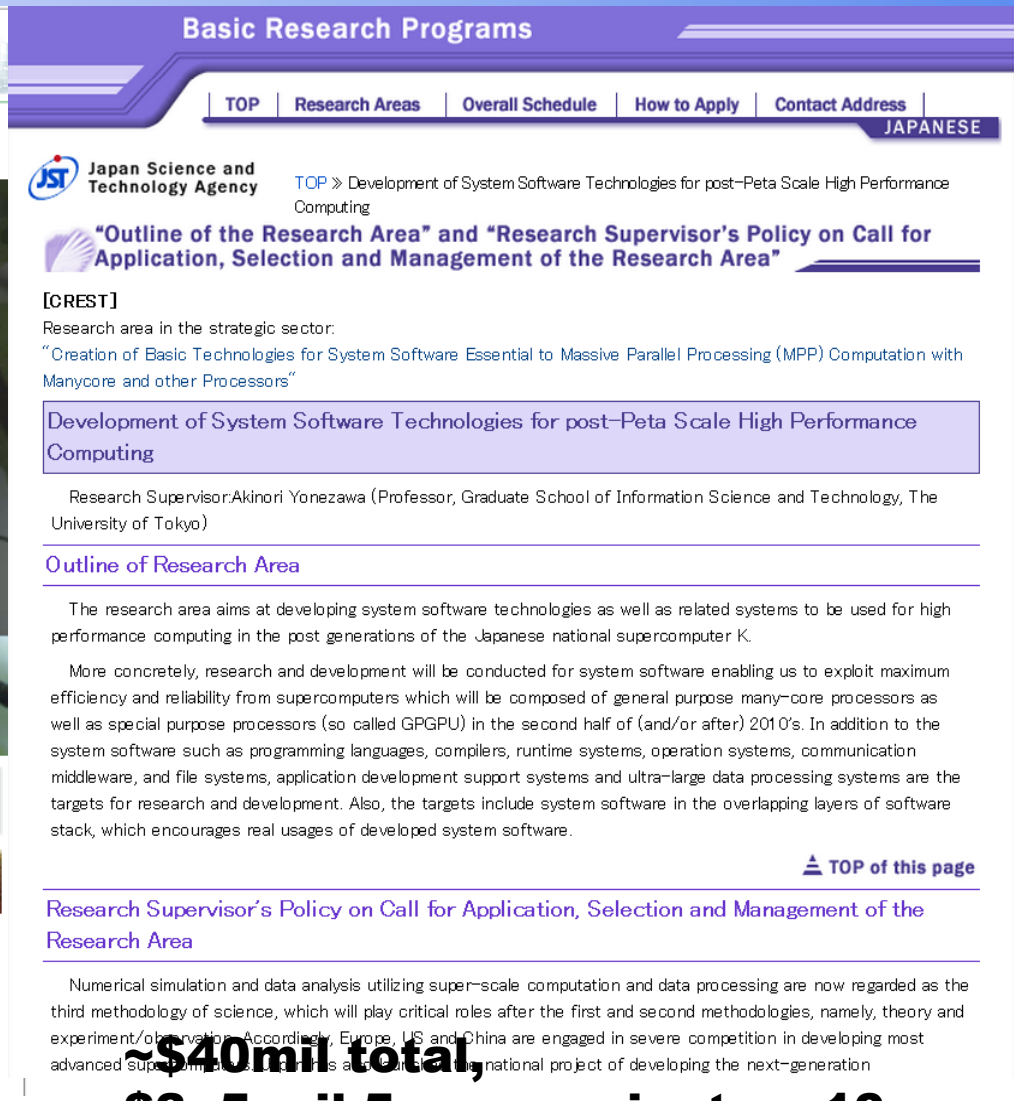
Expected funding and plans beyond NGS

- **JST CREST (Core Research for Evolutional Science and Technology) for "Post-Peta HPC technologies"**
  - **http://www.senryaku.jst.go.jp/teian/en/koubo/04-ch2202.html**
  - **500-600M JPY(5M USD) for 5-7 years (4-5 B JPY in total) ~$40 million total**
  - **System software-oriented, NOT hardware**
  - **Assume post-petascale architecture**
  - **Real Software deliverable pieces required, not just papers**
- **Research for the next of NGS (exascale?) will also be conducted in the AICS (Kobe Center).**
  - **2 teams currently by Ishikawa and Sato**
- **In the 4th Science and Technology Basic Plan (FY2011-FY2015)**
  - **Now under discussion toward exaflops class HPC technology**
  - **Recent turn of events may change priority (green, safety)**

Post Petascale Projects & Organizations in Japan



~$40mil total,
$3~5 mil 5 year projects x 10

Petascale Machines in Japan will be arriving fast circa 2012

- TSUBAME2.0 (2010Q4, 2.4PF Tokyo Inst. Tech.)
- HA-PACS (2011Q4, ~1PF, Univ. Tsukuba)
- Univ. Tokyo (2012Q1 ~1PF, non-cluster)
- Kyoto-U (2012Q2, 0.7PF?)
- KEK (2012, 1.2PF, BG/Q)

- "Kei" (2012Q1 >10PF, Kobe National Facility)
- Most facilities power constrained (worse with recent post-disaster power outages)
- 2014-15 >20 PF "Post-Peta" machines
  - "Hybrid" Many-Core/Multi-Core Architecture

JST-CREST Post-Petascale First Round (Started Apr. 1, 2011)

- Parallel System Software for Multi-Core and Many-Core (OS)
- System Software for Post Petascale Data Intensive Science (Data, Filesystem)
- Highly Productive, High Performance Application Frameworks for Post Petascale Computing (Programming, Frameworks)
- ppOpen-HPC (Numerical Library)
- Development of an Eigen-Supercomputing Engine using a Post-Petascale Hierarchical Model (Numerical Library)

2nd round call due May 17th, to start Oct. 1, 2011

Parallel System Software for Multi-Core and Many-Core

Team Leader:

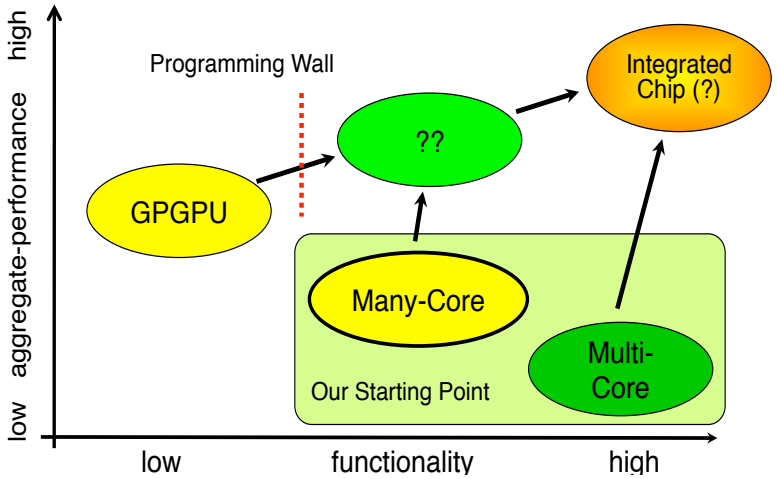
Atsushi HORI, RIKEN/AICS

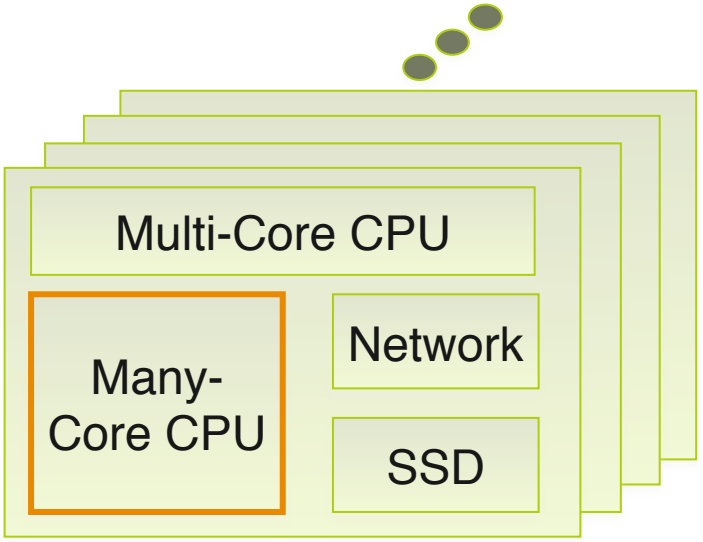
Background and Motivation

Assuming the H/W architecture combining multi-core and many-core CPUs will be the key technology for the exa-scale computing, the OS development for such architecture is important.

Although the architecture combining multi-core and many-core is an infant technology, we shall start R&D for such OS, since the software development often takes several years.

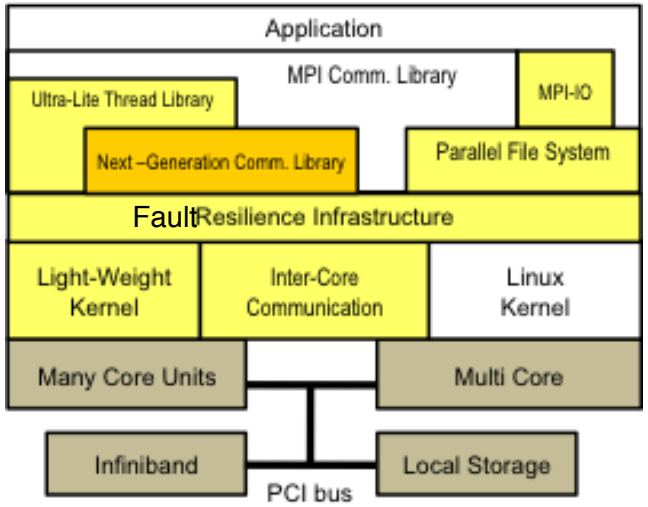
The OS R&D will be started with the cluster consisting of the compute nodes having many-core accelerator.

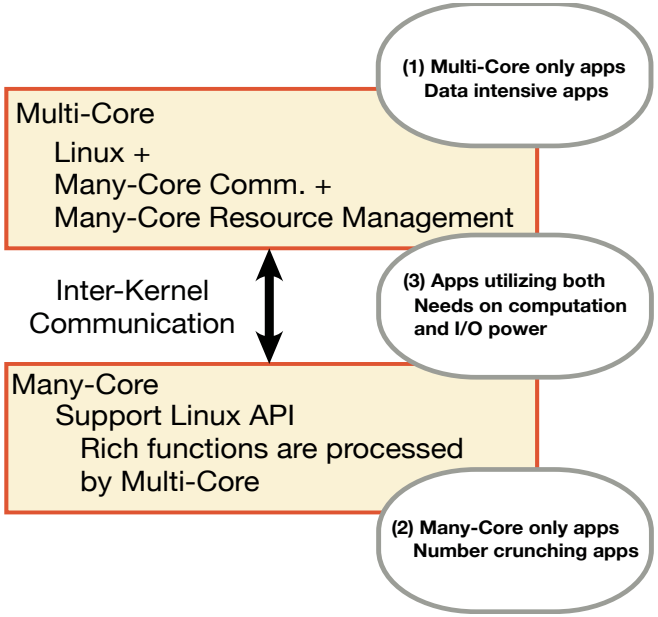




Parallel System Software for Multi-Core and Many-Core

- Research Topics
  - Many-Core OS Kernel
  - Communication and I/O
  - Light-Weight Thread Lib.
  - Fault Resilience Infrastructure
- Outcome
  - Integrated Software Package
    Free and open source
- Research Groups
  - RIKEN/AICS
  - Tokyo Univ. of Agriculture and Technology
  - Kinki Univ.
  - ORNL

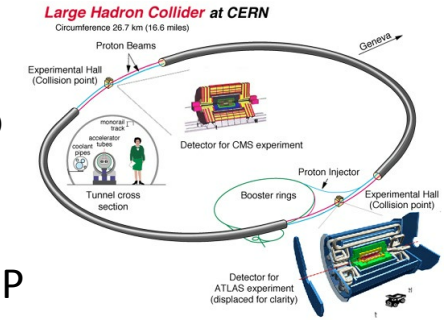




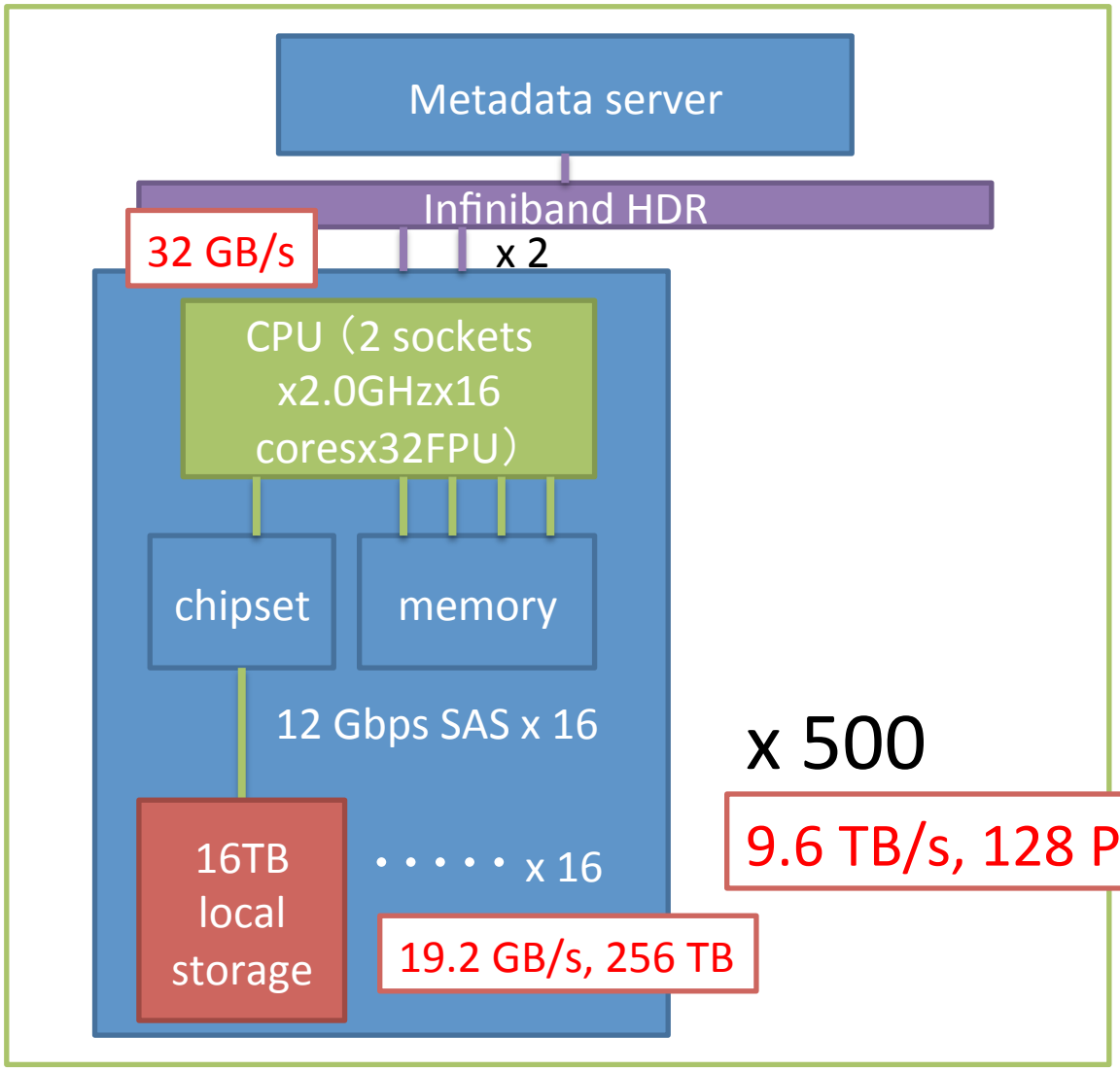
System Software for Post Petascale Data Intensive Science

Co-PI	Institute	
Osamu Tatebe	University of Tsukuba	Project Leader
Yoshihiro Oyama	University of Electro-Communications	

- Objective
  - Develop scale-out file system architecture and software
  - Target snapshot, 1 Exabyte, 100 TB/s, five years later
- Research topics
  - Distributed file system
    - I/O performance scale-out to tens of thousands I/O servers by utilizing access locality
    - Metadata server clustering to scale the metadata performance out
  - Compute node OS
    - File system kernel driver, client caching, operation offload to surplus cores
  - Runtime for Data-Intensive Computing
    - Efficient runtime of workflow execution, MapReduce, and MP IO for the scale-out distributed file system



Target Scale-out Storage Architecture



- 5 years later snapshot
- Non-uniform but scale-out storage
- Develop system software stacks to achieve maximum I/O performance for data-intensive science

CREST
Core Research for Evolutional Science and Technology

Highly Productive, High Performance Application Frameworks for Post Petascale Computing

Naoya Maruyama, Tokyo Tech (PI)
Takayuki Aoki, Tokyo Tech (Co PI)
Kenjiro Taura, U-Tokyo (Co PI)
Kenji Yasuoka, Keio (Co PI)

Project Overview

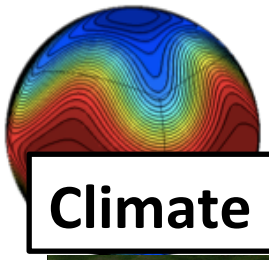
Emergence of large-scale heterogeneous supercomputers → Low productivity due to lack of programming model

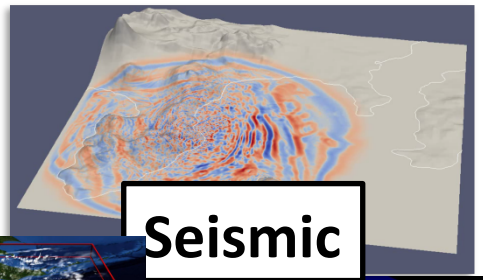
Domain-specific application frameworks for post peta-scale systems that address:

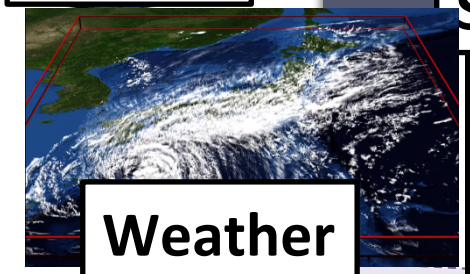
1. Productivity: Auto-parallelization and global memory
2. Performance: Comparable to manually tuned code
3. Resilience: Transparent and ultra-fast checkpointing
4. Power efficiency: Hardware-software cooperative power management

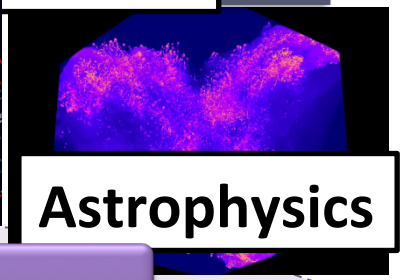
→ Demonstration using TSUBAME3 by developing frameworks for CFD and MD

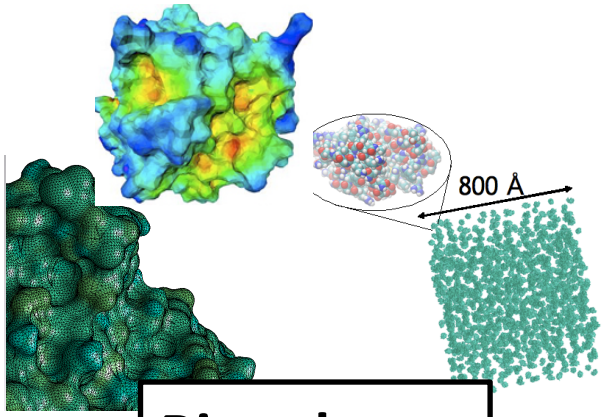
Framework Overview


Climate


Seismic


Weather


Astrophysics



Biopolymer

CFD DSL

MD Library

CHARMM

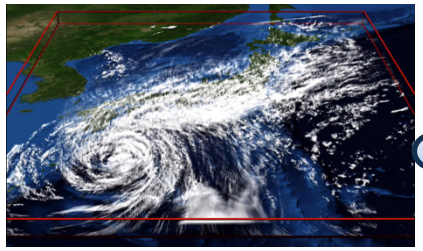
Load Balancing | Auto Parallelization | Auto Tuning | Fault Tolerance

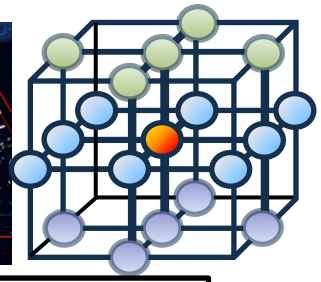
Common Runtime

Partitioned Global Address Space | Scalable Multi-threading | Fine Grained Profiling of Perf. & Power | Fast Checkpointing

100-PF Heterogeneous Supercomputer

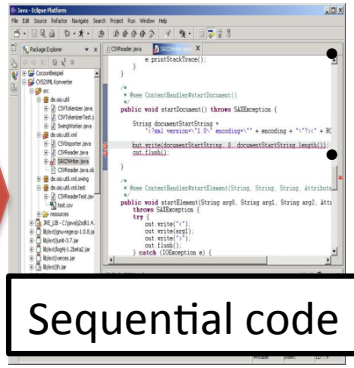
Domain-Specific Programming Model for Heterogeneous Supercomputer





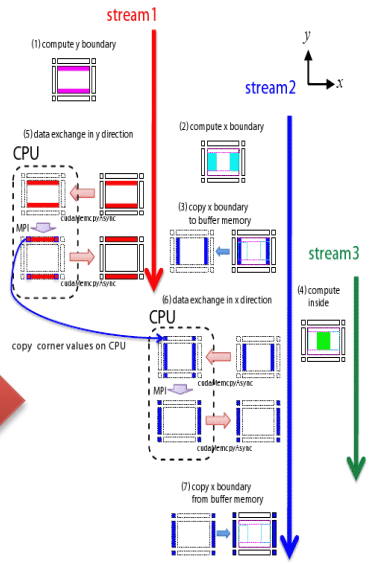
Modeling & discretization

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P - 2\Omega \times \mathbf{u} - \Omega \times (\Omega \times \mathbf{r}) + \mathbf{g} + \mathbf{F}$$



Sequential code

- CPU/GPU hybrid programming
- Memory hierarchy optimization
- Scalability optimization by overlapping communication and computation
- ....



Domain-Specific Language

- Declarative
- Portable
- Shared memory model
- Sequential execution model

Backend

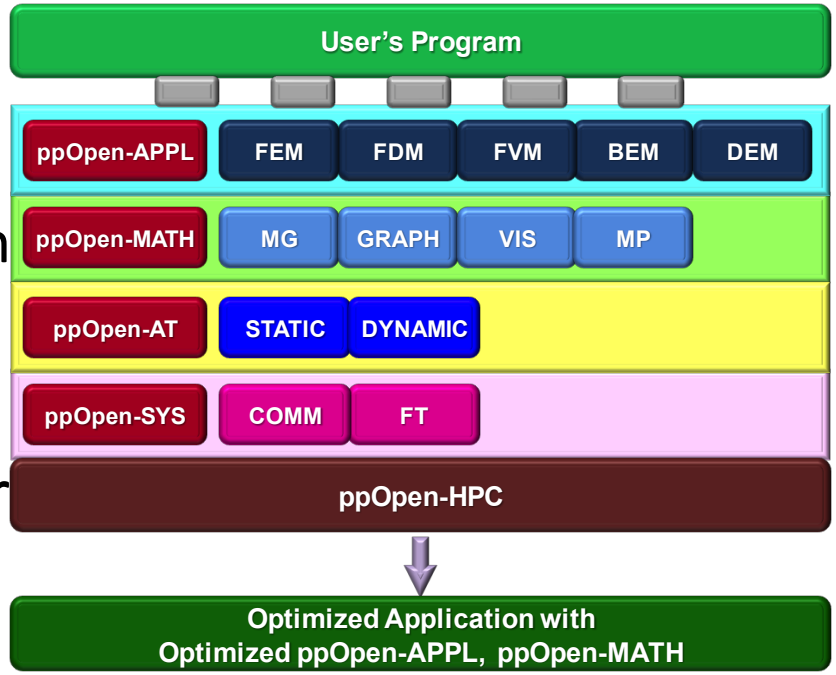
- Automatic parallelization
- Model-based performance and power optimization
- Auto-tuning-based optimization
- Hybrid code

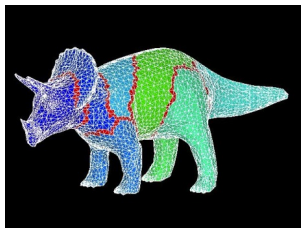
Runtime

- Fault tolerance (checkpointing)
- Model refinement with runtime information
- Further auto tuning

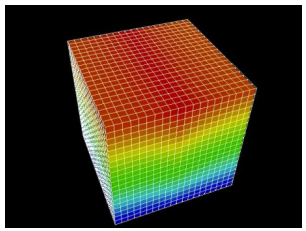
ppOpen-HPC (1/2)

- Open Source Infrastructure
  - for development & execution of optimized & reliable codes
  - on post-peta (pp) scale system with heterogeneous computing nodes
    - Multicore CPU's + Accelerators (e.g. GPGPU and/or Manycores etc.)
- Groups of Libraries, Tools etc. for various types of procedures in scientific computations.
  - ppOpen-APPL
    - FEM, FDM, FVM, BEM, DEM
    - Linear Solvers, Matrix Assembling,
    - I/O, AMR/DLB
  - ppOpen-MATH
    - MG, Graph op's, Visualization, Coupling
  - ppOpen-AT (Autotuning)
    - Static, Dynamic
  - ppOpen-SYS
    - Node-to-node comm., Fault Tolerance

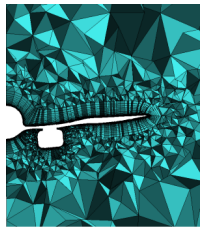




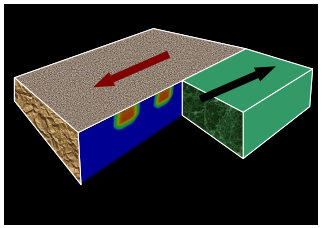
FEM
Finite Element Method



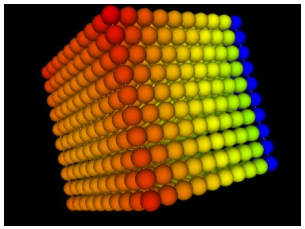
FDM
Finite Difference Method



FVM
Finite Volume Method



BEM
Boundary Element Method



DEM
Discrete Element Method

ppOpen-HPC (2/2)

- Features/Goals of ppOpen-HPC
  - Source code developed on a PC with a single processor by FORTRAN/C is linked with these libraries, and generated parallel code is optimized for post-peta scale system.
    - CUDA, OpenGL etc. are hidden from application developers
  - Automatic tuning (AT) enables smooth and easy shift to further development on new/future architectures through ppOpen-AT
    - Directive-based special AT language (e.g. ABCLibscript) for specific procedures in scientific computing, focused on optimum memory access
  - Co-Design by Computer/Computational Sciences, Numerical Libraries/Algorithms (P.I.: Kengo Nakajima (ITC/Univ. Tokyo))
    - 4 institutes of Univ. Tokyo (ITC, AORI, CIDIR, RACE), Kyoto U. & JAMSTEC
- Related Works
  - Component –based frameworks
  - GeoFEM, HPC-MW, Sphere, OpenMM
- International Contributions
  - HMC (Hybrid Multicore Consortium)
  - IESP

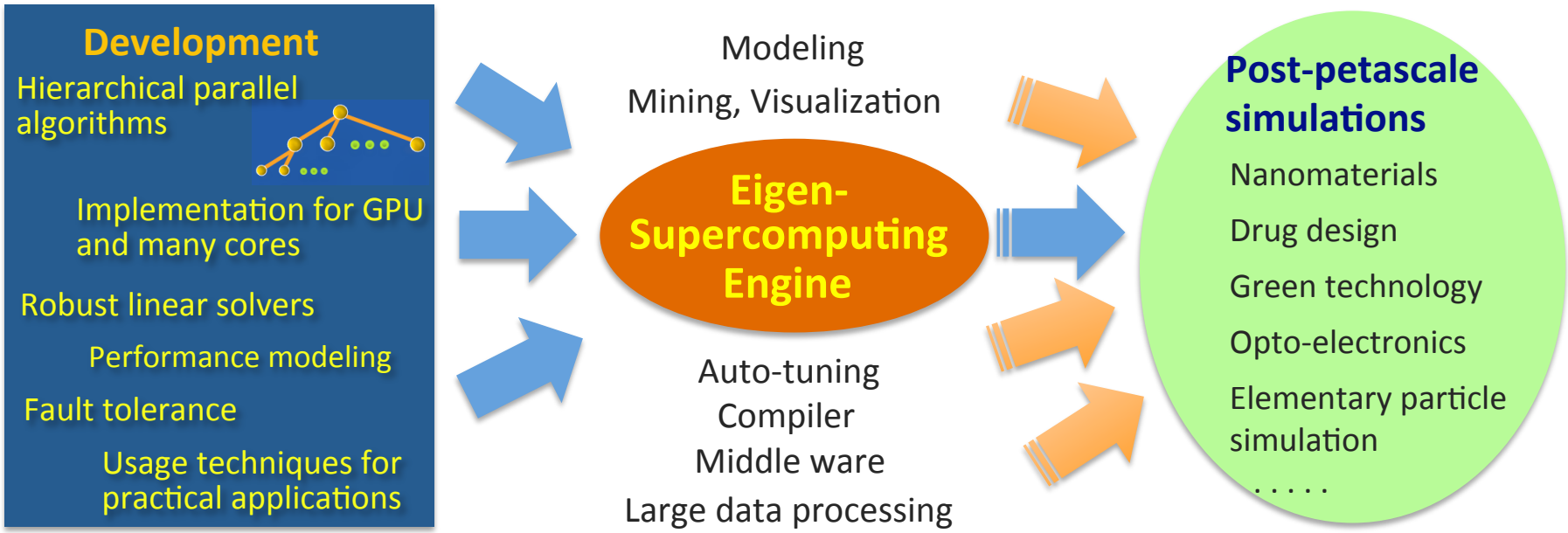
```
#pragma ABCLib install unroll (i,j,k) region start
#pragma ABCLib name MyMatMul
#pragma ABCLib varied (i,j,k) from 1 to 4
   for(i = 0 ; i < n ; i++){
     for(j = 0 ; j < n ; j++){
       for(k = 0 ; k < n ; k++){
            A[i][j] = A[i][j] + B[i][k] * C[k][j];
     } } }
#pragma ABCLib install unroll (i,j,k) region end
```

Development of an Eigen-Supercomputing Engine using a Post-Petascale Hierarchical Model

Project Leader: Tetsuya Sakurai (University of Tsukuba)

Core Member: Toshiyuki Imamura (University of Electro-Communications), Zhang Shao-liang (Nagoya University), Yusaku Yamamoto (Kobe University), Yoshinobu Kuramashi (University of Tsukuba), Takeo Hoshi (Tottori University)

The aim of this research is to develop a massively parallel eigenvalue analysis engine taking advantage of a hierarchical computer structure which is a defining characteristic of a post-petascale machine.

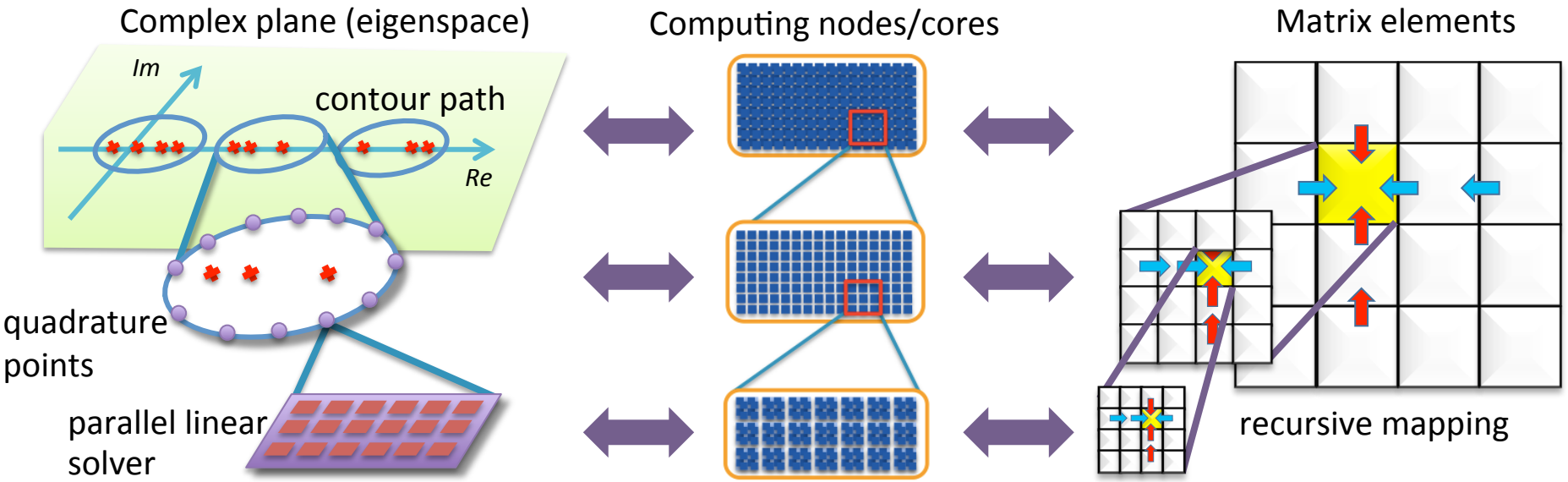


Hierarchical Parallel Algorithms for Eigenvalue

- Design hierarchical parallel algorithms which fully leverage performance of post-petascale architectures.
- Develop software with high performance/scalability/portability/reliability.
- Construct performance models to predict performance on post-petascale machines.
- In collaboration with researchers in fundamental sciences and nano-material science.

Sparse matrix: Localization of fine-grained communication by using filtering of eigenspaces based on contour integration

Dense matrix: Recursive formulation of the tridiagonalization algorithm based on matrix-matrix multiplication



Cluster computing nodes and cores according to hierarchical structures of the

Other Peta/Post Peta Software Projects

- Univ. Tokyo & U-Tsukuba/AICS: e-Science (Ishikawa, Sato…)
  - XScalableMP, etc.
- Tokyo Inst. Technology (+NII): JST-CREST Ultra Low Power HPC + Green Supercomputing (Matsuoka)
- French-Japan JST-ANR (U-Tokyo/Kyoto-U/U-Tsukuba/Tokyo Tech), G8, …