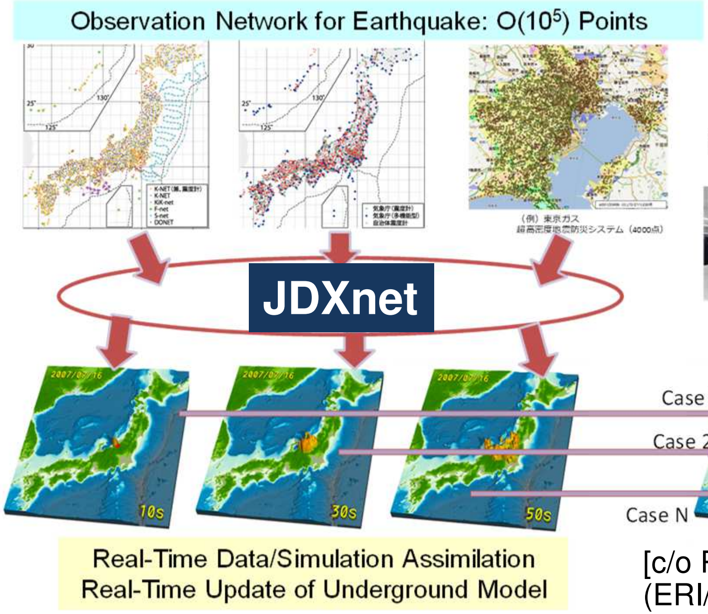
June 2022: Coupler＋WaitIO

3D Earthquake Simulation with Real-Time Data Observation/Assimilation

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







[c/o Furumura]

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

Real-Time Assimilation of "Observation+Computation" in Seismic Wave Propagation [c/o Oba & Furumura]

- **Data Assimilation of Wave Propagation by "Optimal Interpolation Technique"**

$$\underset{\text{Assim.}}{x_n^a} = \underset{\text{Comp.}}{x_n^f} + W(\overbrace{\underset{\text{Obs.}}{y_n} - \underset{\text{Comp.}}{H x_n^f}}^{\text{Residual}})$$

$$\underset{\text{Comp.}}{x_{n+1}^f} = F \underset{\text{Assim.}}{x_n^a}$$

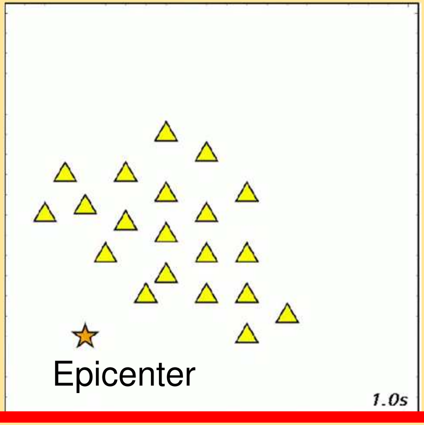
n: Time Step

W: **Weighting Matrix**

F: Wave Propagation simulation

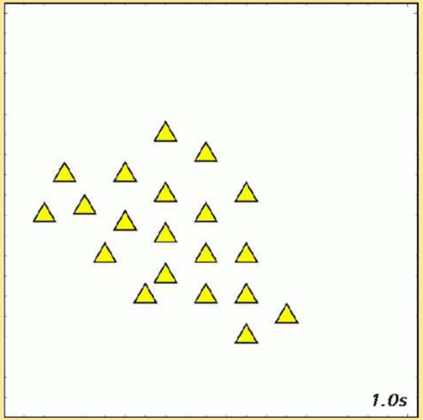
(A) Pure Simulation

△ : Obs. Pts.



(B) Assimilation+Sim.

(No info for Epicenter needed)

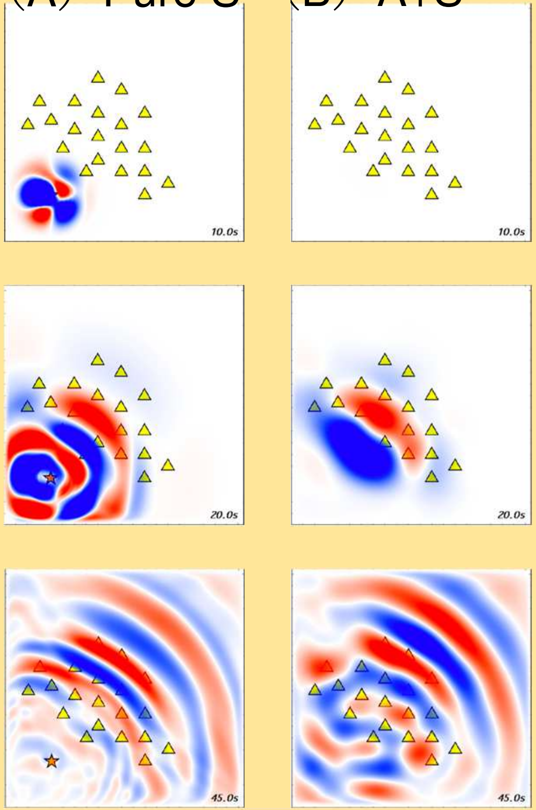


(A) Pure S (B) A+S

10 sec.

20 sec.

30 sec.



Real-Time Assimilation of "Observation+Computation" in Seismic Wave Propagation [c/o Oba & Furumura]

- **Data Assimilation of Wave Propagation by "Optimal Interpolation Technique"**

Residual

Assim. Comp. Obs. Comp.

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

n: Time Step

$\boldsymbol{W}$: **Weighting Matrix**

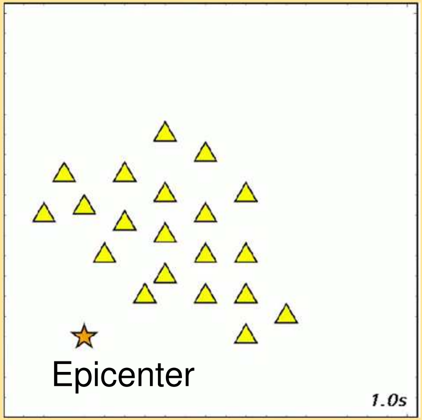
Comp. Assim.

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

$\boldsymbol{F}$: Wave Propagation simulation

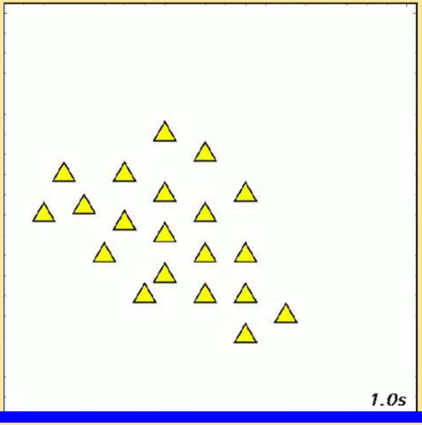
(A) Pure Simulation

△: Obs. Pts



(B) Assimilation+Sim.

(No info for Epicenter needed)

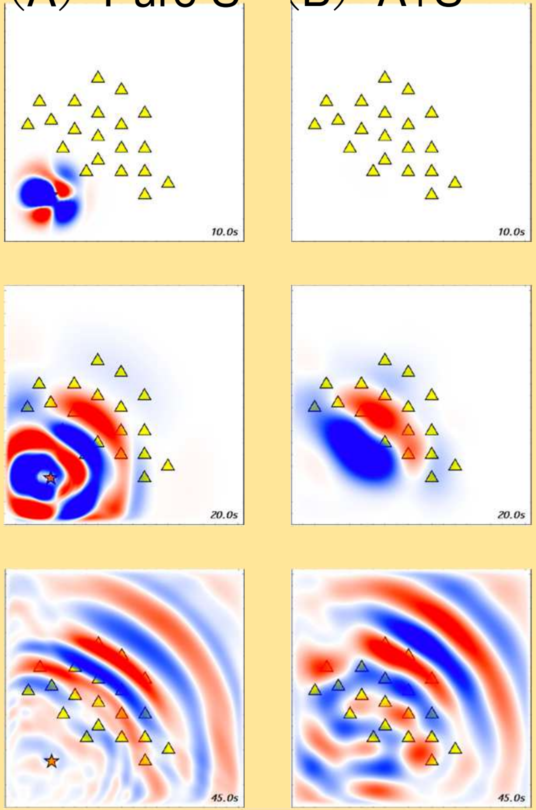


(A) Pure S (B) A+S

10 sec.

20 sec.

30 sec.



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

Assim. Comp. Obs. Residual Comp.

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

Comp. Assim.

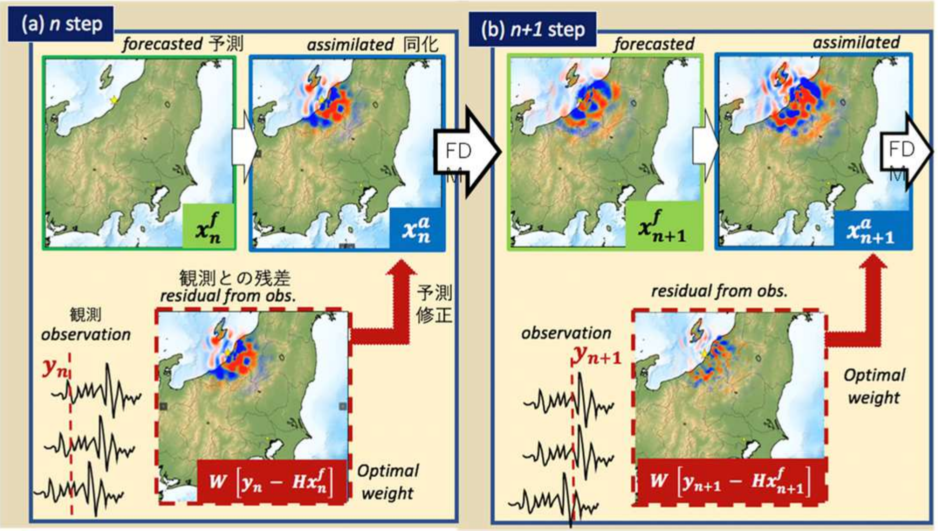
$$x_{n+1}^f = Fx_n^a$$

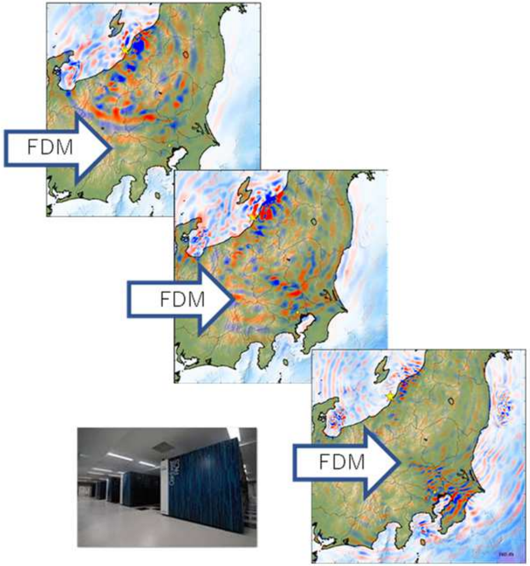
n: Time Step

W: **Weighting Matrix**

F: Wave Propagation simulation

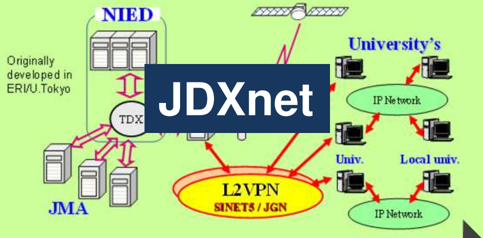
(A+S) Assimilation+Simulation → **(Pure S) Pure Simulation/Forecast**





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

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



Observed Data

SINET6

Obs. Data

External Server for Observed Data

Wisteria BDEC-01

Data/Learning Nodes: Aquarius
Intel Ice Lake + NVIDIA A100
7.20 PF, 578.2 TB/s

2.0 TB/s

Simulation Nodes: Odyssey
Fujitsu/Arm A64FX
25.9PF, 7.8 PB/s

Output & Movie

Filter

Visualizer

h3-Open-SYS/WaitIO-Socket

Seism3D/OpenSWPC-DAF

Mesh Data

UG Model

Communications by WaitIO-Socket

[Kasai et al. 2021]

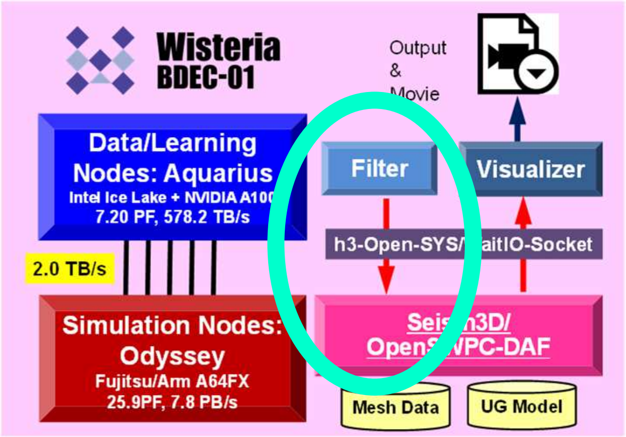
Aquarius: SEND

```
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
```

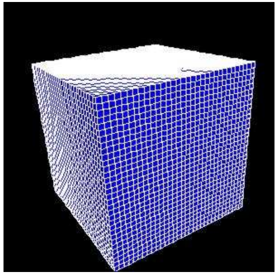
Odyssey: RECV

```
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,...)
```



Example: Off Niigata 2007 Mw6.6 Earthquake

- Observed Data: Stored in External Server
- Aquarius receives observed data, and apply filtering
- “Data Assimilation + Simulation (A+S)”, and “Forecast by Simulation (Pure S)” are separated codes, while same number of computing nodes were used on Odyssey
- Movies were created after simulations (O(10) sec.)



- Seism3D/OpenSWPC-DAF
  - 3D FDM + Optimal Interpolation Technique for Data Assimilation
  - Each Mesh: 240m × 240m × 240m
  - 1,920 × 1,920 × 240 meshes (8.85 × 10^8)
  - 460.8 km × 460.8 km × 57.6 km

$$v_p^n = v_p^{n-1} + \frac{1}{\rho}\left(\frac{\partial \sigma_{xp}^{n-1/2}}{\partial x} + \frac{\partial \sigma_{yp}^{n-1/2}}{\partial y} + \frac{\partial \sigma_{zp}^{n-1/2}}{\partial z}\right)\Delta t \quad (p = x, y, z)$$

Assim. Comp. Obs. Residual Comp.

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

n: Time Step

$\boldsymbol{W}$: Weighting Matrix

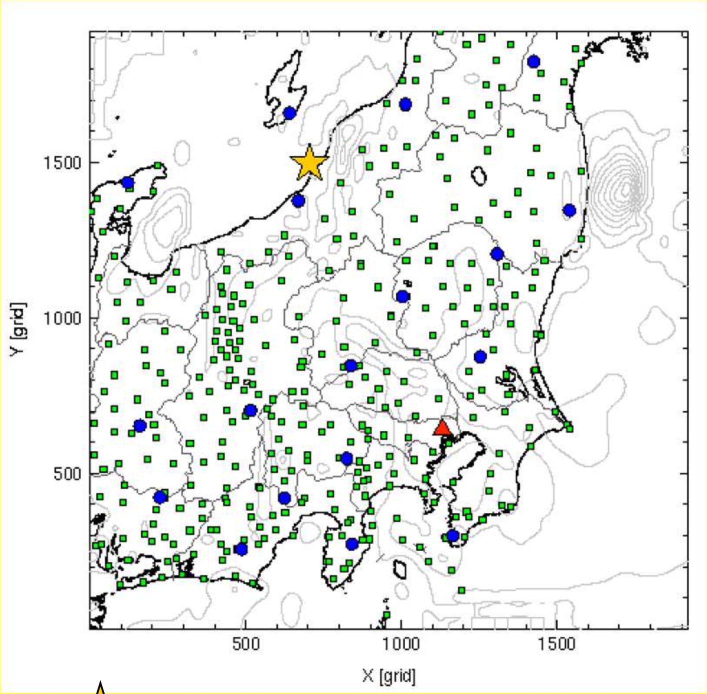
Comp. Assim.

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

$\boldsymbol{F}$: Wave Propagation simulation

Off Niigata 2007 Mw6.6 Earthquake

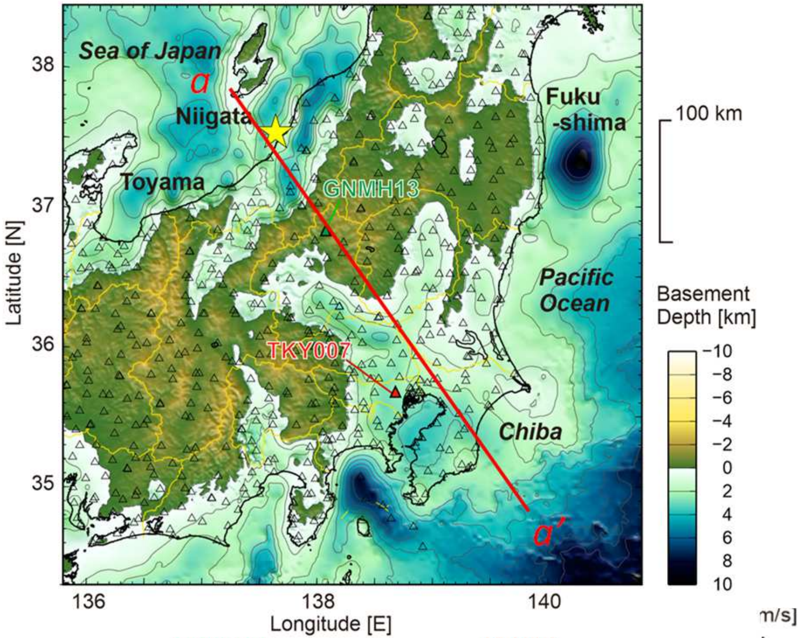
[c/o Prof. T. Furumura, ERI/U.Tokyo]

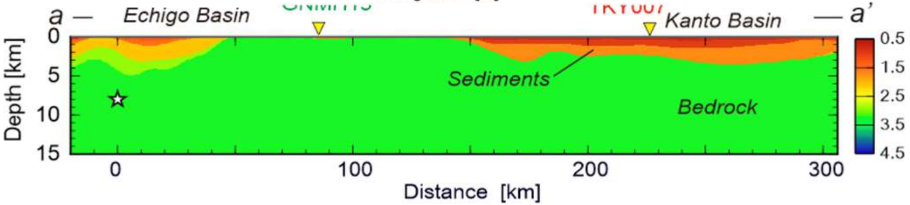


Epicenter

Hi-net (Short Period) 349 pts

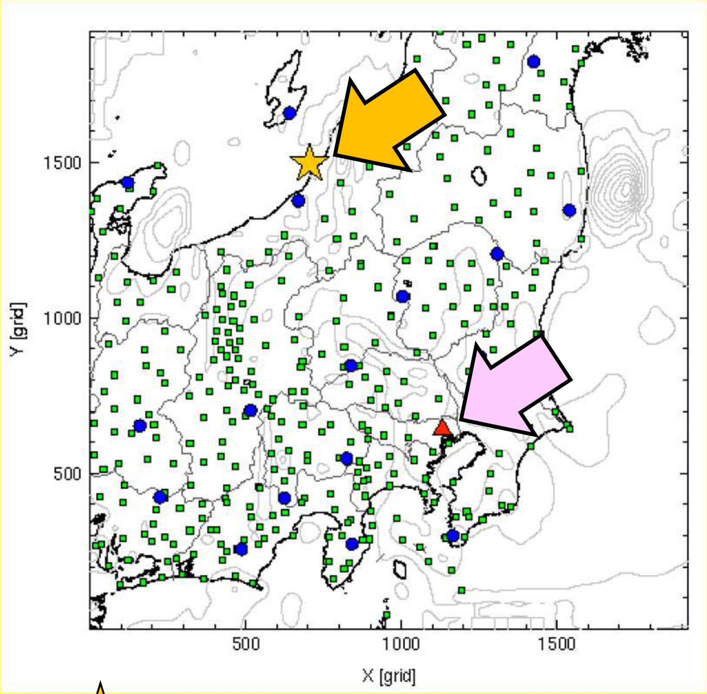
F-net (Broadband) 18 pts





Off Niigata 2007 Mw6.6 Earthquake

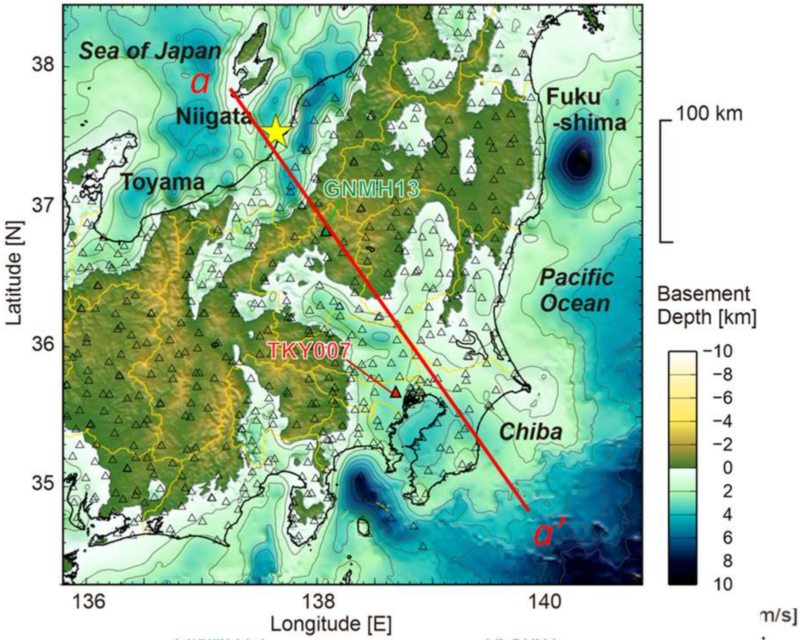
[c/o Prof. T. Furumura, ERI/U.Tokyo]

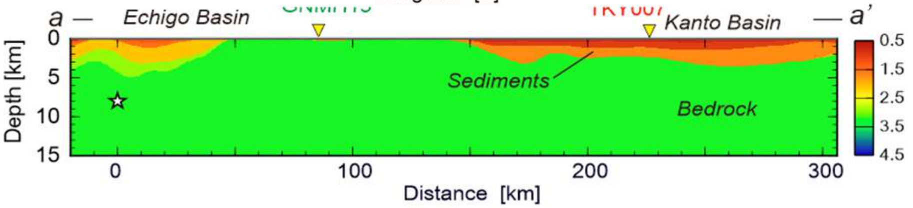


☆ Epicenter

■ Hi-net (Short Period) 349 pts

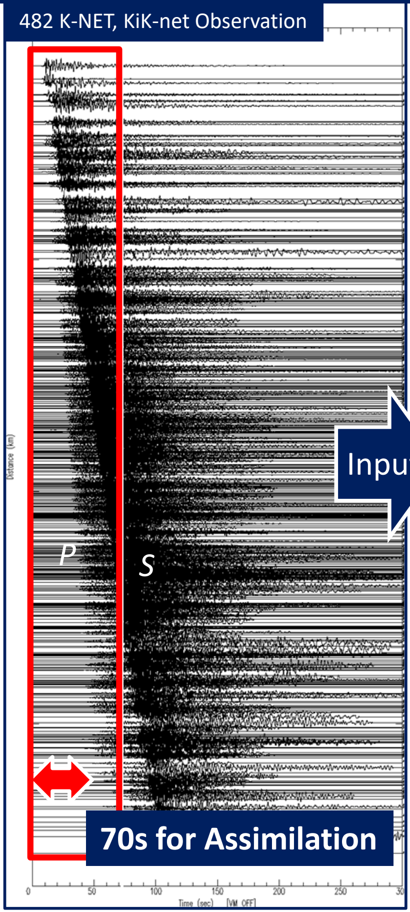
● F-net (Broadband) 18 pts



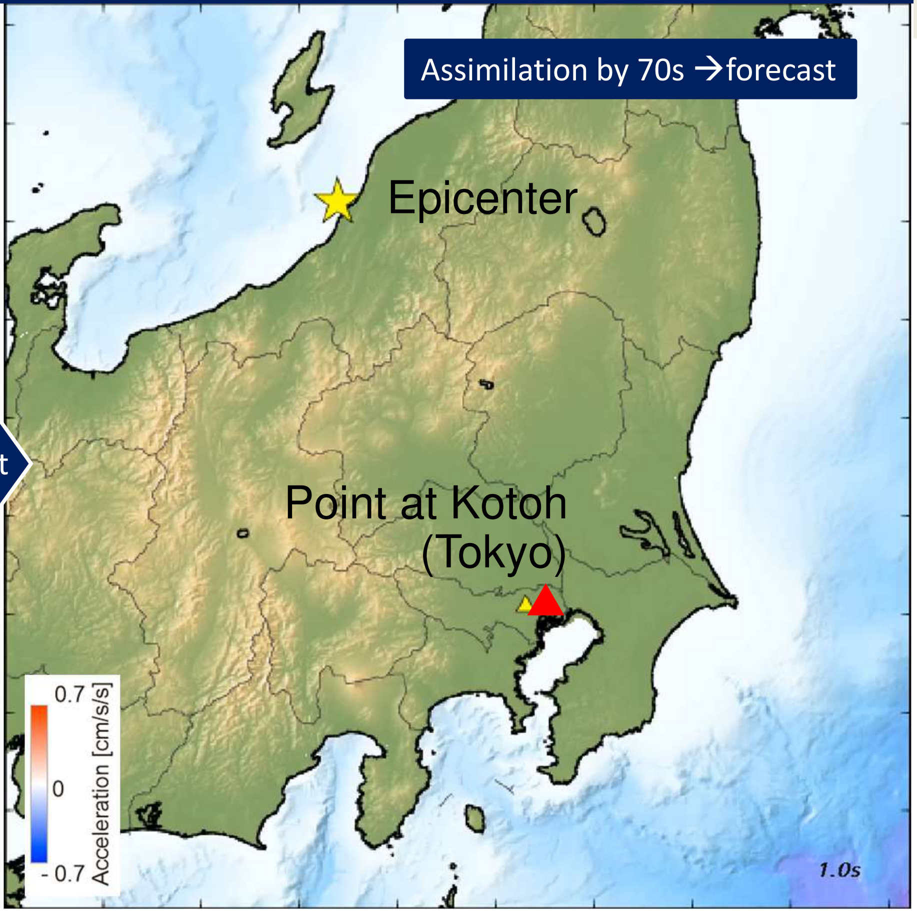


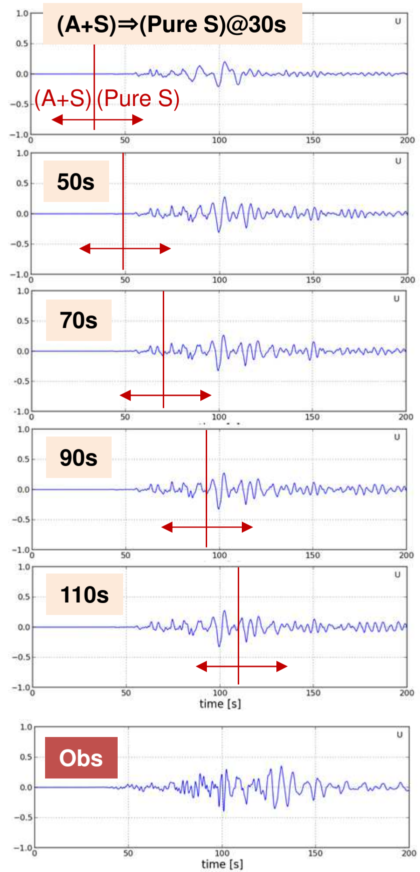
Data Assimilation + Pure Simulation/Forecast

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

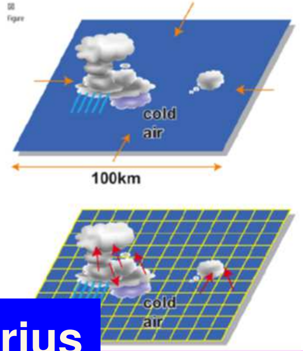


Input



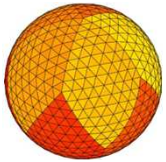


Atmosphere-ML Coupling
[Yashiro (NIES), Arakawa (ClimTech/U.Tokyo)]



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

Odyssey



Phys

High Resolution Atmospheric Model
(Convection-Resolving Mode)

75%

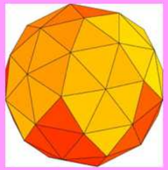
ML App
(Python)

h3-Open-UTIL/MP
(Coupler)
+
h3-Open-SYS/WaitIO-Socket

Training with high-resolution
NICAM data

Replacing Physical Process
in Low-Resolution NICAM
with Machine Learning

Aquarius



ow Resolution Atmospheric Model
onvection-Parameterization Mode)

Replacing this part with AI

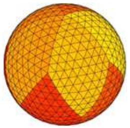
25% → **~0%**

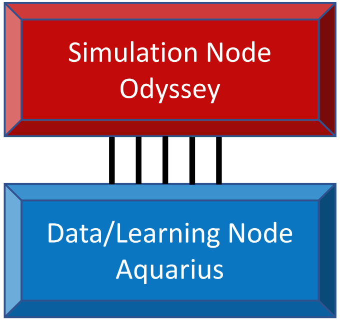
Diagram of applying ML to an atmospheric model

Experimental Design

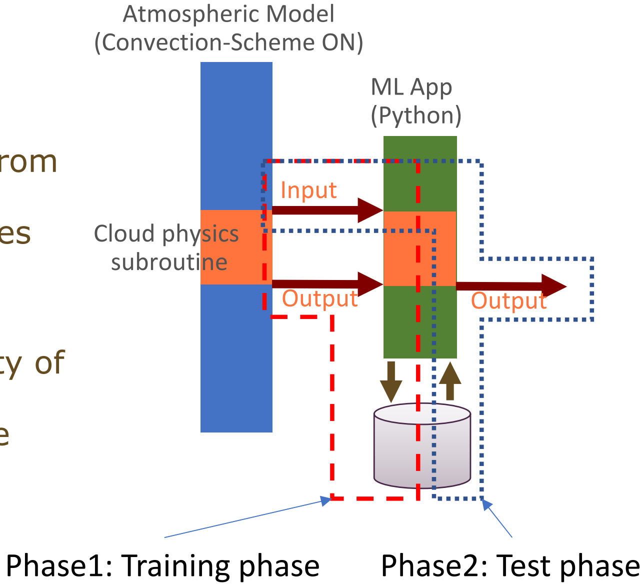
- Atmospheric model on Odyssey
  - NICAM : global non-hydrostatic model with an icosahedral grid
  - Resolution : horizontal : 10240, vertical : 78
- ML on Aquarius
  - Framework : PyTorch
  - Method : Three-Layer MLP
  - Resolution : horizontal : 10240, vertical : 78
- Experimental design
  - Phase1: PyTorch is trained to reproduce output variables from input variables of cloud physics subroutine.
  - Phase2:Reproduce the output variables from Input variables and training results
- Training data
  - Input : total air density (rho), internal energy (ein), density of water vapor (rho_q)
  - Output : tendencies of input variables computed within the cloud physics subroutine

$$\frac{\Delta rho}{\Delta T} \quad \frac{\Delta ein}{\Delta T} \quad \frac{\Delta rho\_q}{\Delta T}$$



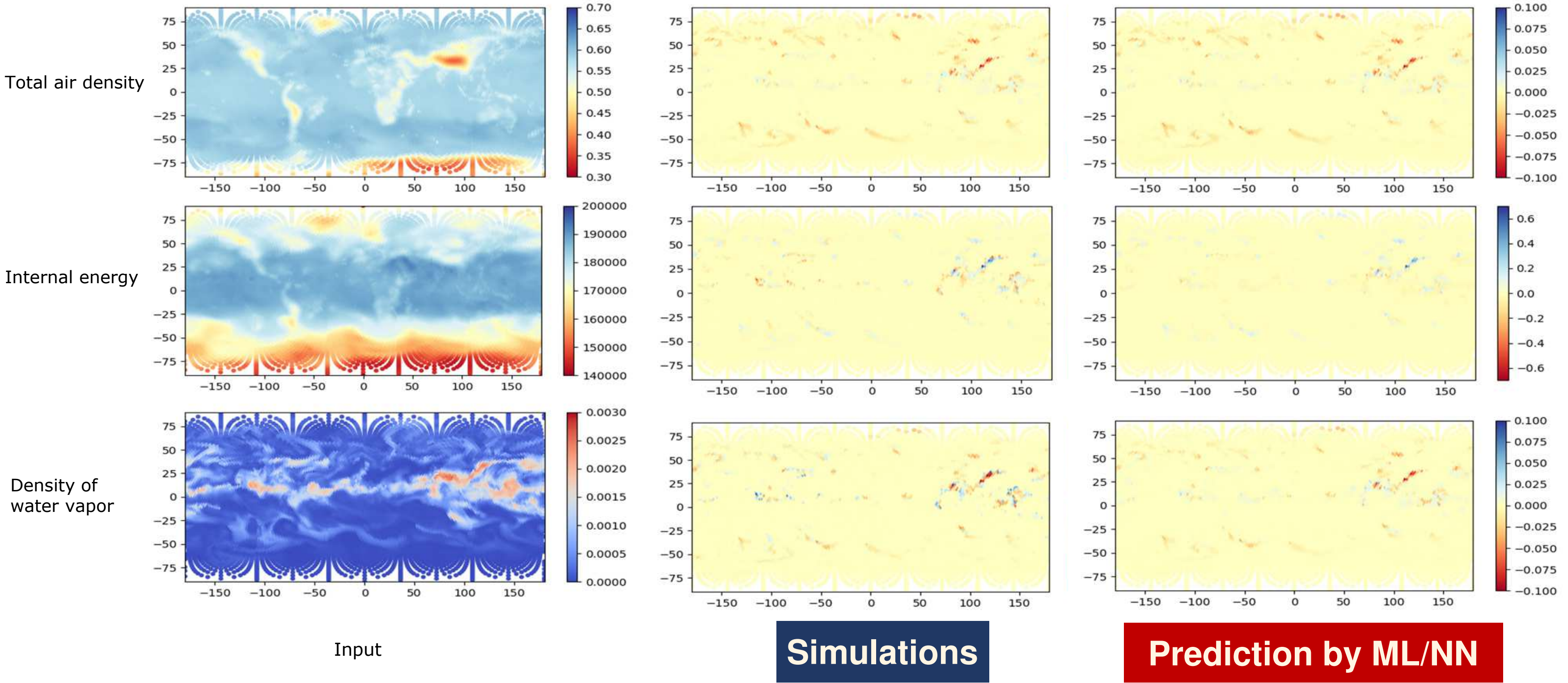


PyTorch



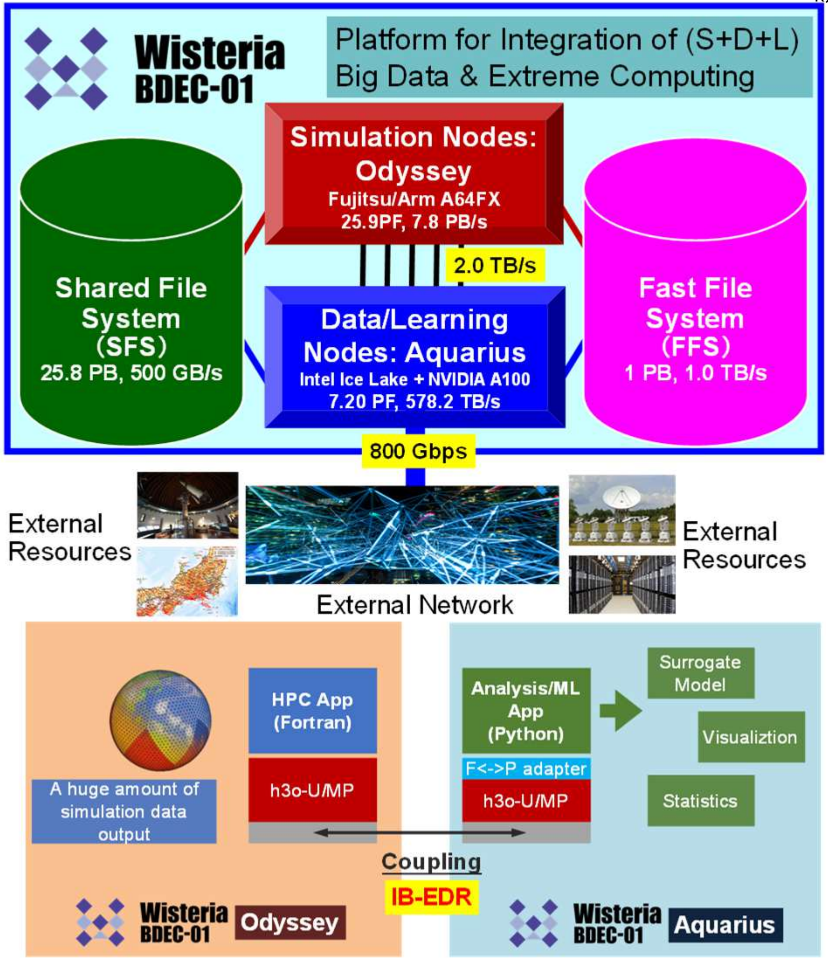
Test calculation

- 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



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 for scheduling needed



Summary

- Wisteria/BDEC-01
- h3-Open-BDEC
- Integration of (S+D+L)
  - Earthquake Simulation + Real-Time Data Assimilation
  - Coupling of Atmosphere Simulations & AI