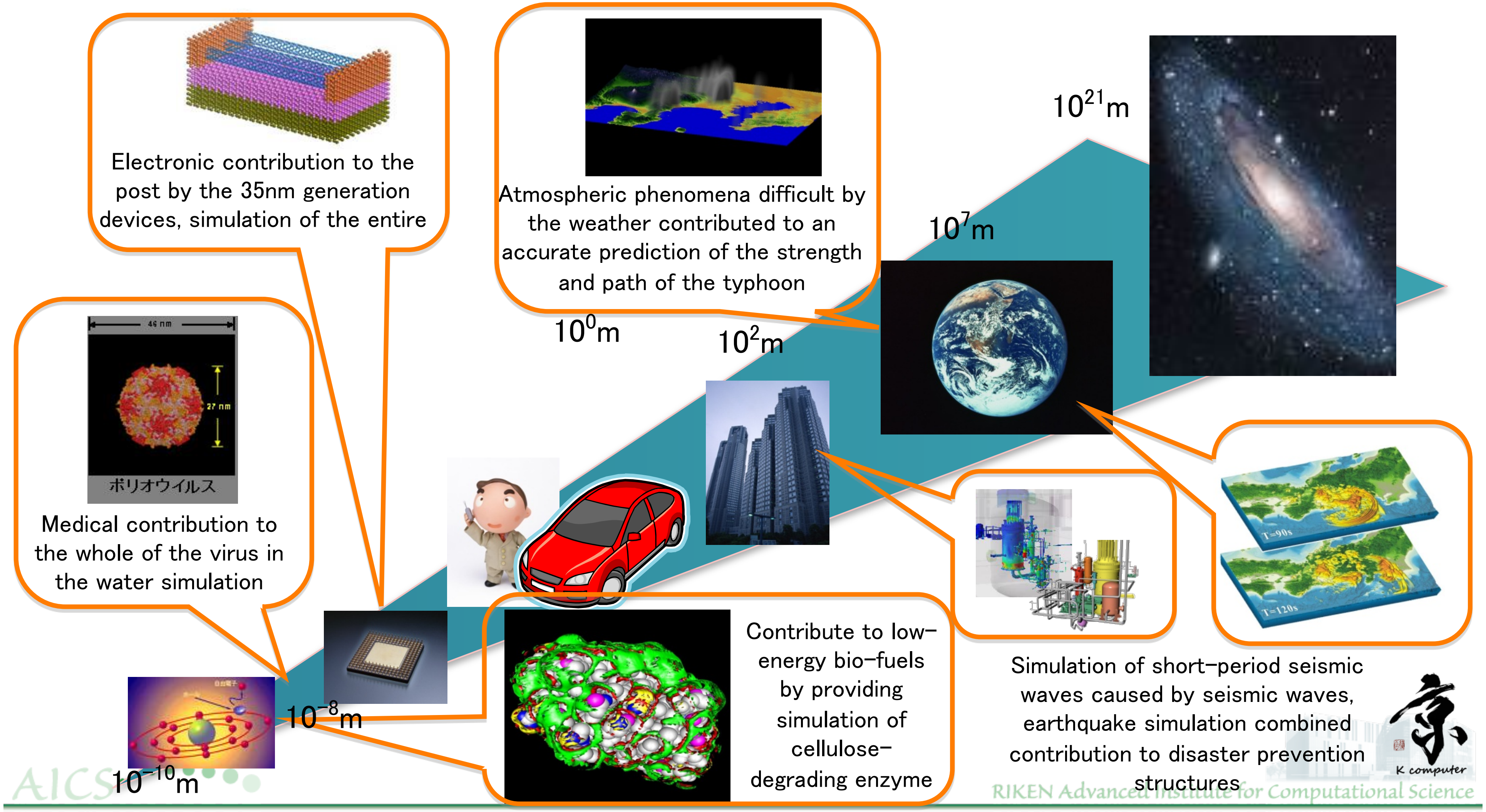
# Performance Improvement of Application on the K computer

November 13, 2011

Kazuo Minami
Team Leader, Application Development Team
Research and Development Group
Next-Generation Supercomputer R & D Center
RIKEN

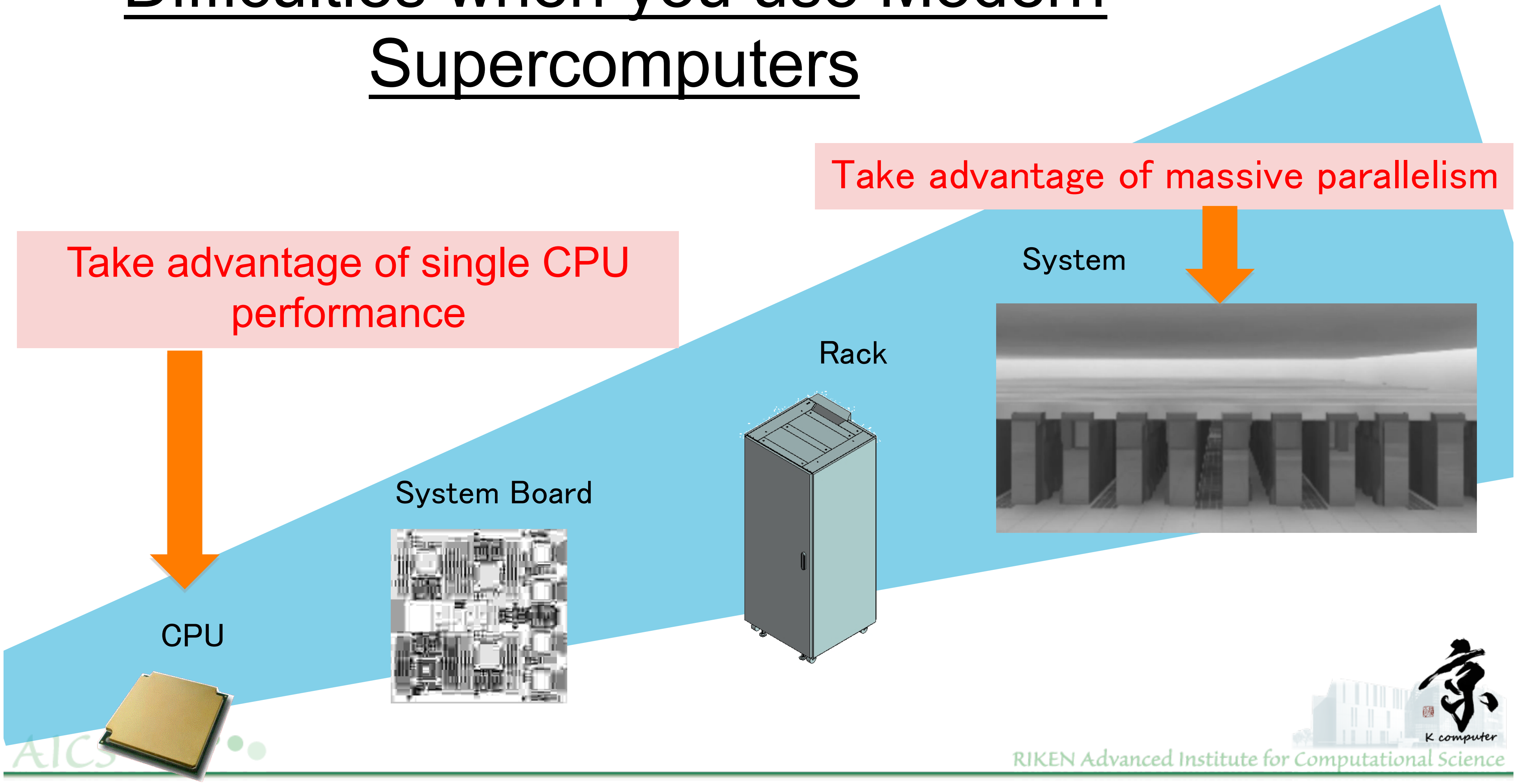


K computer

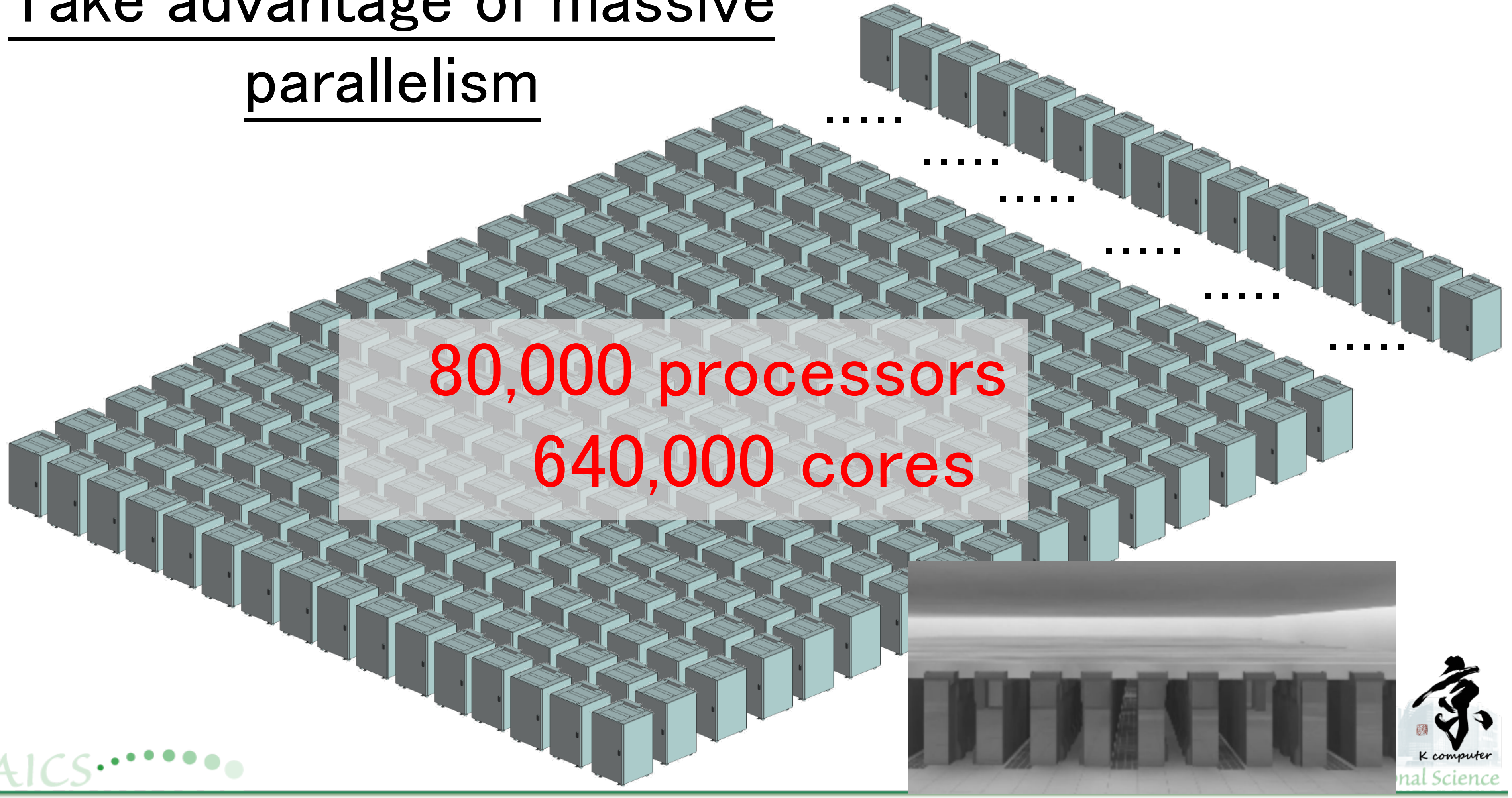


Electronic contribution to the post by the 35nm generation devices, simulation of the entire
Atmospheric phenomena difficult by the weather contributed to an accurate prediction of the strength and path of the typhoon
$10^{21}$m
$10^{7}$m
$10^{0}$m
$10^{2}$m
46 nm
27 nm
ポリオウイルス
Medical contribution to the whole of the virus in the water simulation
$10^{-8}$m
$10^{-10}$m
Contribute to low-energy bio-fuels by providing simulation of cellulose-degrading enzyme
T=90s
T=120s
Simulation of short-period seismic waves caused by seismic waves, earthquake simulation combined contribution to disaster prevention structures
京
K computer
AICS
RIKEN Advanced Institute for Computational Science

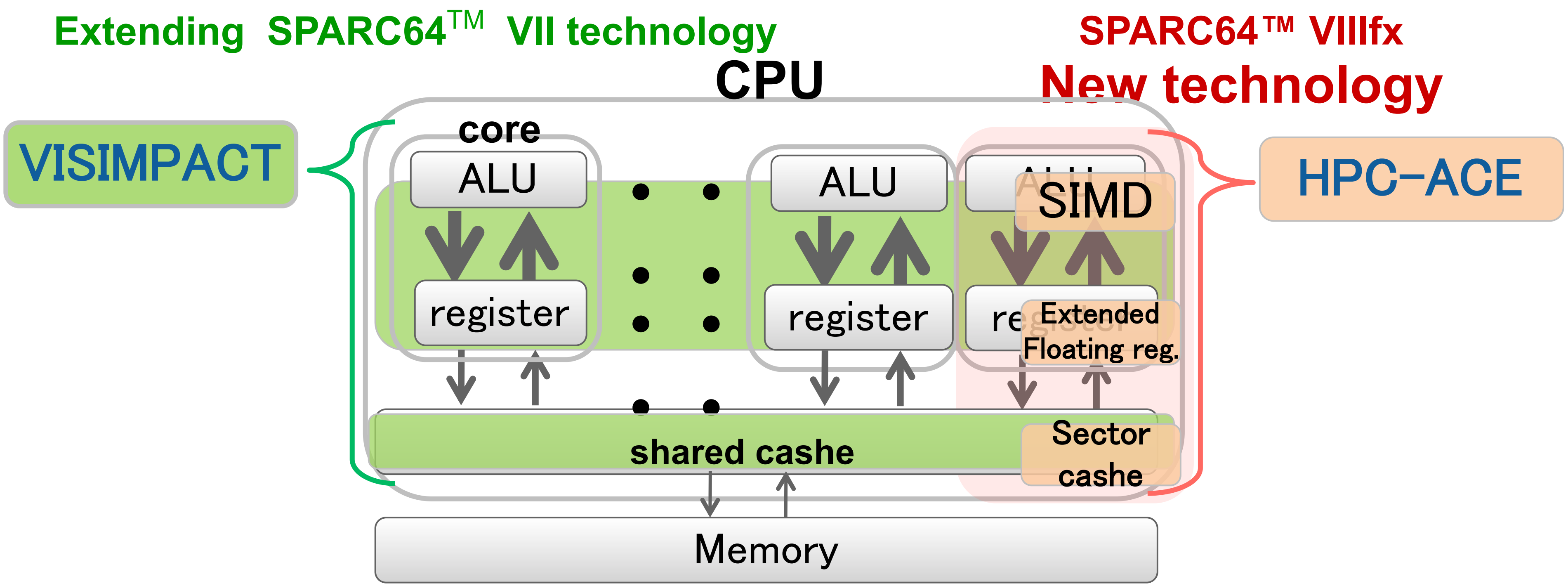
# Difficulties when you use Modern Supercomputers



# Take advantage of massive parallelism



# Take advantage of single CPU Performance




**Key point of enhancing high performance**

**→ Make the best use of new technology for compiler and applications**

# Improvement of Applications Performance at RIKEN

# Purpose

✓ To check application's performance, prior to the operation of K computer

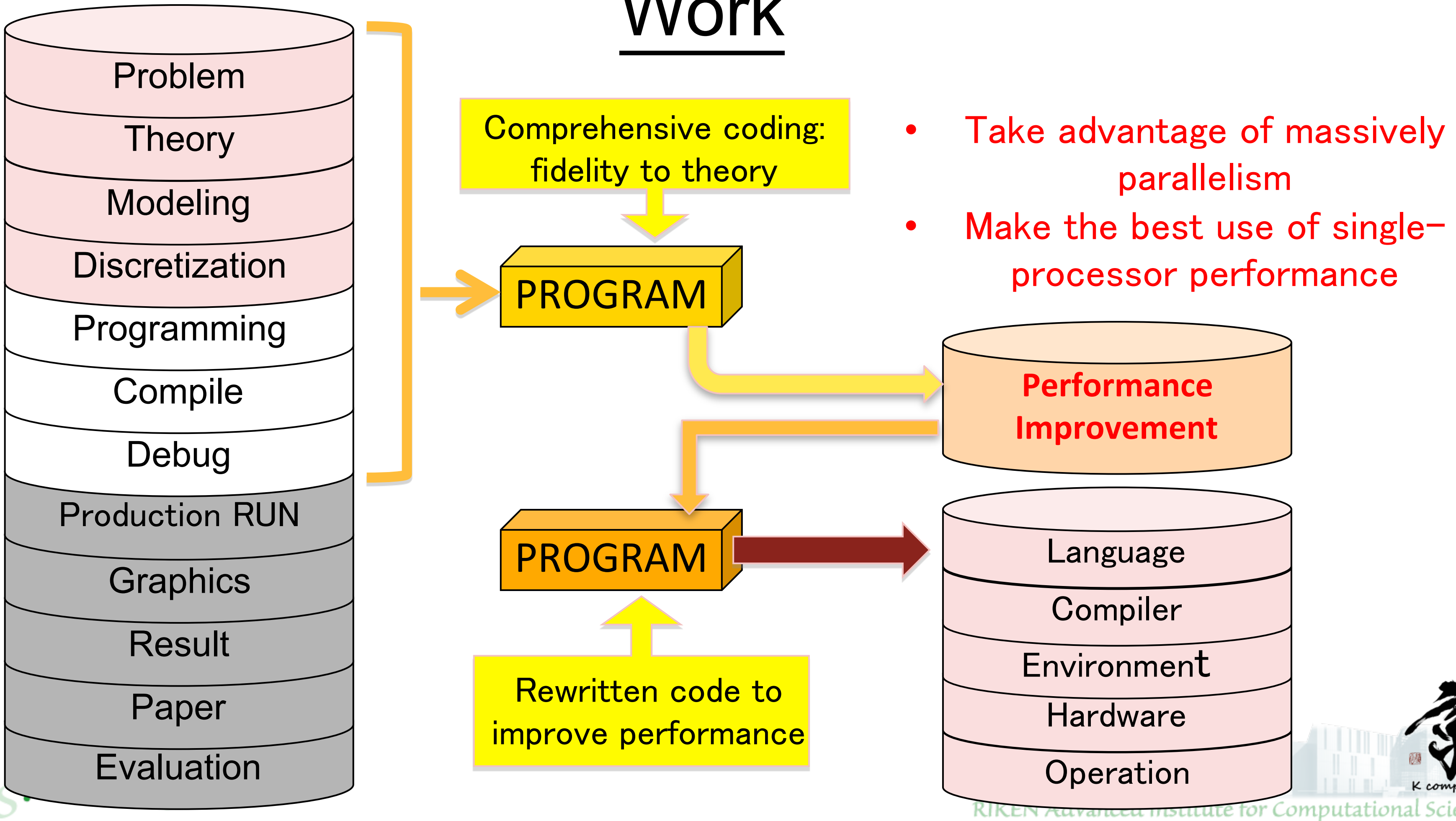
Select some applications (Nano/Engineering/Earth Science/Physics)

- By considering, application's domain and computing characteristics; *B／F or parallelism*

# Target Applications

| Program Name | Discipline | Outline | Behavior in Computational Science | Scheme |
|---|---|---|---|---|
| NICAM | earth science | Nonhydrostatic ICosahedral Atmospheric Model (NICAM) for Global-Cloud Resolving Simulations | In Earth Simulator the peak performance ratio was 25-40%, however, large value of Byte/FLOP is required. The single CPU tuning is essential by using K computer. | FDM (atmosphere) |
| Seism3D | earth science | Simulation of Seismic-Wave Propagation and Strong Ground Motions | | FDM (wave) |
| FrontFlow/Blue | engineering | Unsteady Flow Analysis based on Large Eddy Simulation (LES) | | FEM (fluid) |
| PHASE | material science | First-Principles Simulation within the Plane-Wave Pseudo potential formalism | Single processor tuning is available by applying matrix multiplication to the kernel. However, the lack of parallelism occurs in the original parallel approach. The development of parallelism is required. | DFT (plane wave) |
| RSDFT | material science | Ab-initio Calculation in Real Space | | The real-space DFT |
| LatticeQCD | physics | Study of elementary particle and nuclear physics based on Lattice QCD simulation | Single processor tuning by using K computer and parallel tuning based on Tofu-topology are necessary. | QCD |

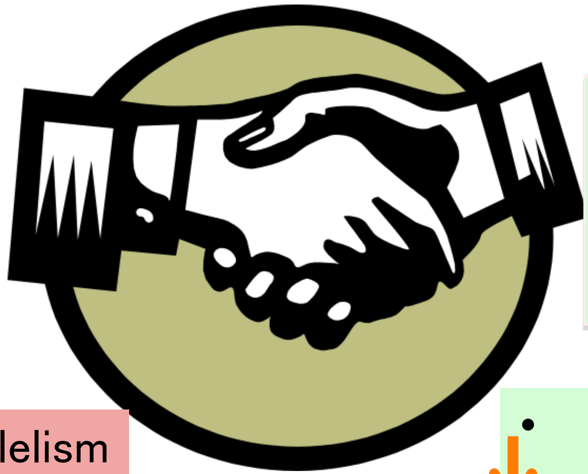
# Work



Problem
Theory
Modeling
Discretization
Programming
Compile
Debug
Production RUN
Graphics
Result
Paper
Evaluation
Comprehensive coding: fidelity to theory
PROGRAM
• Take advantage of massively parallelism
• Make the best use of single-processor performance
Performance Improvement
PROGRAM
Rewritten code to improve performance
Language
Compiler
Environment
Hardware
Operation

# Collaboration

## Application Developers



## RIKEN

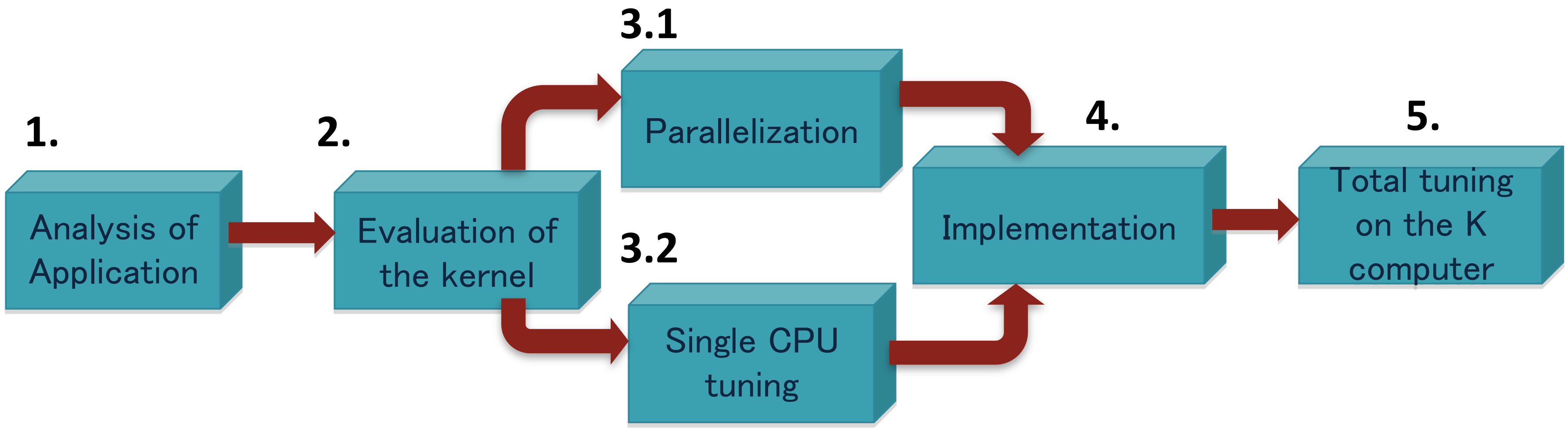
**Application Developers**

- Studying massive parallelism and high-performance
- Developing the code using test samples

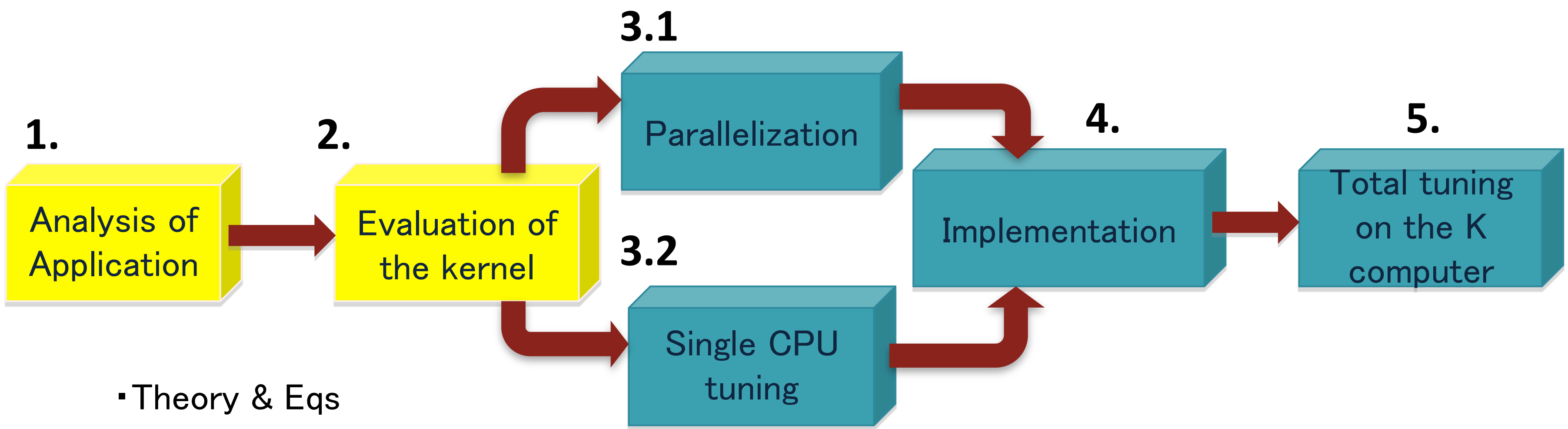
**RIKEN**

- Studying massive parallelism and high-performance
- Trial parallelization and performance tuning based on knowledge of hardware

# Procedures of Performance Improvement

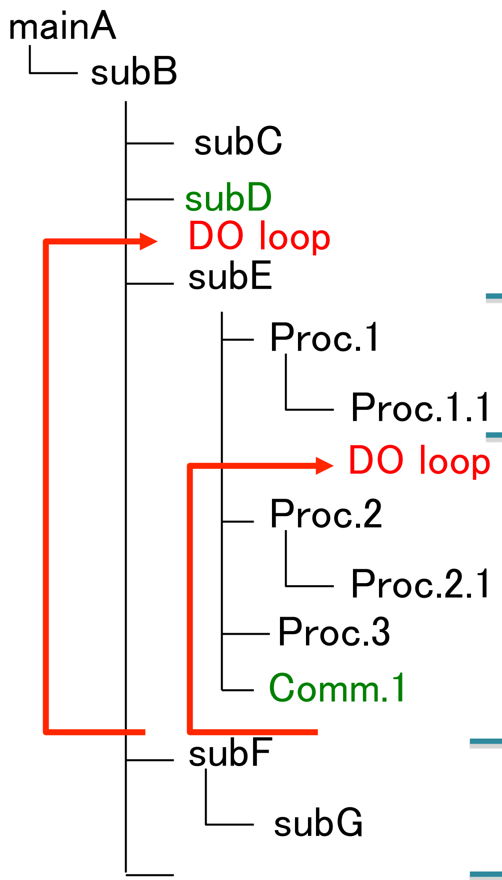


# Analysis of Application
# Evaluation of the kernel




- Theory & Eqs
- Structure of code
- Algorithms

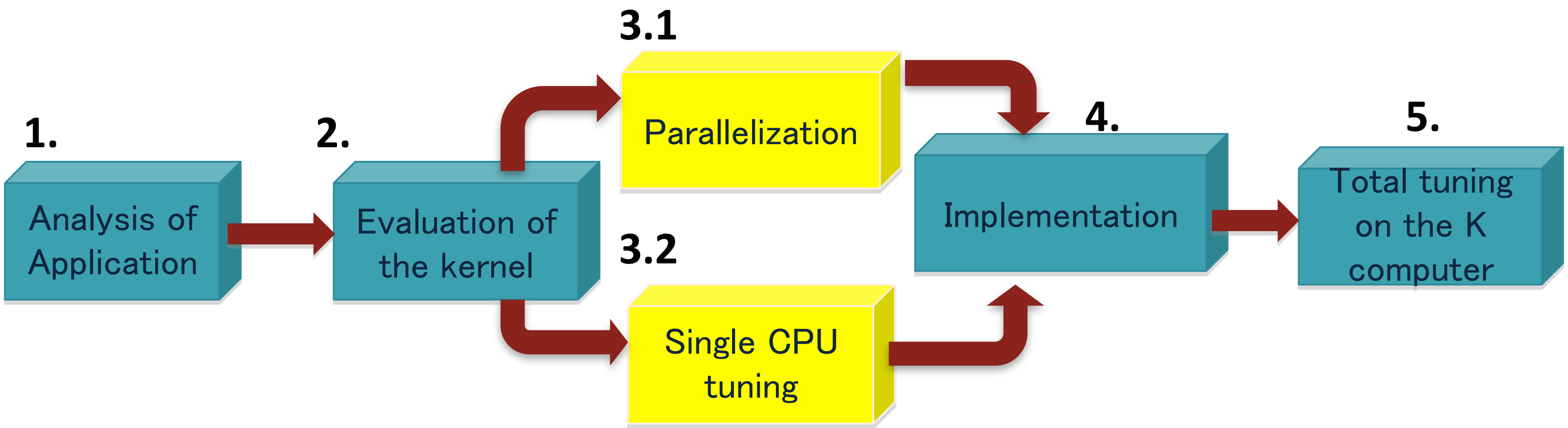
**1.** Analysis of Application




| | Calc. Time<br>Scalability | physical processes | Characte<br>－ristics | estimation | Kernel |
|---|---|---|---|---|---|
| Block 1 (Calc.) | | | Partial parallel | N proportional | |
| Block 2 (Calc.) | | | Fully parallel | $N^3$ proportional | ○ |
| (Comm.) | | | Neighbor comm. | Proportional to the adjacent surface | ○ |
| Block 3 | | | | | |

**2.** Evaluation of the kernel

# Parallelization & Single CPU Tuning



# 3.1 Parallelization

Key points for aiming at High Parallel

(1) Does non-parallel parts remain there？ If so, no problem？
(2) Is load imbalance getting worse at high parallel？
(3) How much does the neighboring comm. time occupy at high parallel？
(4) How much does the global comm. time increase at high parallel ？

*These evaluations are crucial*

Approach

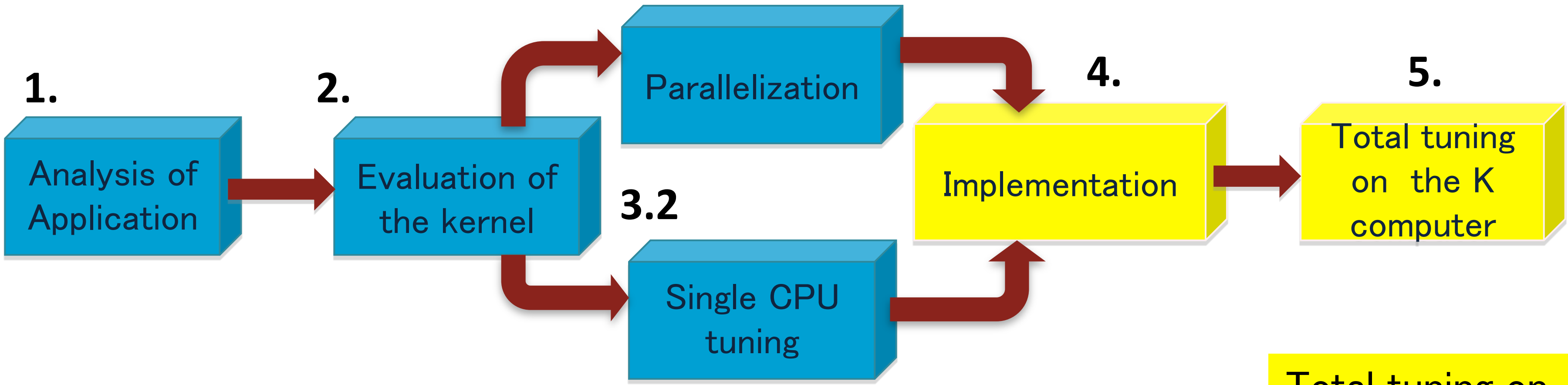
(1) Setting a target problem
(2) Making a test sample (100 parallelism)
(3) Measuring and evaluating the test sample; execution time, load imbalance, communication time between neighbors, global communication
(4) If no problem, measuring parallel performance through weak scaling
(5) If not, measuring parallel performance through strong scaling and find its cause

Strong scaling：measuring performance by increasing parallelism under the constant scale of the whole problem
Weak scaling： measuring performance by increasing parallelism under the constant scale of one CPU problem

## 3.2 Single CPU Tuning

(1) Extract kernels
→making them the independent test programs
(2) Trials for increasing performance
→ applying ideas using K computer
(3) Estimate the work volume
→ make it clear that what the impact on the whole code is and estimate the amount of work volume for introducing the performance model.
(4) Fix the model of increasing performance
→ evaluate the tests and select and fix the best.

# Mounting high-performance model and Tuning for K computer




**Implementation (Mounting performance model)**

Apply the single CPU performance model to the massive parallelized code.
Estimated to be large amount of work volume for design , programming, debugging.

**Total tuning on the K computer**

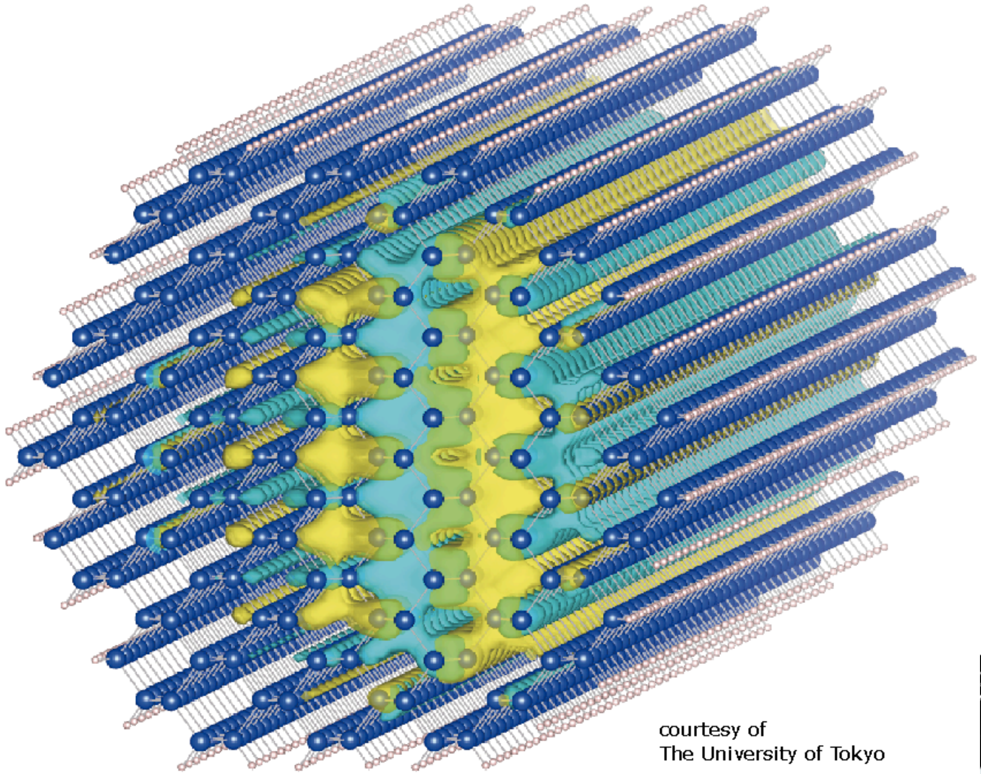
Tuning the code for K computer
**(1)Measurement**
・Parallel, single CPU performance .
**(2)Cheking and finding out**
・Performance problems
**(3)Problem solving**
・Solving the problems and apply the solution

# Example of Performance improvement

- RSDFT
- Seism3D

# RSDFT

- ✓ Real Space DFT(RSDFT) is a simulation technique most suitable for massively-parallel architectures to perform first-principles electronic-structure calculation based on density functional theory.



courtesy of
The University of Tokyo

# DFT (Density Functional Theory)

## -Solving Kohn-Sham equation-

Variational principle of the total energy with respect to the electron density leads to the Kohn–Sham equation,

$$n(\mathbf{r}) = \sum_{i:\ \text{occupied}} |\varphi_i(\mathbf{r})|^2$$

$$\left[-\frac{1}{2}\nabla^2 + v_{\text{nucl}}(\mathbf{r}) + \int \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}' + \frac{\delta E_{xc}[n]}{\delta n(\mathbf{r})}\right]\varphi_i(\mathbf{r}) = \varepsilon_i \varphi_i(\mathbf{r}) \quad .$$

# Flow of the SCF calculation

**Kohn–Sham equation**

$$\left[-\frac{1}{2}\nabla^2 + v_{\mathrm{nucl}}(\mathbf{r}) + \int \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}' + \frac{\delta E_{XC}[n]}{\delta n(\mathbf{r})}\right]\varphi_i(\mathbf{r}) = \varepsilon_i \varphi_i(\mathbf{r})$$

**Electron density**

$$n(\mathbf{r}) = \left|\sum_i \varphi_i(r)\right|^2$$

Nonlinear equation

**Self-Consistent Field procedure**

Computational complexity
(Number of atoms : N)

| | | Computational complexity |
|---|---|---|
| 1 | ( CG ) Conjugate gradient method | $O(N^2)$ |
| 2 | ( GS ) Ortho-normalization by Gram-Schmidt method | $O(N^3)$ |
| 3 | Density and potential update | $O(N)$ |
| 4 | ( SD ) Subspace diagonalization | $O(N^3)$ |

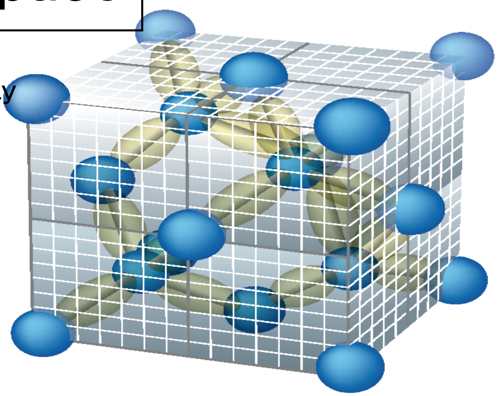
**GS and SD parts are hot spots.**

# Solving Kohn-Sham equation
## *-RSDFT( Real Space Density Functional Theory code)-*

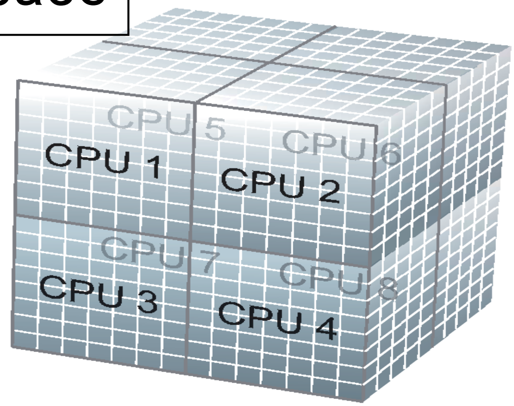
J.-I. Iwata *et al.*, J. Comp. Phys. (2010)

**Real space**

Blue : Si atom
Yellow: electron density



**CPU space**




Kohn–Sham differential equation is converted to *M*–th order finite–difference equation (usually use M=6)

$$\frac{\partial^2}{\partial x^2}\varphi_j(x,y,z) = \sum_{m=-M}^{M} c_m \, \varphi_j(x+mH,y,z)$$

Advantages

- Almost free from FFT, reducing communication burden ⇒ high efficiency
- Flexible boundary condition to wave–functions ⇒ molecules, clusters, surfaces, etc.

# Parallelization and optimization (1/3)

The RSDFT has been

- parallelized by spaces,
- optimized by rewriting matrix-matrix product with BLAS Level3 DGEMM.

10-20% efficiency to the peak performance on PACS-CS and T2K-Tsukuba, which are large scale PC clusters

# Parallelization and optimization (2/3)

What is need for an ultra-large-scale simulation on the K computer?

- parallelize by orbitals in addition to parallelization in spaces to attain sufficient parallelism

Kohn-Sham equation

$\phi_i$: *Orbitals, which are independent*

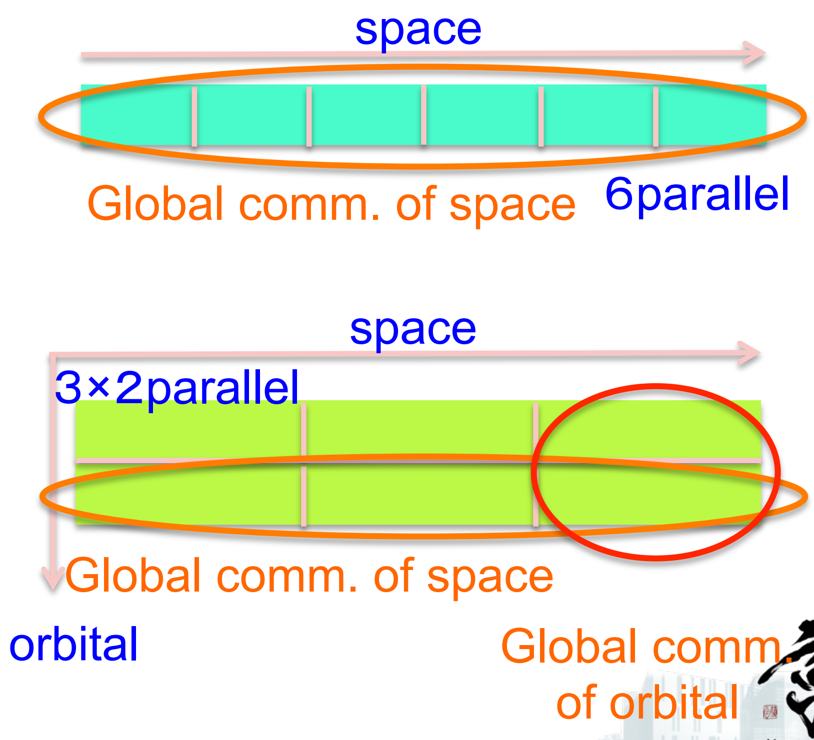
$$\left[ -\frac{1}{2}\nabla^2 + v_{\text{nucl}}(\mathbf{r}) + \int \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}' + \frac{\delta E_{\text{XC}}[n]}{\delta n(\mathbf{r})} \right] \varphi_i(\mathbf{r}) = \varepsilon_i \varphi_i(\mathbf{r})$$

- optimize load balance of parallel tasks and mapping to the Tofu network
- use highly-tuned DGEMM routine for the K computer
- employ a new eigensolver EIGEN optimized for massively parallel processors

*Imamura el al. SNA+MC2010 (2010)*

# Parallelization and optimization (3/3)

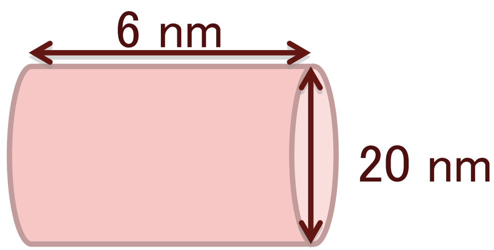
- ✓Global communication among all processors is needed when using only space parallelization.
- ✓It makes communication time increasing.
- ✓Global communication of space and orbital can be reduced by introducing 2 level parallelization of space and orbital.



# Challenge to 100,000 atoms simulation(1/2)

## Measured conditions

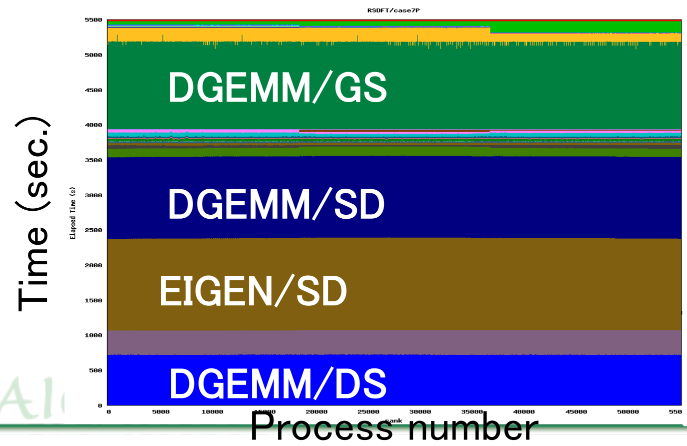
- Measured block is "ONE" iteration of SCF
- Model is SiNW with 107,292 atoms
  - ✓ Grids： 576x576x192
  - ✓ Orbitals： 229,824
- Total number of parallel processes：55,296
  - ✓ Spaces:18,432 x Orbitals: 3
- Total peak performance : 7.07PFLOPS
  - ✓ 55,296 nodes(442,368 cores)



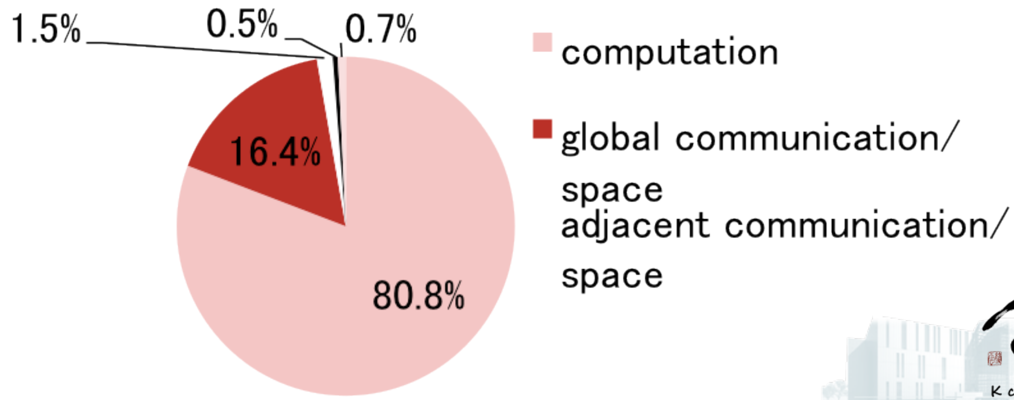
# Challenge to 100,000 atoms simulation(2/2)

- Sustained performance is **3.08 PFLOPS** /SCF.
- **43.6 %** efficiency to the peak performance.
- Communication cost is 19.0% of all execution times.
- One iteration time of SCF is 5,500 sec. (1.5 hours)

**Load balance among processes**

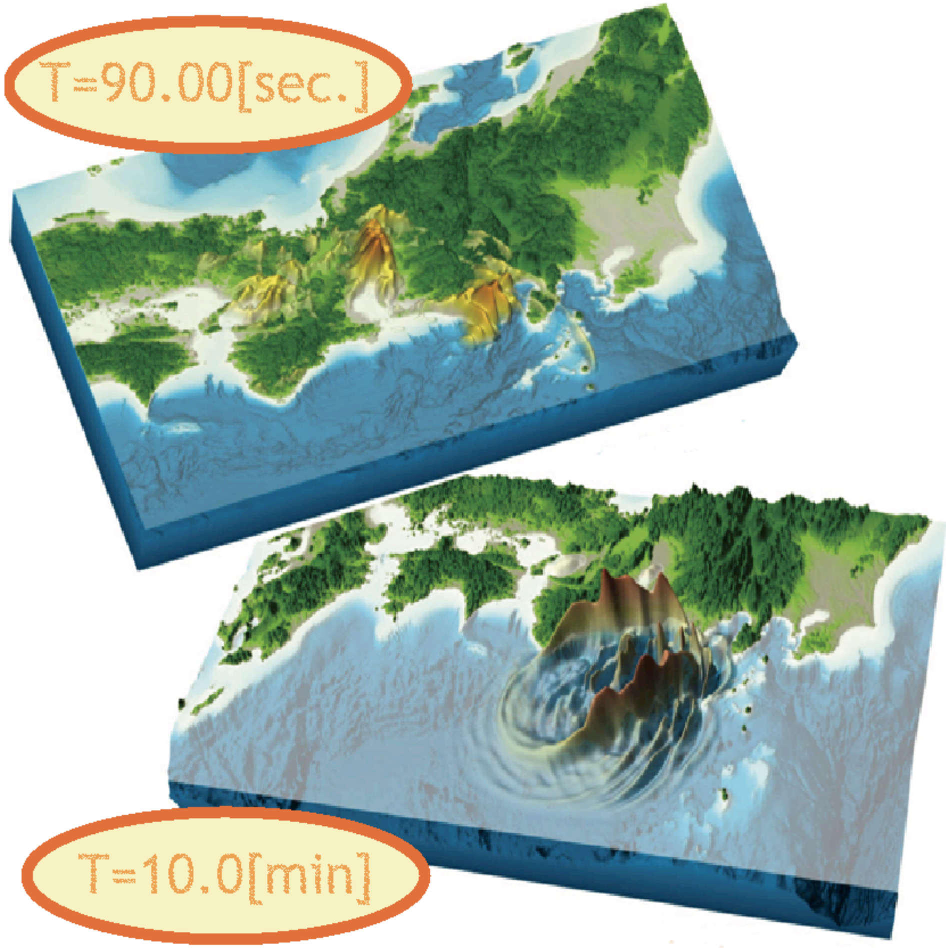



**Distribution of computational cost**



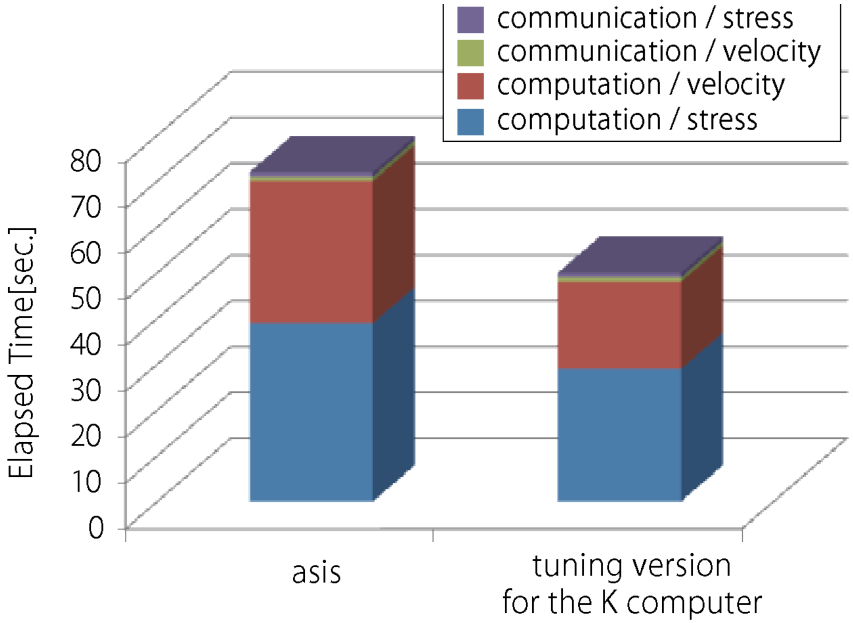
# Seism3D

- The Seism3D numerically solves the equation of motion of viscoelastic medium by massively parallelized finite difference method for the simulation of seismic wave propagation from the large earthquake in the heterogeneous earth.



# Performance Improvement Result (Tentative)

- Using array fusion, loop division and loop unrolling techniques in order to improve performance in computation parts.
- As a result, 45% speed.
- In communication part, we are studying an optimal rank mapping on the Tofu interconnect.



# Summary

## Improvement of Applications Performance at RIKEN

- We are developing the K computer and have to demonstrate its performance by using real applications.
- We have improved performance for selected 6 application in terms of both single CPU and parallel performance since 2009.

## Example of Performance improvement

- RSDFT：We obtained a sustained performance of 3.08 peta-flops, or 43.6% efficiency to the peak performance.
- Seism3D：45% speed up