Current status of the project "Toward a unified view of the universe: from large scale structures to planets"

Jun Makino

Center for Planetary Science/Department of Planetology
Kobe University

Feb 7 2022 4th R-CCS International Symposium

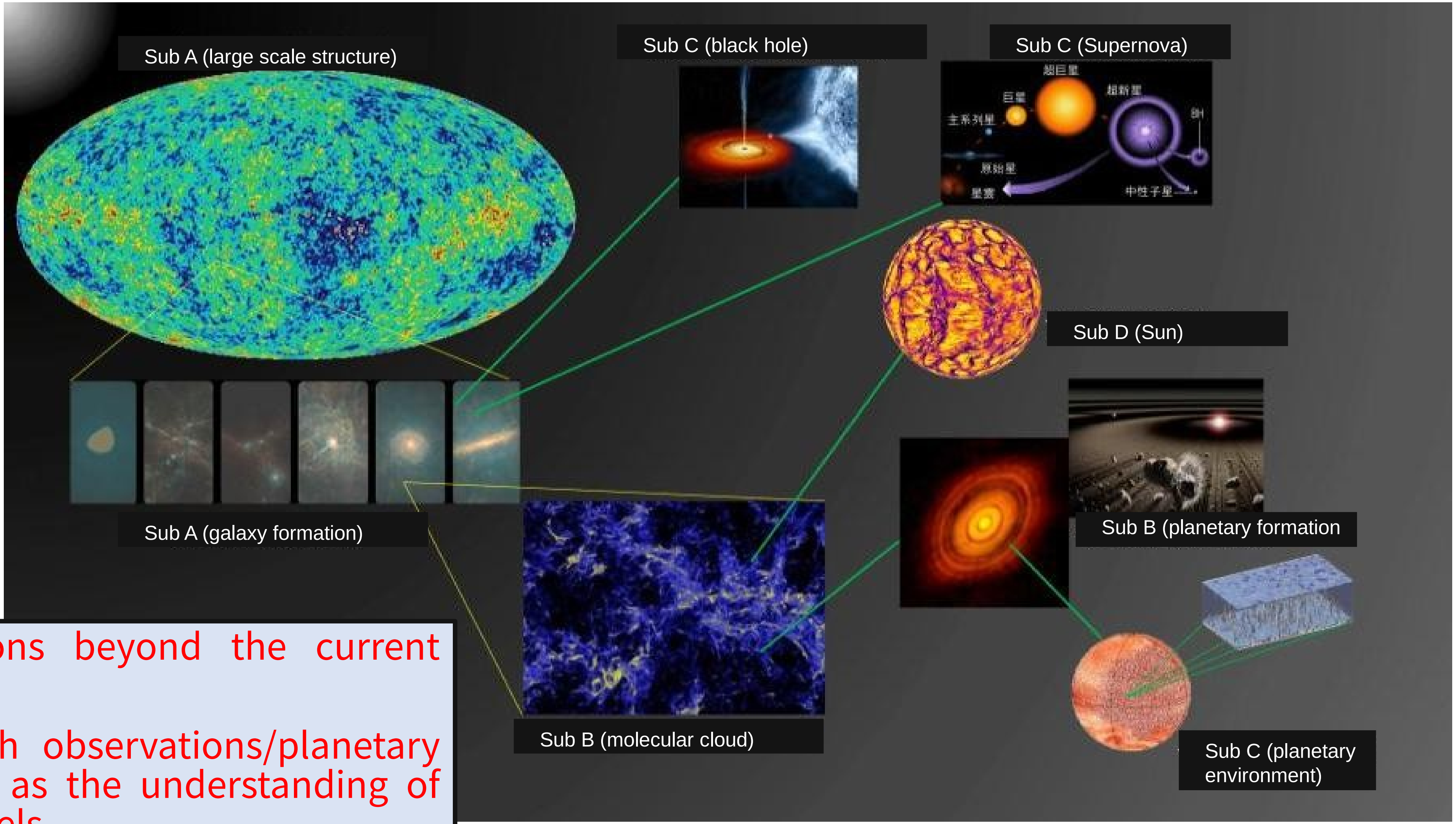
Talk Plan

1. Goal and Overview of the project
2. Project Structure
3. Overview of the result
   1. Sub Project A
   2. Sub Project B
   3. Sub Project C
   4. Sub Project D
4. A word on post-Fugaku

1． Goal and Project Overview

Goal: Construct a holistic and unified understanding of the formation and evolution of hierarchical structures in the universe by combining multi-level simulations with the latest observations

Simulations with sufficient resolution at each level of the universe have just becom possible with the supercomputer "Fugaku" and the software developed in Priority 9 and Budding Researchers Program 3.



Perform the simulations beyond the current status of the art

Direct comparison with observations/planetary mission data as well as the understanding of interaction between levels

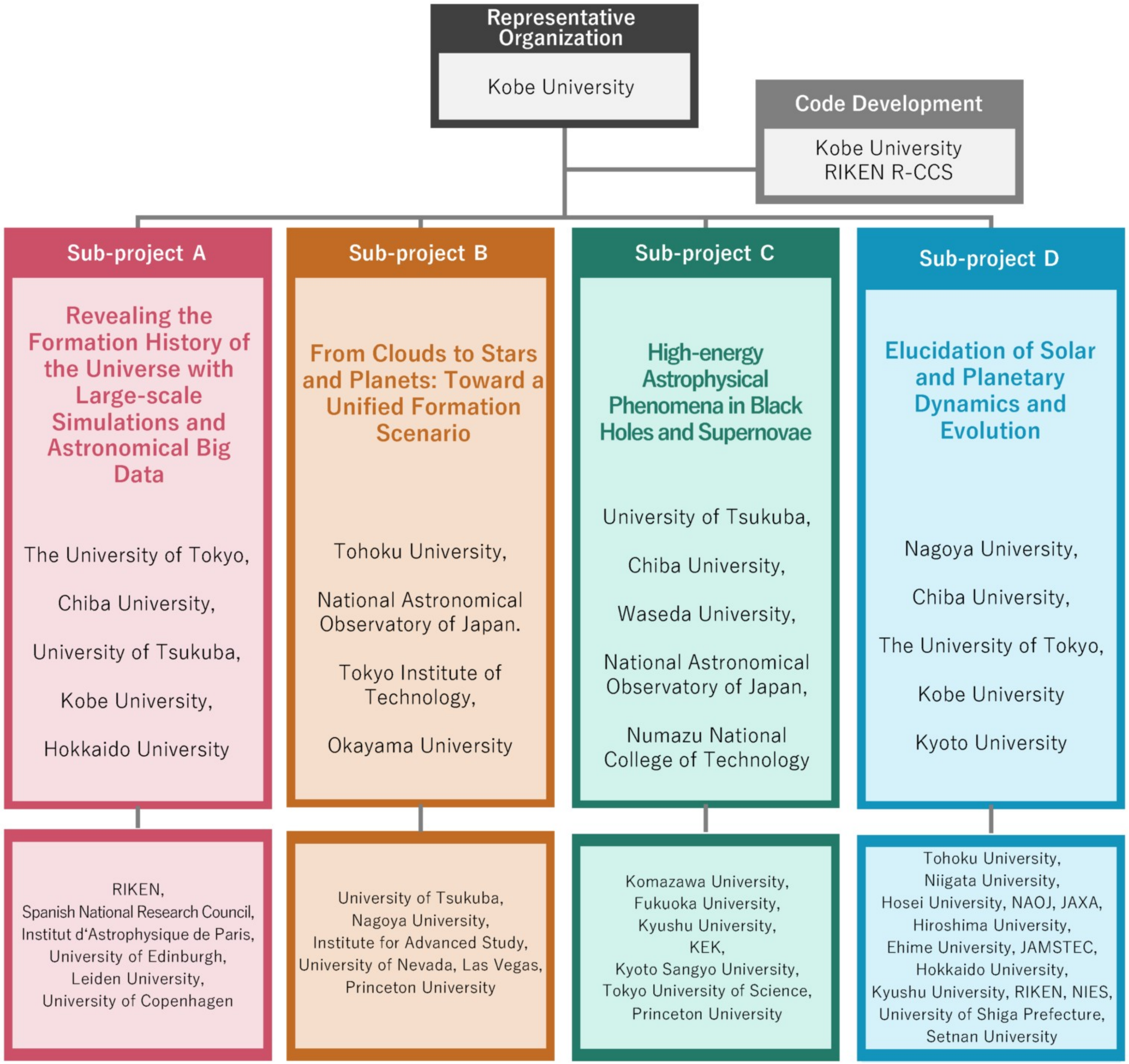
2. Project Structure

“All-Japan” team of simulation reserchers of astrophysics and planetary science

21 research institutes

97 researchers

Toward a unified view of the universe: from large scale structures to planets



3． Result Overview

3.1 Sub A

Goals:

- Dark matter halo simulation: Perform 50mpc/h cube simulations with resolution high enough to resolve minihaloes in which first-generation stars form, and follow the evolution down to z=10.
- Understanding of the impact of neutrinos on large-scale structure formation in the universe: Perform Vlasov simulations to study the effecto of neutrinos
- Galaxy-formation Simulation with star-by-star resolution: Use a new code ASURA-FDPS to achieve the star-by-star resolution for galaxy formation simulation
- Formation process of compact binaries in high-density star clusters: Use new codes with new P3T algorithms to perform 3M-star realistic simulation of globular clusters.

Status

- Dark matter halo simulation: Runs finished
- Understanding of the impact of neutrinos on large-scale structure formation in the universe: Several runs finished and being analyzed
- Galaxy-formation Simulation with star-by-star resolution: Dynamics code ready to run on Fugaku and code with subphysics under testing.
- Formation process of compact binaries in high-dencity star clusters: Production runs are running,

3.2 Sub B

- Goals:
- Formation of molecular clouds and cloud cores: Perform high-resolution MHD simulations to investigate the formation process of molecular clouds
- Global magnetohydrodynamic simulations of protoplanetary disks: Investigate the structure of turbulence in disks by high-resolution MHD simulations taking into account the distribution of dust particles in protoplanetary disks and non-ideal magnetohydrodynamic effects
- Growth of planetecimals: Perform large-scale N-body simulations and SPH simulations to study the formation process of planetary systems.
- Dust growth in turbulence in protoplanetary disks: Use DNS simulation to understand the early stage of dust growth in proroplanetary disks.

- Achievements:.
- Formation of molecular clouds and cloud cores: The Athena++ code has been ported to Fugaku using uTofe libraries and achieved good scaling up to 16K nodes.
- Global magnetohydrodynamic simulations of protoplanetary disks: (used the same Athena++ code)
- Growth of planetecimals: GPLUM code based on P3T algorithm has been prdted to Fugaku and achieve good scaling up to 1k nodes. Also FDPS-SPH code has been ported to Fugaku.
- Dust growth in turbulence of protoplanetary disks: Total 40963 particle tracks in a large scale direct numerical simulation of turbulence (40963 grid points)

3.3 Sub C

- Goals:.
- General relativistic radiation magnetohydrodynamic code INAZUMA: Study the effect of angular resolution of radiaion
- Higher-order MHD code CANS+: Perform Simulations of low-luminocity accretion disks to study the effect of spatial resolutio
- PIC calculations: Perform the world's first three-dimensional simulation of ion-electro systems
- GR radiation transport calculation with RAIKOU: Clarify the elationship between radio and gamma-ray sources.
- First-principles simulation of neutrino transport in supanova: Develop a calculation method to suppress numerical instabilities around the coordinate axes. Extend computational domain to 5,000 km, and calculation time to 50 ms after the explosion.
- 3DnSNe: Perform long-time calculations of supernova explosions.

- Results
- INAZUMA: The need of anular resolution for radiation has been verified.
- CANS+: Performance on Fugaku improved by a factor of three.
- PIC calculation: The world's first 3D calculation of ion-electron systems was successfully performed.
- RAIKOU: confirmed the misalinment of radio and gamma-ray sources
- Neutorino transport: succeeded to extend the simulation time to 50ms.
- 3DnSNe: Developed a method to extend the time step by using non-uniformangular grid. By this method, succeeded in the 3D calculation of the iron core in the Si-O-burning phase.

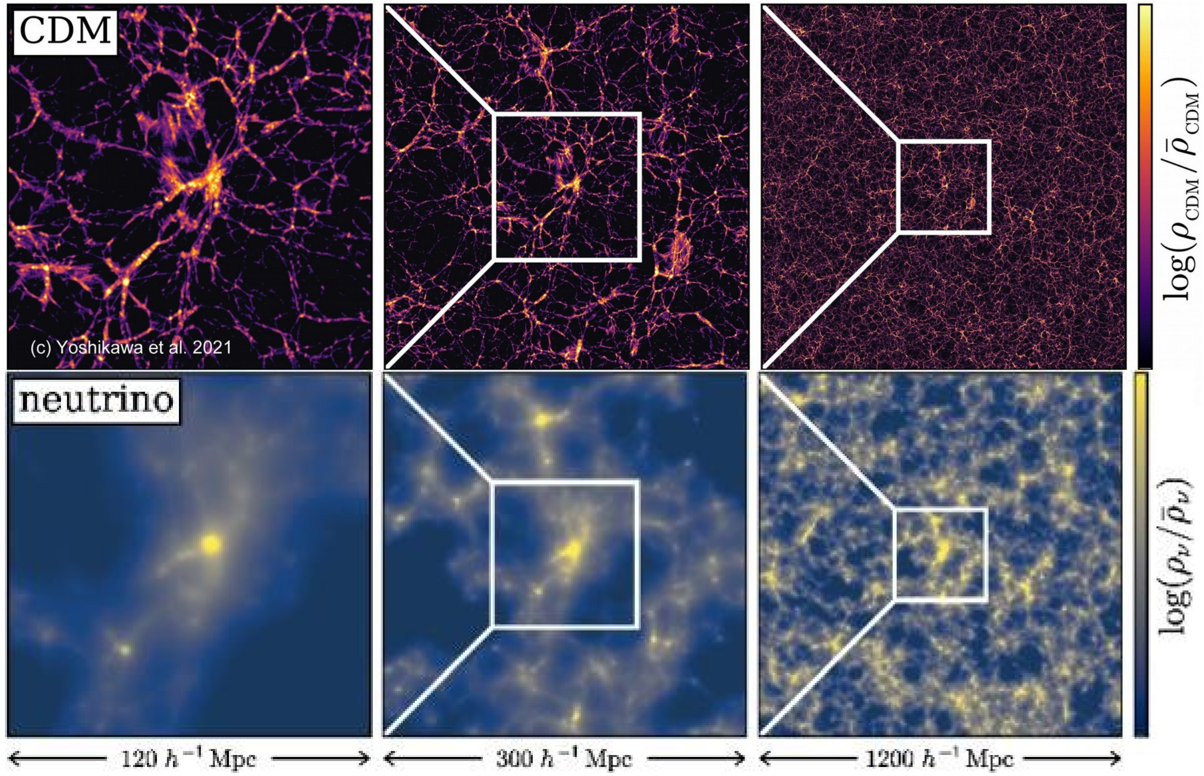
3.4 Sub D

- Goals:.
  - Perform the world's largest global calculation of the solar convection layer to reproduce the velocity structure
  - Introduce more realistic radiative heating to the atmosphere of Venus and and test the global non-hydrostatic model.
  - Perform 3D mantle convection of Earth and Moon on Fugaku.
  - Perform, for the first time, global simulation of the atmosphere of giant gas-planet
  - using an inelastic rotating spherical-shell model.
- Achievements
  - Successfully reproduced the differential rotation of the Sun which is consistent with observations using a simulation with 5 billion grid points. Solved the "convective conundrum" in solar physics (Hotta & Kusano, 2021, Nature Astronomy).
  - Succeeded in simulating the solar convection layer with 12.8 billion points and for 12 million time steps, reproducing the observations more closely.
  - Developed the atmosphere model of Venus that includes solar heating with diurnal changes, and succeeded in global non-hydrostatic calculations including thermal tidal waves. A higher resolution calculation was also performed, and the observed north-south asymmetric structure was reprodiced.
  - The spherical shell version of AcuTEMAN (Kameyama et al., 2005), yACuTEMAN, was ported to Fugaku.
  - The performance of inelastic rotating spherical-shell code has been improved.
- Tuning results: solar convection layer 10% of theoretical peak performance, spherical harmonic function library 43%.

3.5 Main results(1)

Sub A

Understanding of the impact of neutrinos on large-scale structure formation in the universe:
Achieved the weak- and strong-scaling efficiency of 82%, and completed the calculation similar to that performed on Tianhe-2 with much better quality in 1/9 of wallclock time, by using the full system of Fugaku. This result was selected as one of the finalists of 2021 Gordon Bell Prize

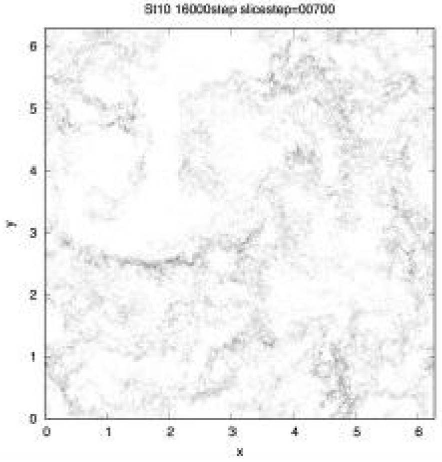


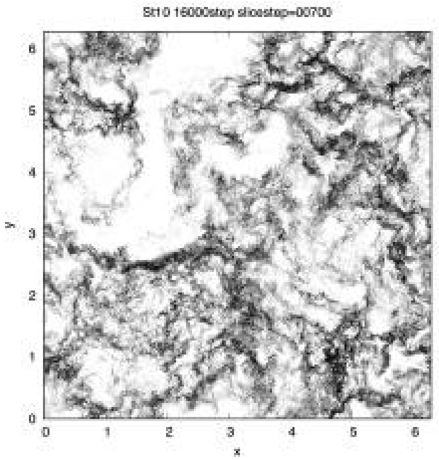
DM and Neutrino distribution from Fugaku full-system run

Sub B

Dust growth in turbulence of protoplanetary disks
Total 4096^3 particle tracks in a large scale direct numerical simulation of turbulence (4096^3 grid points). Data base of dust collisional growth in high-Reynolds number (Re=36500) turbulence has been built.

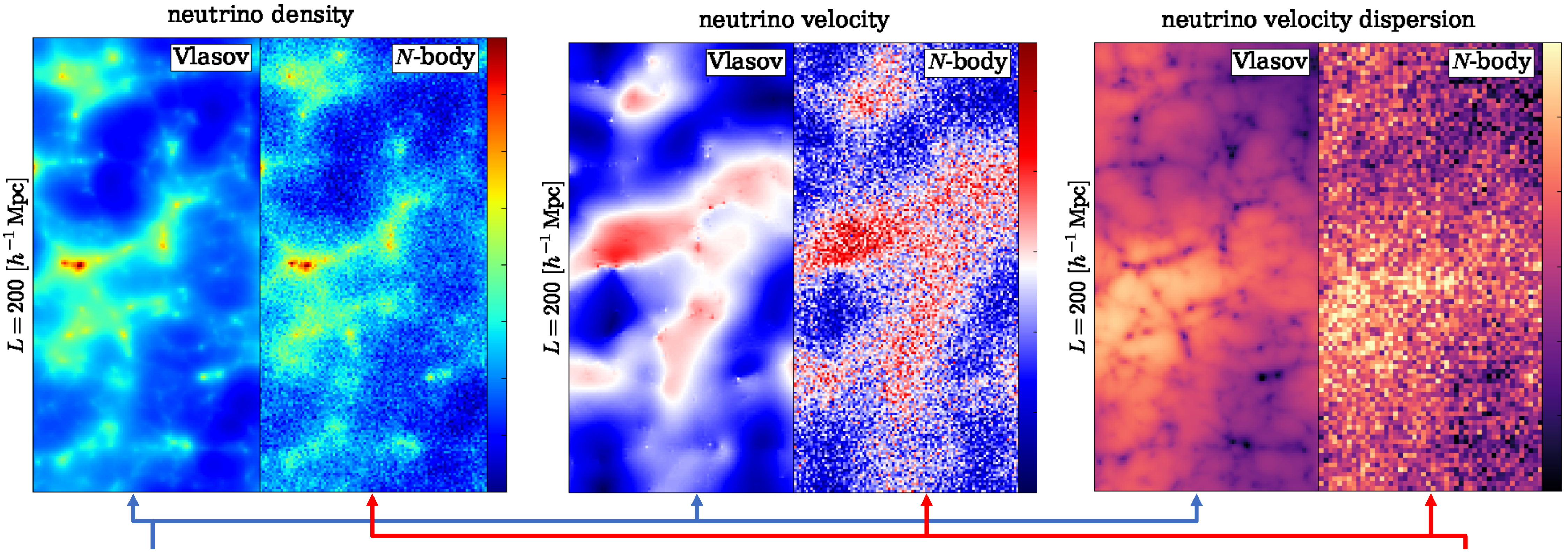
Before→:
2048^3 grids and
512^3particles. Not enough collisions to follow growth





←After:
2048^3 grids and
1024^3 particles Sufficient nnumber of collisions.

Comparison between N-body and Vlasov Simulations



- neutrinos simulated with a Vlasov simulation

# of dark matter particles : $N_{\mathrm{CDM}}=768^3$

# of mesh grids for Vlasov simulation : $N_x=192^3$, $N_v=64^3$

- neutrinos simulated with an N-body simulation

# of dark matter particles : $N_{\mathrm{CDM}}=768^3$

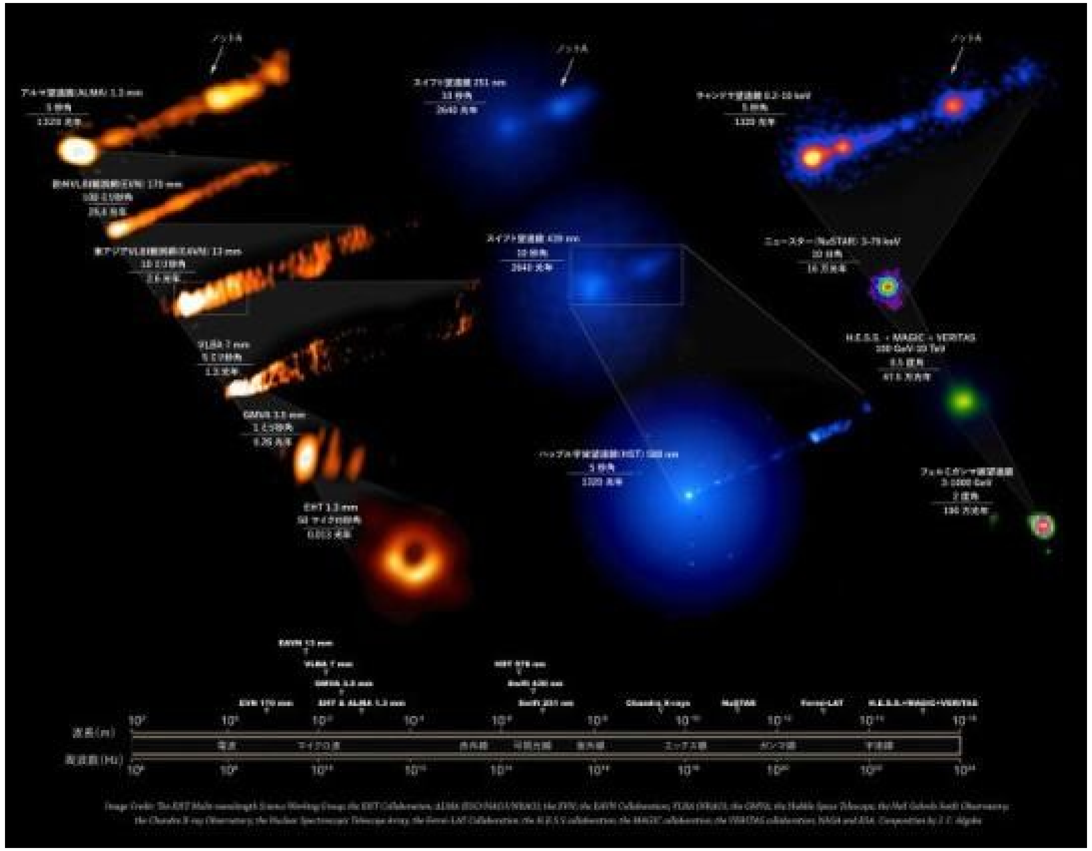
# of neutrino particles : $N_\nu = 8\ N_{\mathrm{CDM}} = 1536^3$

3.5 Main Results(2)

Sub C

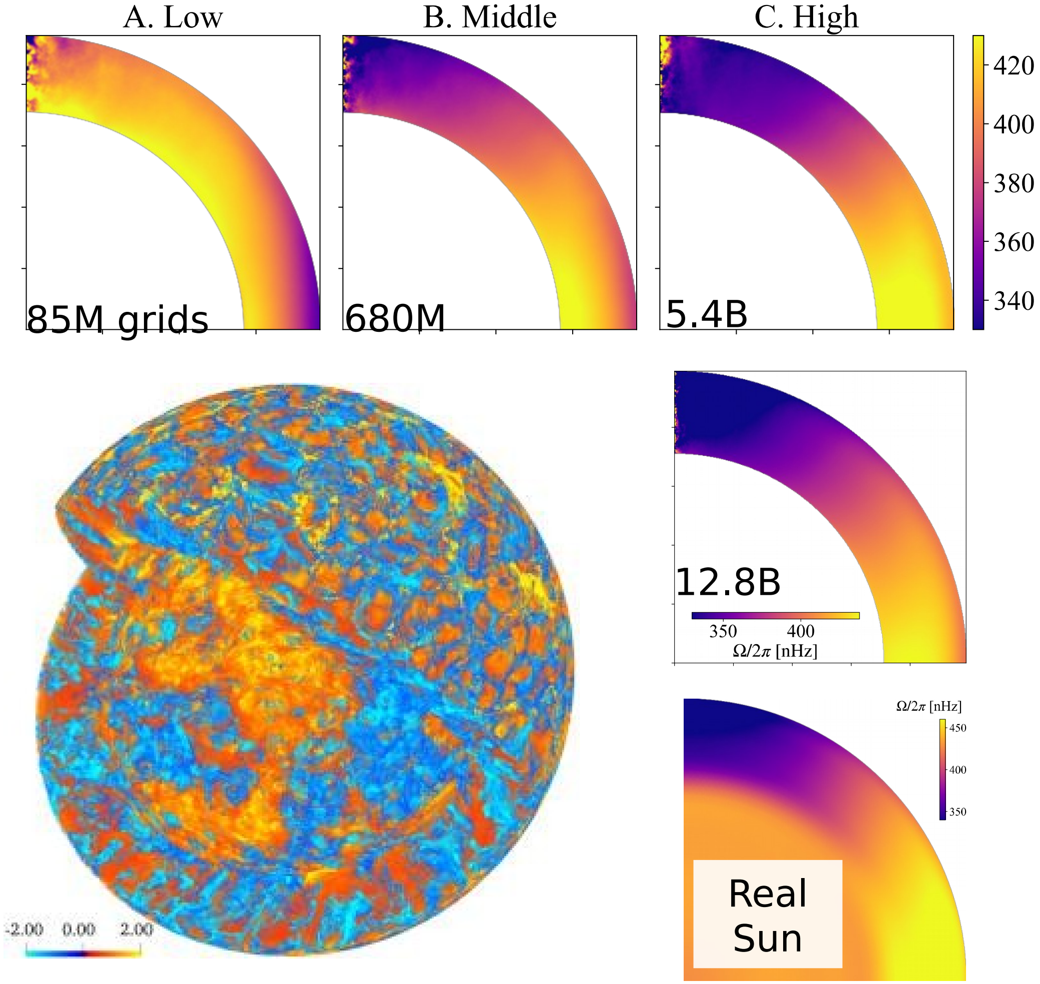
GR radiation transport calculation with RAIKOURAIKOU

RAIKOU simulation and multi-band observation using Event Horizon Telescope and other instruments demonstrated that M87 radio and gamma-ray sources are misaligned (EHT MWL Science Working Group et al. 2021)



Sub D

- **Sun**



Successfully reproduced the differential rotation of the Sun for the first time. (Reduced Speed of sound method, new finite-difference scheme etc)

Current Status of Fugaku and Issues

- Not limited to our project but for all users (and developers) of Fugaku
- My position is a bit special: Part of RIKEN development team (team leader of Co-design team), and the leader of one of “Program for Promoting Researches on the Supercomputer Fugaku”
- Maybe useful for current users and also for post-Fugaku project

Is Fugaku easy to use?

- What developers say(I’m part of the team):
  - Because the K computer utilized the basic design of the Fujitsu CPU (SPARC64), only a limited set of basic software (OS) could be used. Users had to convert or develop their own basic software to use the K computer. Supercomputers developed with government funds should be designed to be easy for users to use.
  - Takahito Tokita, president of Fujitsu, said, "We have worked on the development of Fugaku with a focus on ease of use. We adopted the Arm design which is widely used as CPUs for smart phones. We also adopted Linux OS which is widely used by corporations, to increase compatibility.
  - Many programs developed companies and researchers around the world can now be used easily.
  - （Nikkei Shinbun July 2nd 2020)

This is certenly true

The CPU of K was SPARC and thus there were many differences from x86- or arm-based systems, even though the OS was Linux. Sometimes user programs could not be complied. Some of basic OSS, (such as Python) were not available initially.

One of the reasons was that Fujitsu C/C++ compiler was not compatible with gcc extensions.

On Fugake, the "trad" compiler retains the same problem, but the "clang" compiler is far better and we can even use real gcc (through spack). This is really very useful and great improvement from K.

However...

Many applications can be compiled and run on Fugaku (I do not go into problems with compiler bugs and missing capabilities of MPI etc here), but run slowly.

- The efficiency of programs highly tuned on x86 falls by a large factor (a factor of 5 is not rare) on Fugaku
- Similar or even worse degradation when moved from K
- This is true even after kernel codes are rewritten with SVE intrinsics.

What Happened ?

Comparison with Skylake Xeon

	A64fx	Skylake
FP latency	9	4
L1$ latency	8-11	4
L2$ latency	37-47	14
L3$ latency	-	50-70
OoO instruction window	128	224
Load/Store	One at time	Both

Much larger latency
half instruction window
half L1 bandwidth

↓

A64fx cannot fill its pipeline for loops which show good performance on Skylake Xeon

Example: gravity/electrostatic interaction

- K: 75% was possible with loop unrolling etc
- Skylake: 60% or around from simple code
- A64fx: around 10% with perfect SVE code
  - 30% is possible with loop fission, strip-mining etc(Nomura et al. 2020)
  - By minimizing the number of instructions in loop (manual rewrite of compiler-generated code), >50% seems possible （Nitadori 2021)
- Compared to K
  - Latency is much larger
  - Number of architecture registers is too small. As a result, neither programmers nor compiler can make good use of physical registers

Why this happened ?

- The following is an “educated guess”
- Desiginers have maximized the FP performance within the power and die-size limit
  - L1$ bandwith consumes large power
  - OoO resources as well
  - Pipeline stages are increased to use low-voltage, small transistors
- Quite understandable (at least for me as a low-power ML processor designer)
- The result for matrix multiplication is truly impressive. Not too far from (initial) numbers of NVIDIA A100
- However, performance of other kernels are ….

Did we know?

- Effective reduction of FP unit latency is possible (Odajima et al. 2018, 2020)
  - Combine two 512-bit units to form one 1024-bit units, or let both units operate on 1024-bit vectors in every two cycles.
  - Apparent latency is reduced and thus OoO operation becomes more effective
  - SMT would have similar effect too.
- $ latencies are more difficult…

How should we use Fugaku, then?

- GEMM works great.
  - Use schemes with which innermost kernels are GEMMs
    - Several particle methods do have this options
    - In principle DDM of FEM is such a scheme

  - Spherical harmonics recurrence also works great.
  - There may be other kernels which work well.
- A more general approach?
  - ?

Lessons Learned

- We cannot fulfil three goals at the same time:Performance・Range of high-efficiency operations・compatibility of ISA
  - Xeon: Range of high-efficiency operations and compatibility
  - GPU: Performance and Range
  - A64fx：Performance and compatibility

- We will be deep into the post-Moore era: Not much gain from process tech.

	1990	2000	2010	2020	2030
Design Rule	1μm	130nm	45nm	“N7”	“N2”
Vcc	5V	1.8V	1V	0.8V	0.7V?
Power reduction in 10 years	ー	60	10	6	2.5?

Summary

- Our project is in a reasonable shape
  - Many simulation codes are now working on Fugaku
  - One of them has been selected as Gordon Bell Finalist
  - Impressive scientific results have been produced and more is to come
- To achieve high efficiency on Fugaku is not easy.
  - Tradeoff between performance, compatibility and range of high-efficiency operations.
- We need to find new ways to use Fugaku efficiently, and lessens learned should be reflected to the post-Fugaku project