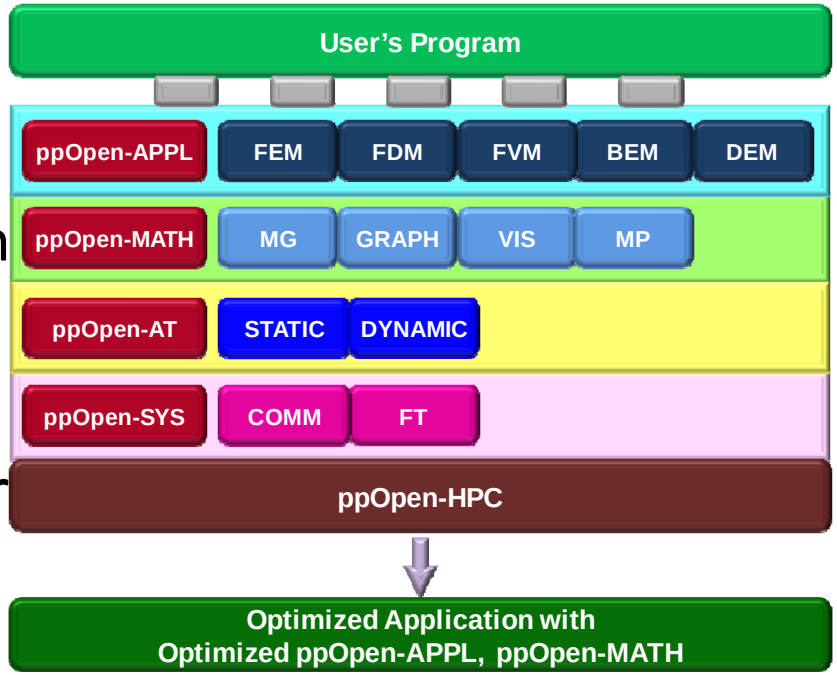
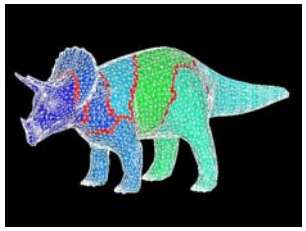
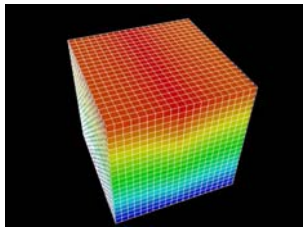
ppOpen-HPC (1/2)

- 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
    - Static, Dynamic
  - ppOpen-SYS
    - Node-to-node comm., Fault Tolerance



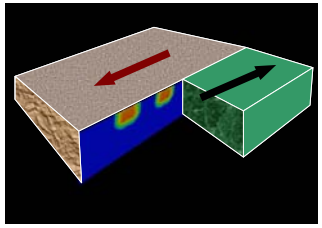


FEM
Finite Element Method

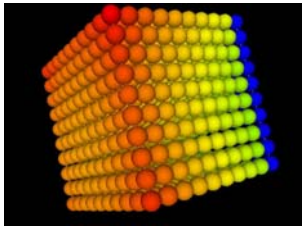


FDM
Finite Difference Method

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
```