AICS

RIKEN Advanced Institute for Computational Science

DESIGN OF PRACTICAL FRAMEWORK TO UTILIZE BIG DATA ON EXASCALE COMPUTER

Kenji ONO

Advanced Visualization Research Team

Advanced Institute for Computational Science, RIKEN

/ Kobe University / University of Tokyo

ORAP Forum 2014-04-10

TOC

- Exascale computer project in Japan
- Obtaining knowledge from large-scale dataset
  - Data management
  - Workflow
  - Visualization on supercomputer

EXASCALE COMPUTING

- Exascale computing = (FLOPS && power && data)
  - Power efficiency & data manipulation are stronger limitation
- Alternatives
  - Manycores >> Latency core
  - Embedded core >> BG
  - Accelerator
- Scientific results become more important >> applications
  - Science roadmap

FEASIBILITY STUDY FOR HIGH PERFORMANCE COMPUTING INFRASTRUCTURE

- During 2012-2013
  1. Univ. of Tokyo — Latency core
  2. Univ. of Tsukuba — Accelerator
  3. Tohoku Univ. — Vector
  4. RIKEN — Applications, roadmap

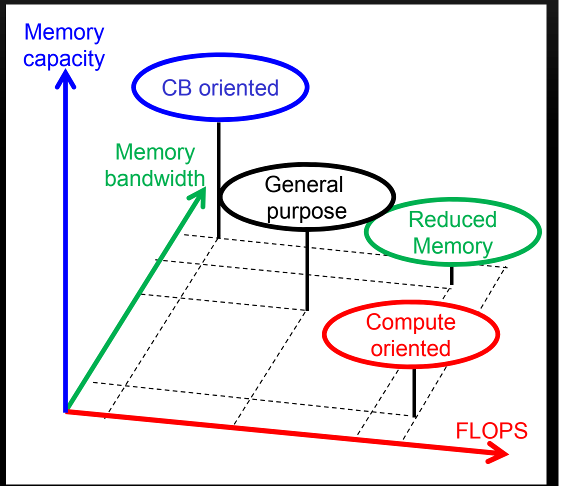
LATENCY CORE BASED ARCHITECTURE

- U Tokyo
  - Based on K computer architecture
  - Improve power efficiency per FLOPS
    - Low-voltage, enhanced pipeline, large-cache, high-clock,...
  - Target applications for benchmark
    - ALPS, RSDFT, NICAM, COCO, NTchem,...
    - Apps are taken from the science roadmap
  - Capability computing
  - Co-design
    - Applications, System software, Architecture

ACCELERATOR BASED ARCHITECTURE

- Strong scaling & power efficiency
  - MD, Lattice QCD, Stencil applications

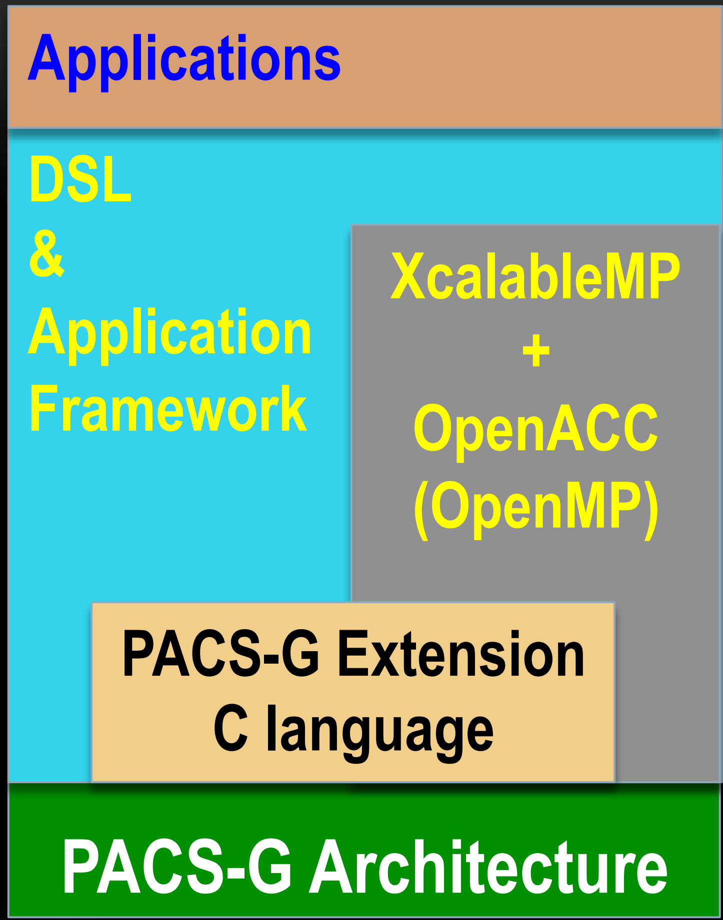
- Architecture centered
  - Master – Latency core + global memory
  - PE – Accelerator + Local memory + high-speed interconnect
  - Extreme SIMD operation



- Way to use
  1. Off road model (Host + Accelerator)
  2. Accelerator only model
  3. Cooperation model

PROGRAMING MODEL

- PGAS-G C extension language
- XcalableMP + OpenACC
- DSL
- Directive to specify off-roading
- MPI



FEASIBILITY STUDY OF APPLICATIONS

- Extraction of social / scientific challenges for 5-10 years later
  - Join more than 100 researchers, 35 organizations (Univ., Institute, Gov., Company)
    - Bio
    - Nano
    - Earth science, disaster
    - Advanced manufacturing
    - Fundamental science
    - Social / economical science
- Mini-application
  - More realistic performance evaluation

WHAT ARE THE HURDLES TO BE SURMOUNTED?

- Energy and Power
- Concurrency
- Reliability (Resiliency)
- Programming

2ND PART OF MY TALK

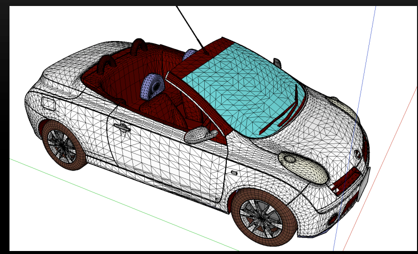
- Extreme computing for manufacturing process
- Example >> Automotive CFD

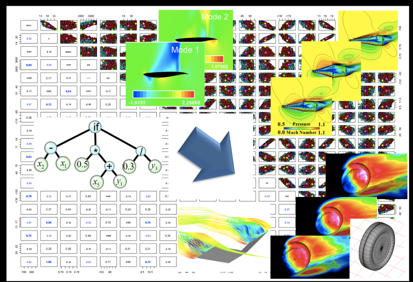
- Three scenarios to exploit an exascale computing environment
  1. Express simulation
  2. Grid search / optimization
  3. Utilization of database

IMPACT OF EXTREME-COMPUTING FOR PRODUCT DESIGN

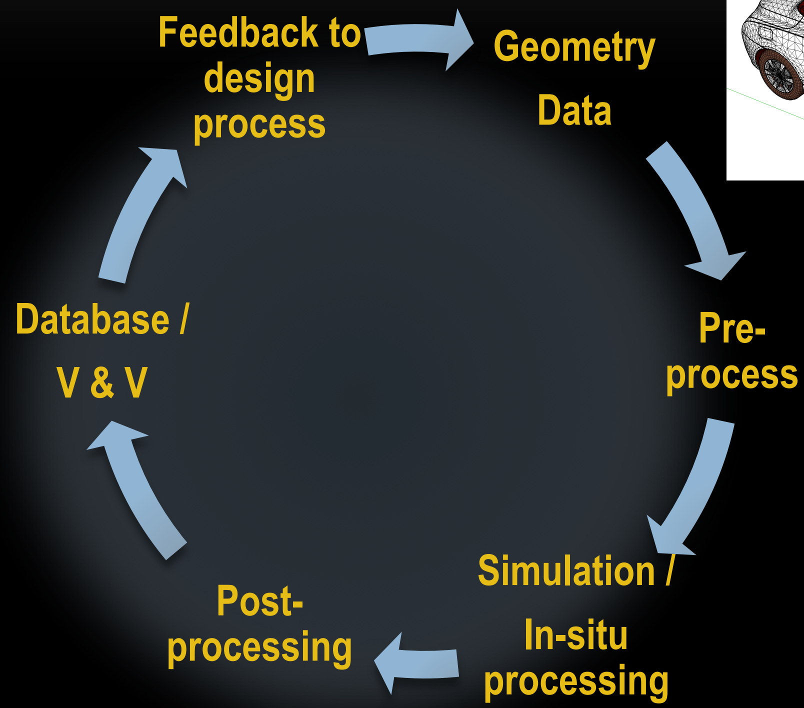
- HPC will change a style of product design
  - Reduce time cost
    - A solution in a short period of time
    - Many trials in shot turnaround time
      - Parametric study with details becomes feasible > MOO
  - Increase reliability
    - Reliability of the results becomes higher as the resolution increases with adequate solution method, e.g., LES in CFD.
  - Tackle complicated phenomena
    - More physics

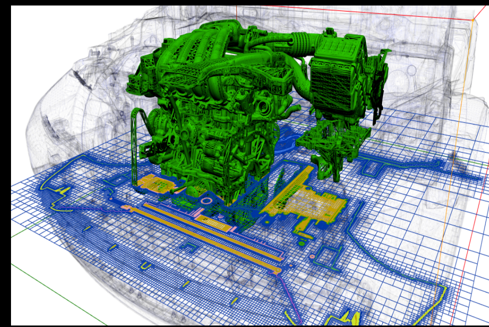
SIMULATION PROCESS IN INDUSTRIAL APPLICATION





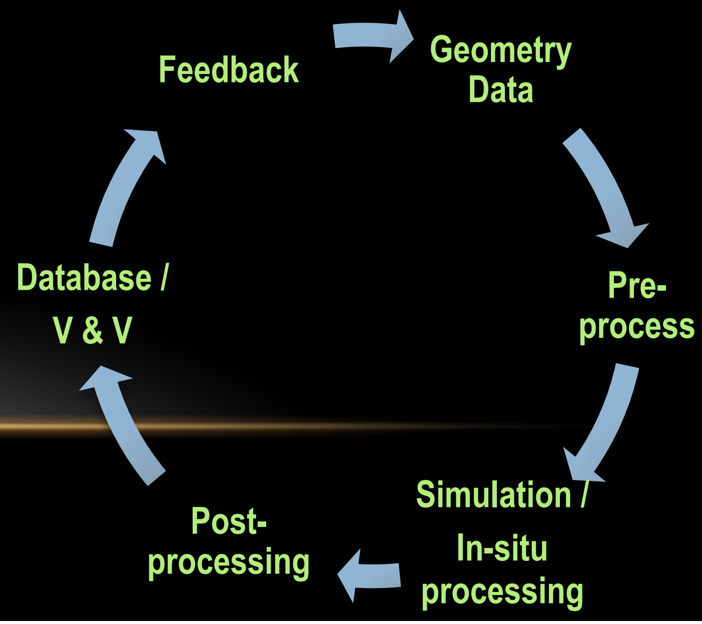
JAXA Oyama





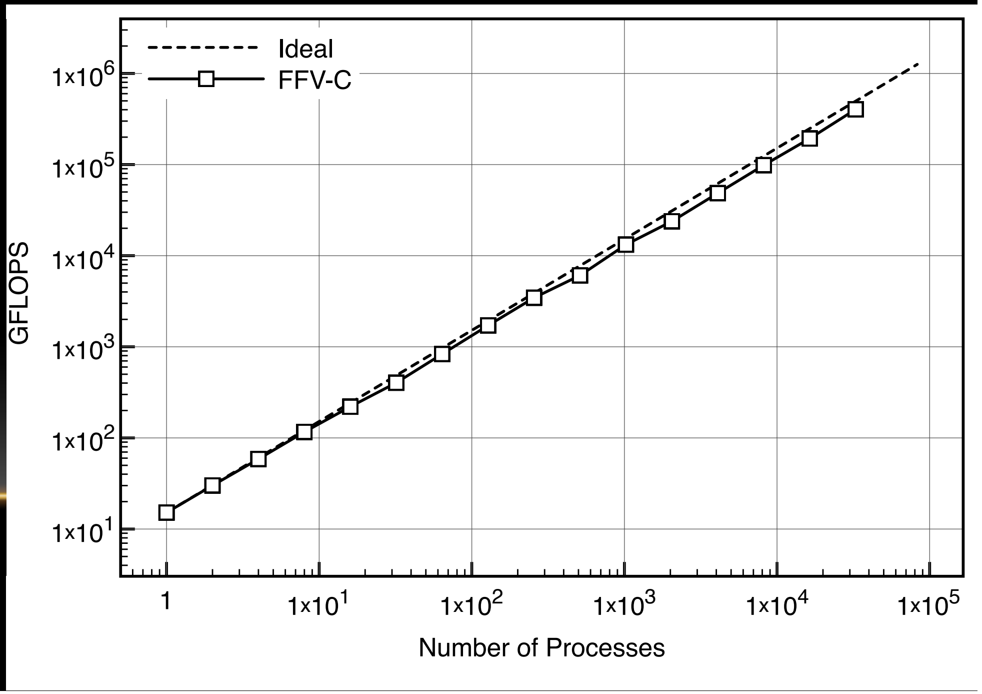
ISSUES TO BE ADDRESSED FOR LARGE-SCALE CFD

- Analysis model
  - Grid generation of 10G-100G range, file based method is distant
- Parallel computation
  - Performance, load balancing
- Post-processing
  - Parallel visualization and data exploration for large-scale dataset
  - Data re-use
- Keys
  - File handling
    - File I/O performance
  - Automation
    - Workflow
  - Database

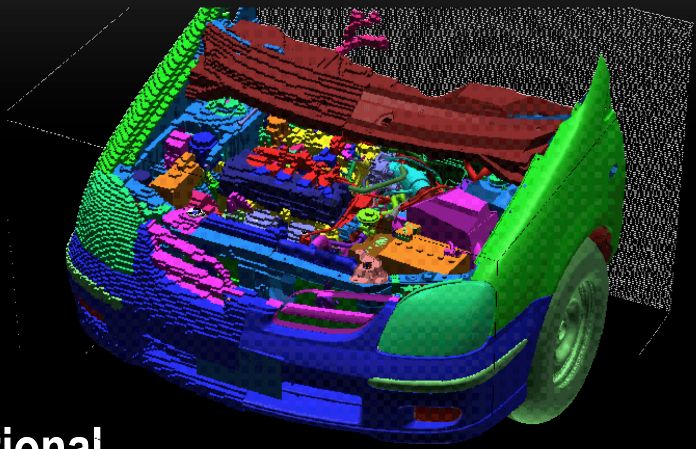


SCENARIO 1 : EXPRESS CFD SIMULATION

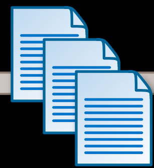
- Grid generation for large-scale simulation
  - Automatic, on the fly generation
  - Cartesian grid base approach
- High-performance solver
  - Hybrid parallel
  - Over 80% at 32768 processes
- Post-process
  - Visualization
  - Data analysis



FROM PRE-PROCESS TO ON THE FLY



- Generate grid before computation
- Quality depends on operators
- Need time to transfer files

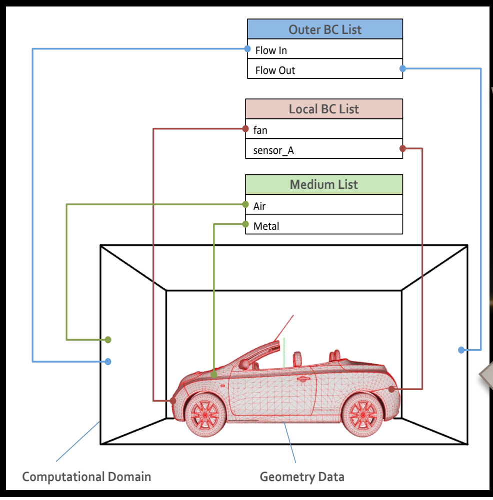


Conventional

Prepare decomposed grid file for a specific number of divisions

Parallel Computation

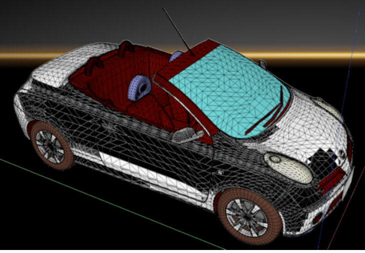
Automatic grid generation




```
DomainInfo {
  Global_origin   = (-0.5, -0.5, -0.5   )
  Global_region   = (1.0,  1.0,  1.0    )
  Global_voxel    = (64   , 64   , 64   )
  Global_division = (1    , 1    , 1    )
  ActiveSubDomain_File = "hoge"
}
```

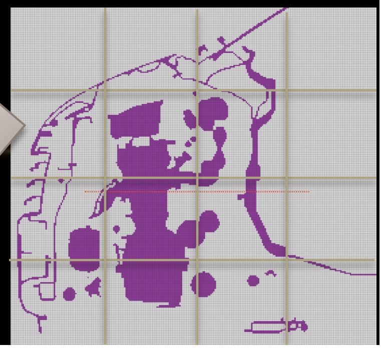
Domain information & BC (Ascii)

Parallel Computation

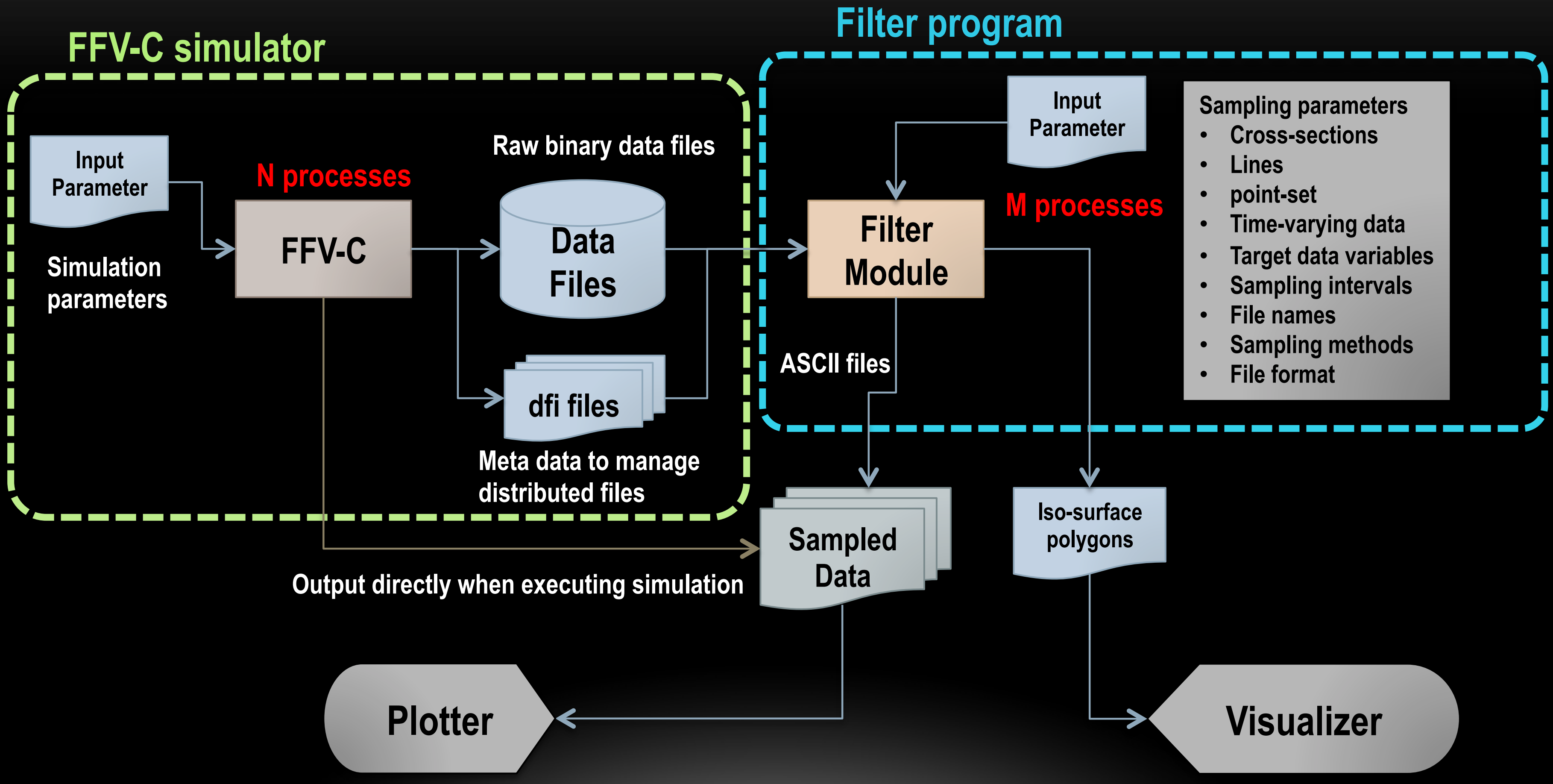


Geometry

Robust Algorithm



POST-PROCESS – DATA SAMPLING



SCENARIO 2 : GRID SEARCH / OPTIMIZATION

- Obtain optimal parameters of design
  - Parameters have trade-offs between performance

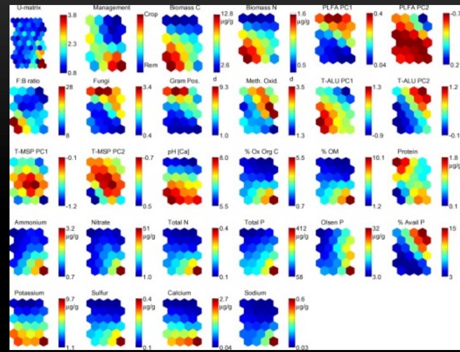
- Design of parameter space

- Automatic execution / retrieve results

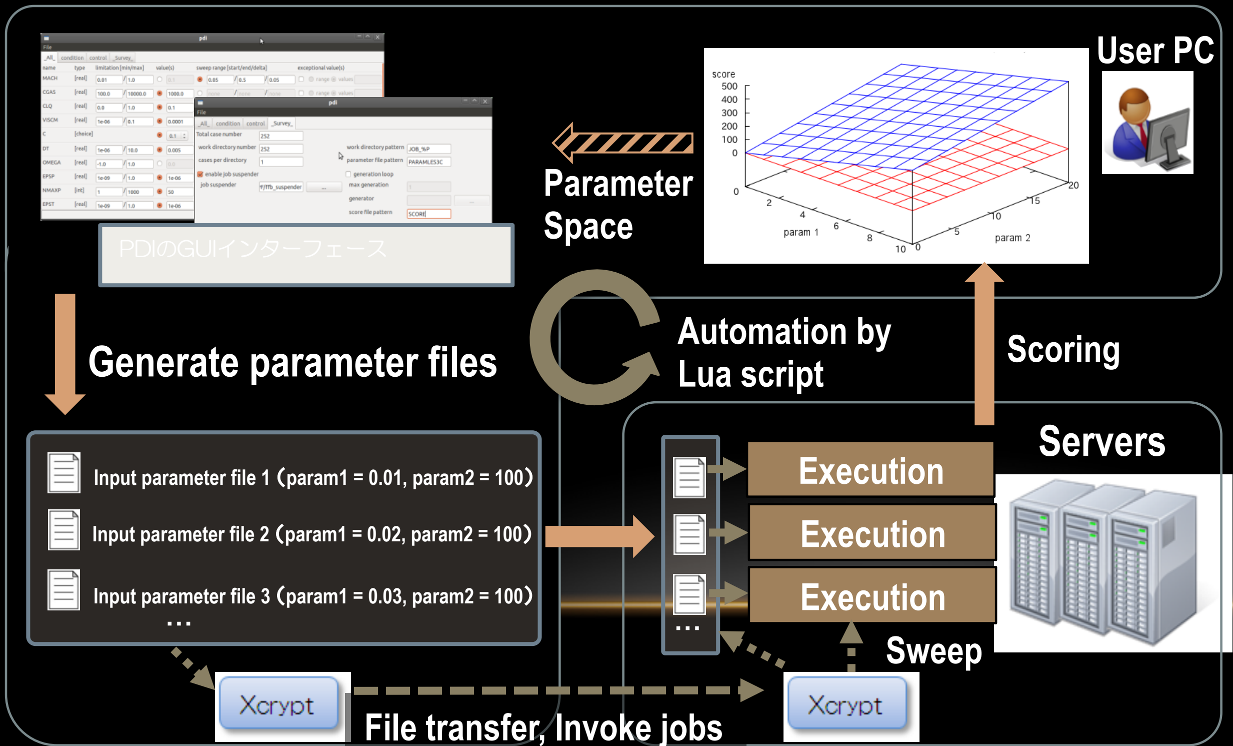
- With optimization engines

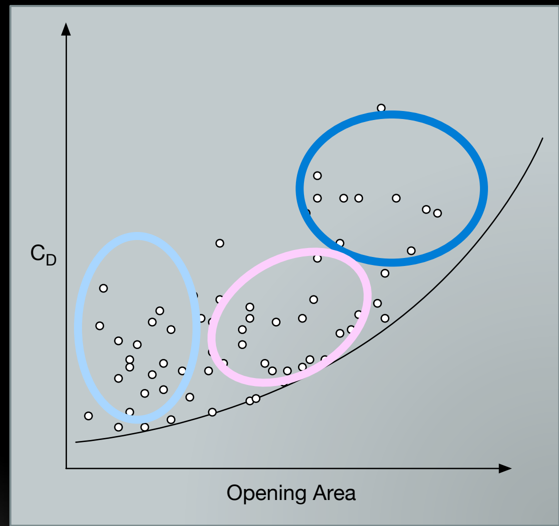
PARAMETER STUDY

- Optimization
- Many calculations for different parameters against design variables
- Search optimal parameters in the parameter space



Sensitivity Map





Clustering Analysis

SCENARIO 3 ; ZERO DESIGN CYCLE TIME

Pratt & Whitney

- Compress leading time of design
  - Compute all cases in parameter space
  - Register results of all cases in DB
  - Then, DB can provide data that is required to design in real-time
- New paradigm of design
  - demands EC and BD

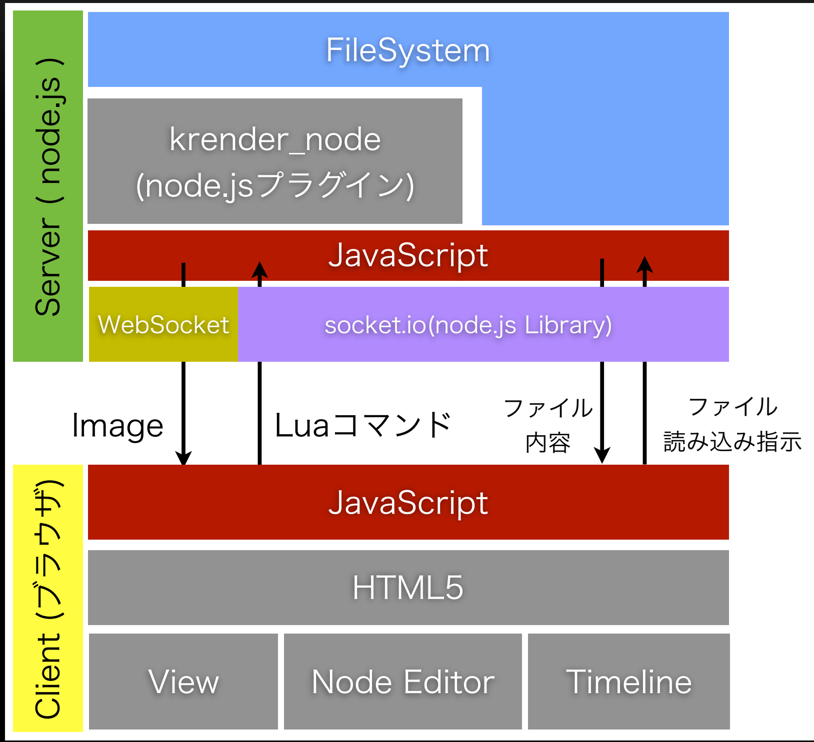
TECHNICAL INFRASTRUCTURE

- Workflow
- Data management
- Database

AUTOMATION

- Workflow
  - Script
  - Multi-platform
  - Can be operated on remote environment

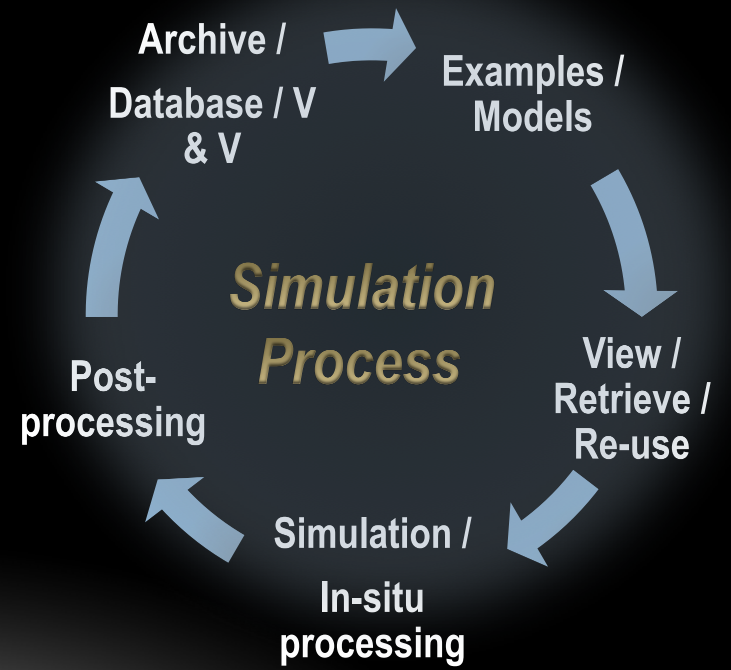
- Lua script
  - powerful, fast, lightweight, embeddable



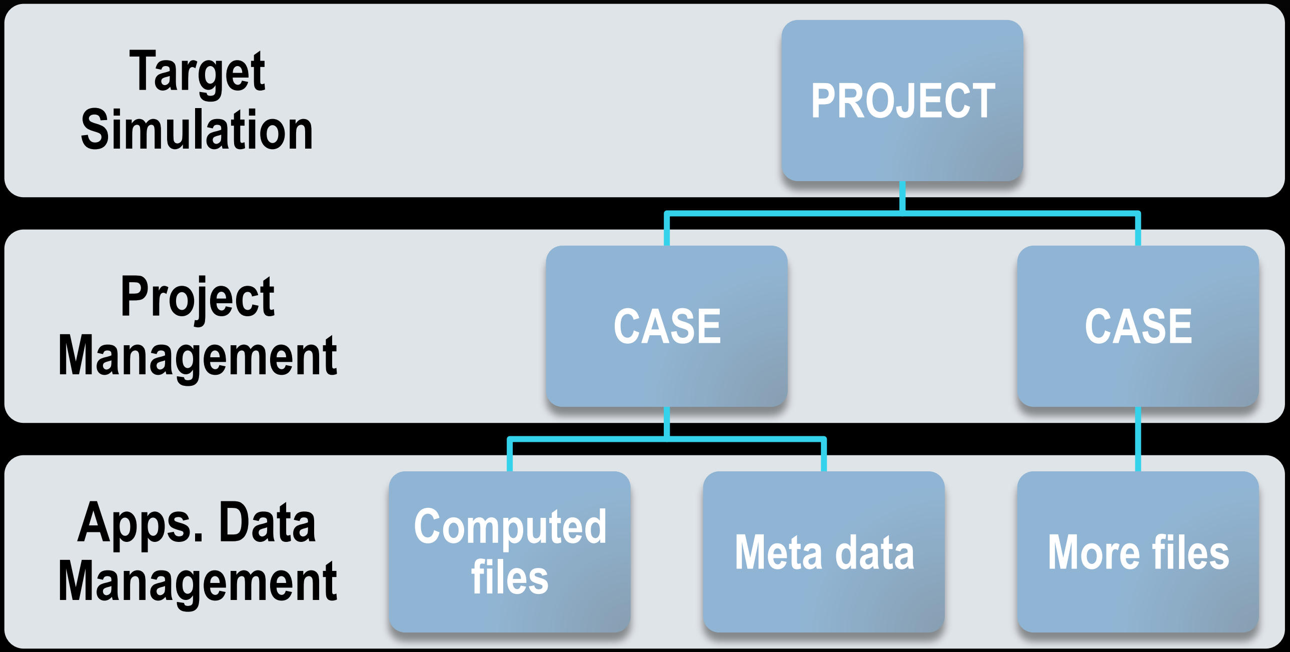
PROJECT DATA MANAGEMENT

- Resource management of a project
  - all information; HW info., input files, calculated result files, and derived files
  - Case
    - a unit of execution of a simulation
  - Project
    - a set of cases

- Data management enables us to
  - automatic processing
  - collaboration with database
  - grid search
  - provenance tracking

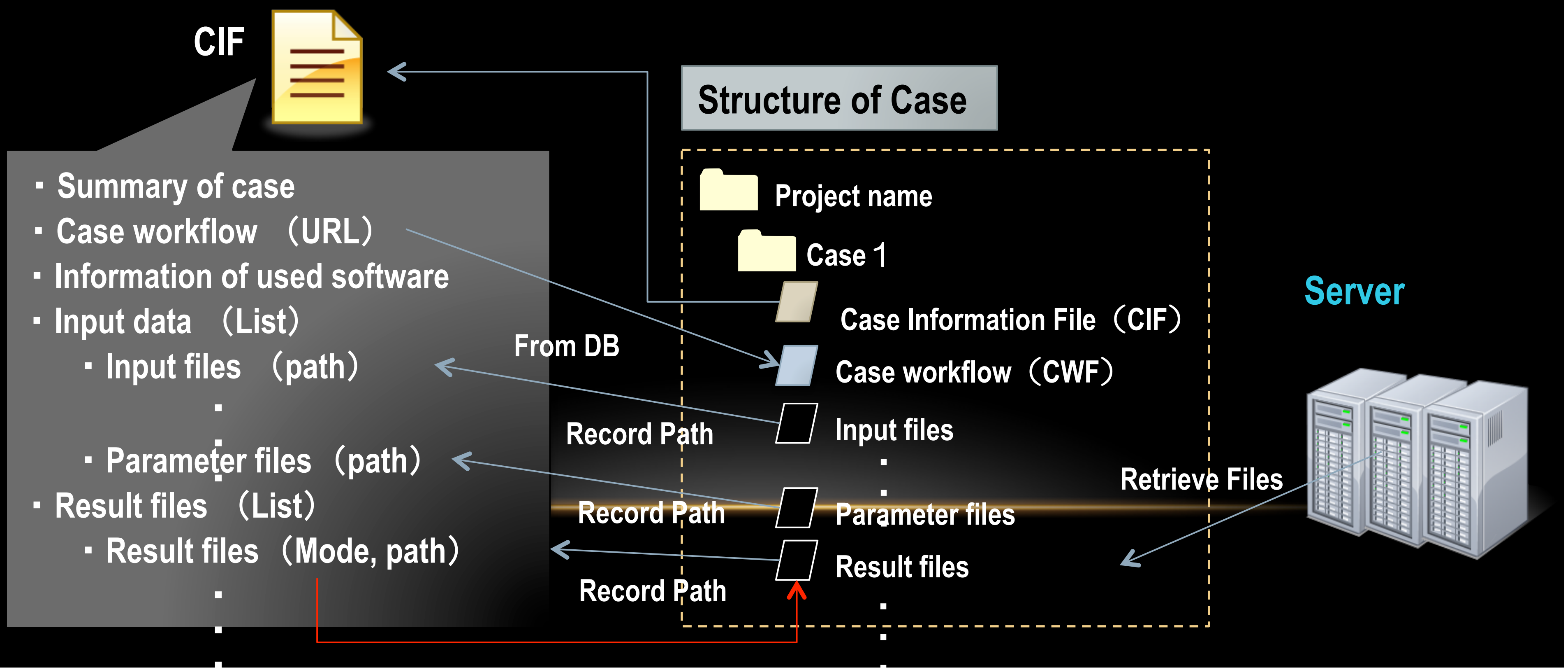


HIERARCHY OF DATA



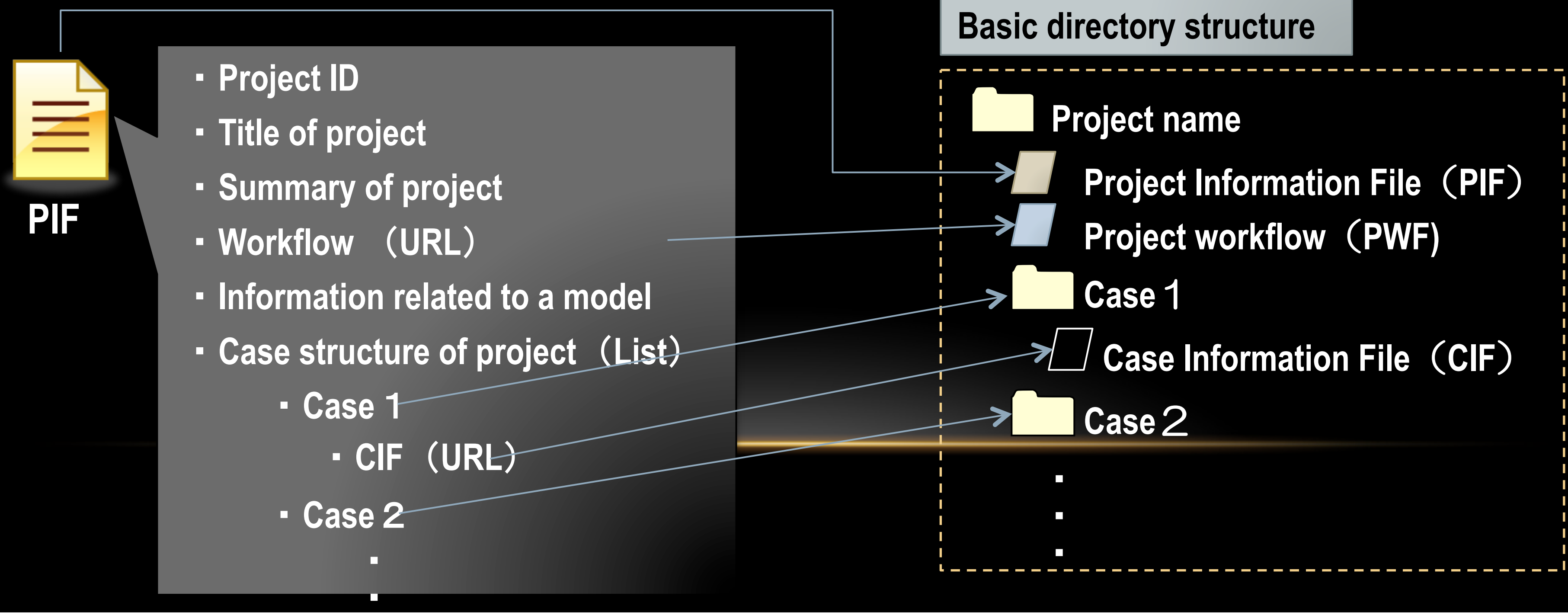
CASE INFORMATION FILE

- Case
  - a unit of execution of a simulation
  - Case Information File (CIF) describes contents



PROJECT INFORMATION FILE

- Project
  - a set of cases
  - Project Information File (PIF) describes contents



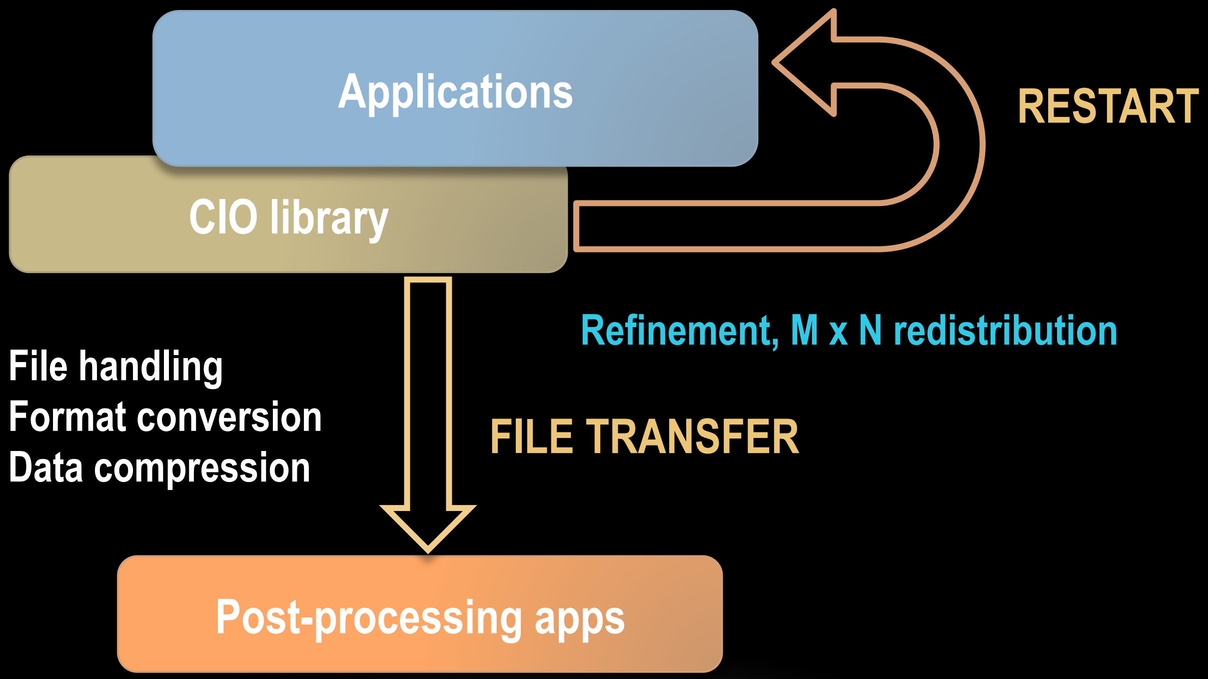
APPLICATION DATA MANAGEMENT

- It is important to design a way of management for domain specific applications
  - For each data structure
  - Use-case scenarios; Restart, Data transfer between apps.

- Example: Distributed file management for domain decomposition based simulation on Cartesian data structure
  - Directory management
  - Restart
  - Mutual exploitation of file I/O between a simulator and a post processing

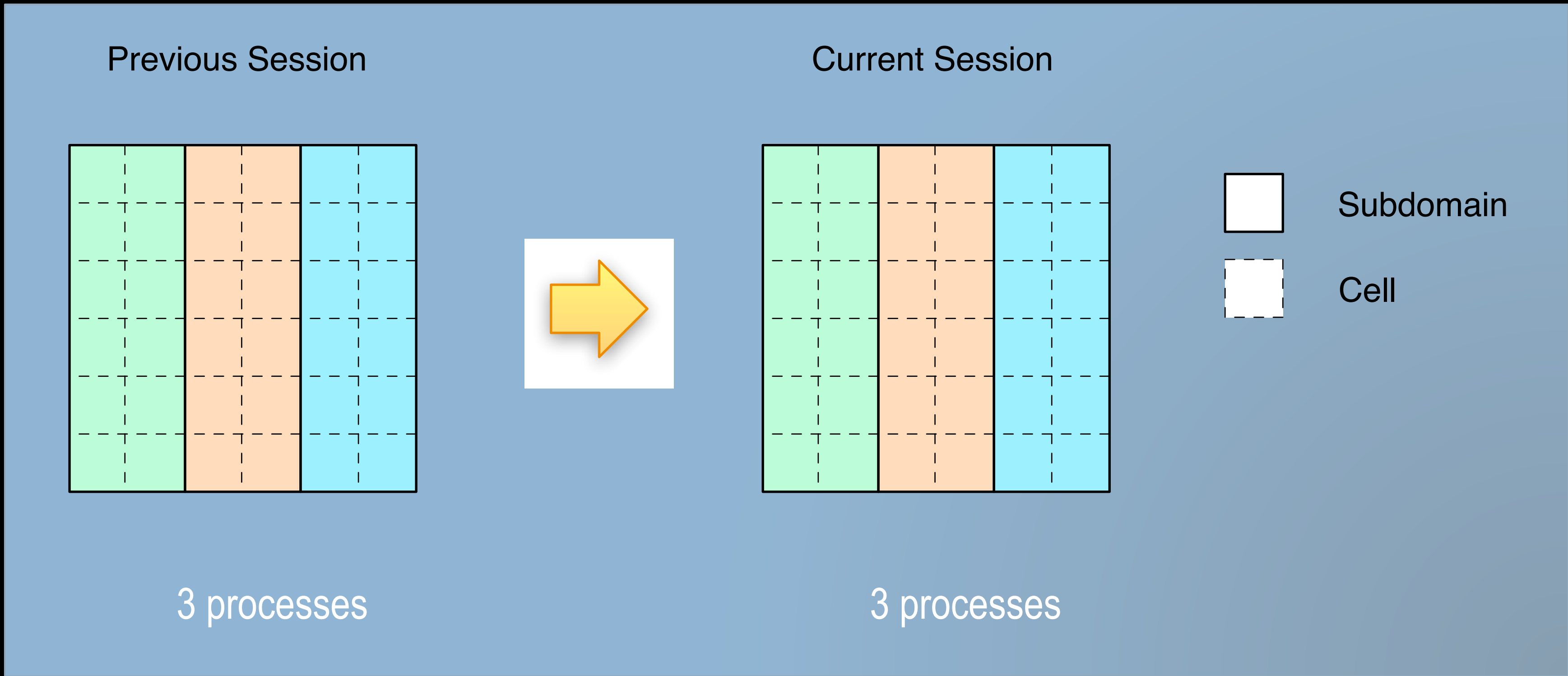
CIO (CARTESIAN I/O) LIBRARY

File management function for Cartesian data structure on distributed parallel environment



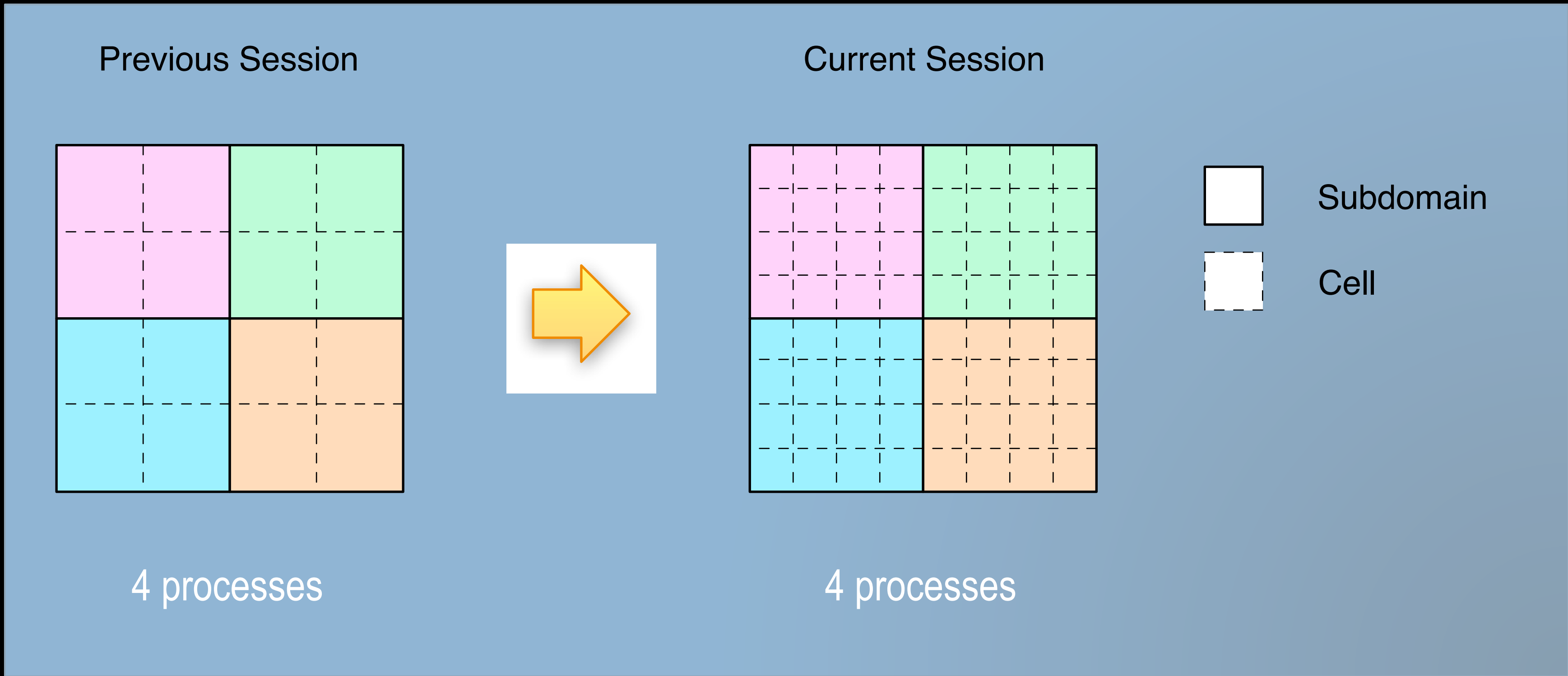
USE CASES 1

Same # of processes, same resolution => Std. restart



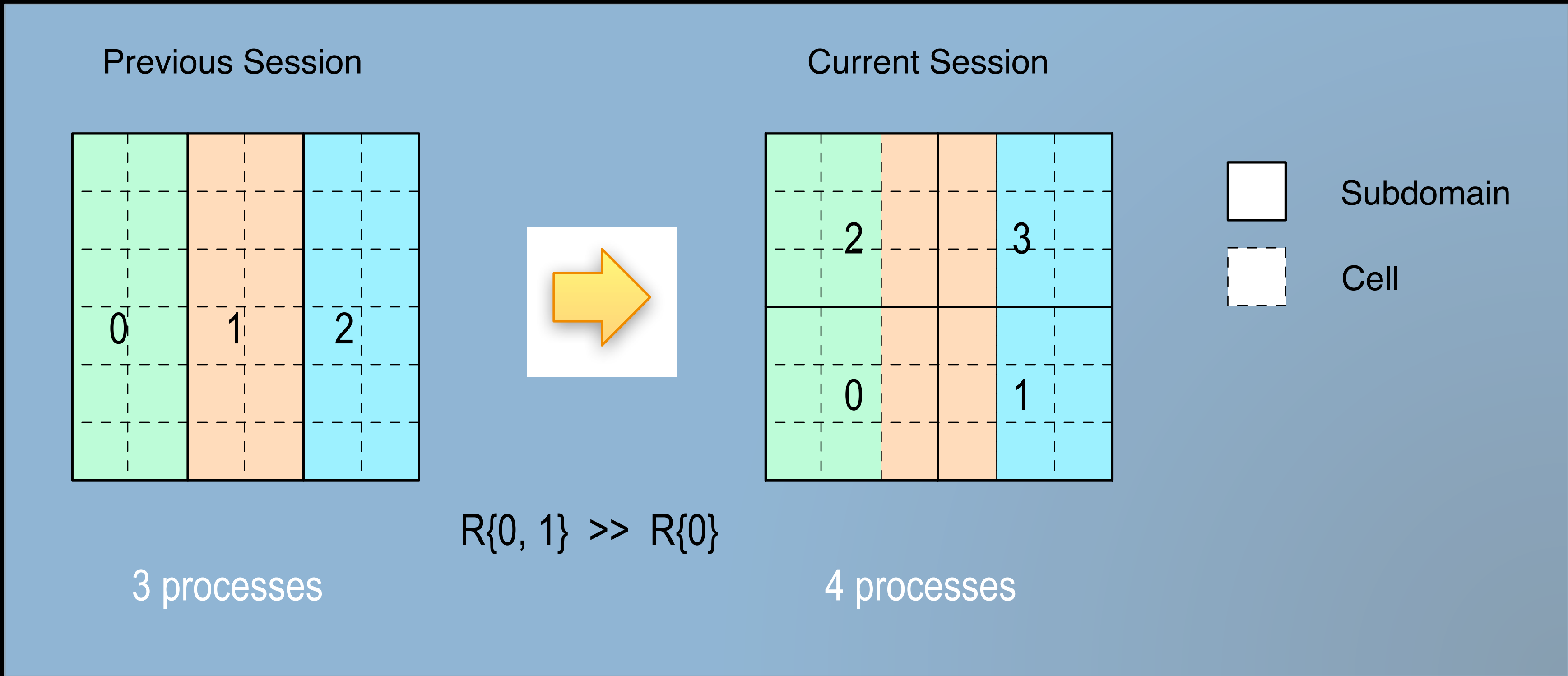
USE CASES 2

Same # of processes, different resolution => Refinement restart



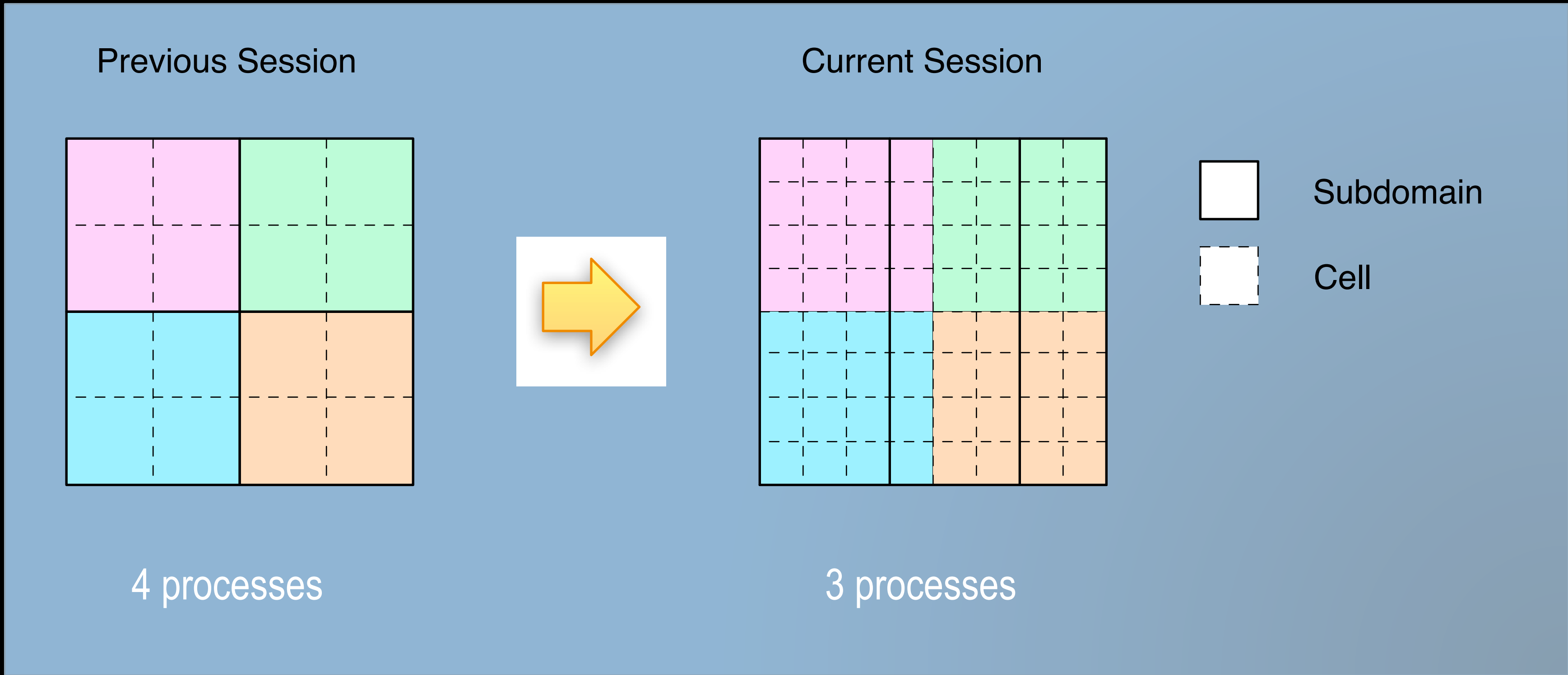
USE CASES 3

Different # of processes, same resolution => M X N restart

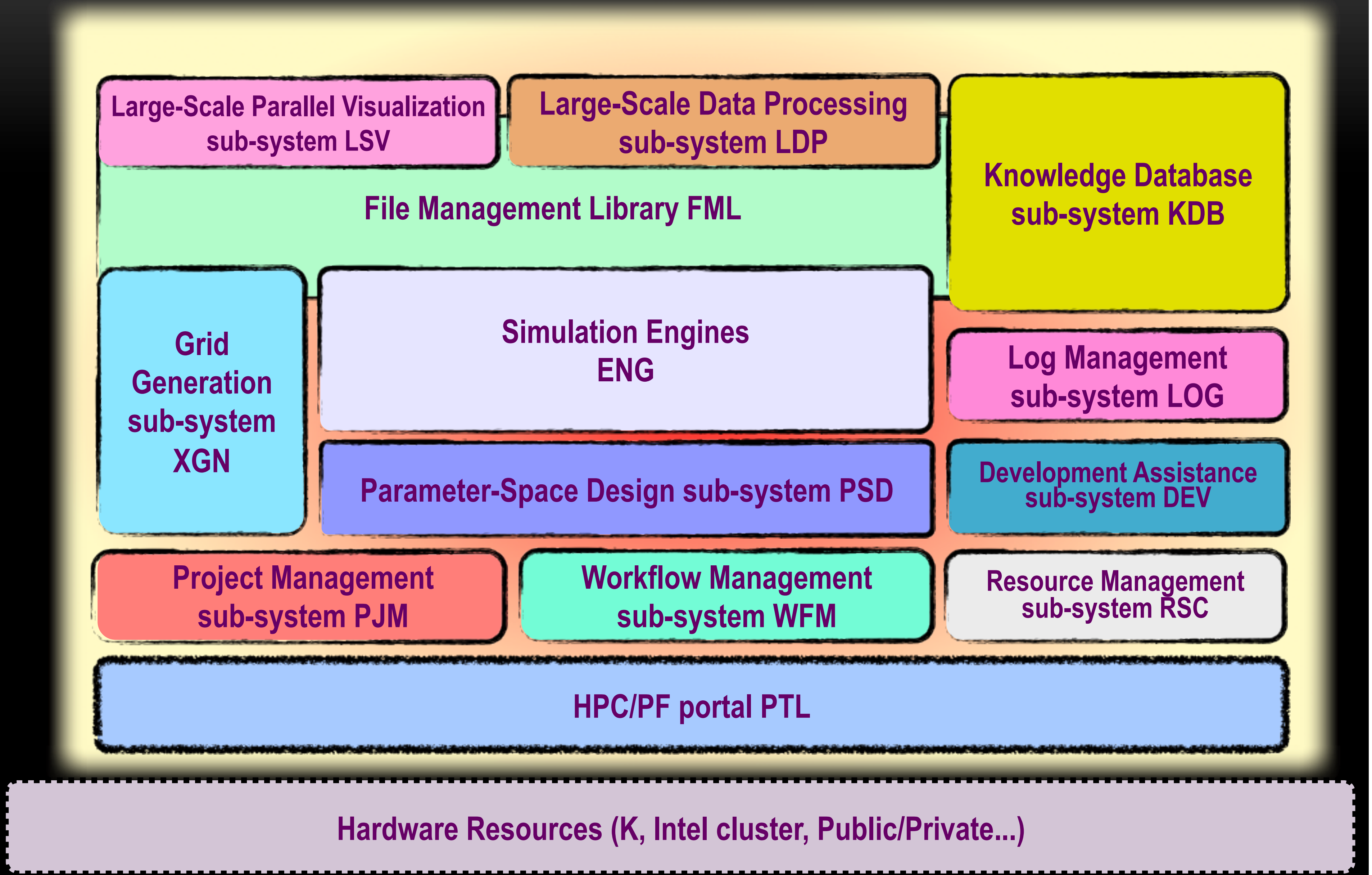


USE CASES 4

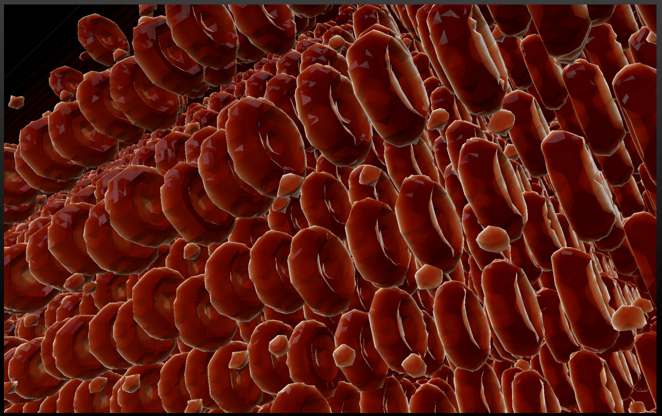
Different # of processes, different resolution => M x N /w refinement



COMPONENTS OF EXECUTION ENVIRONMENT

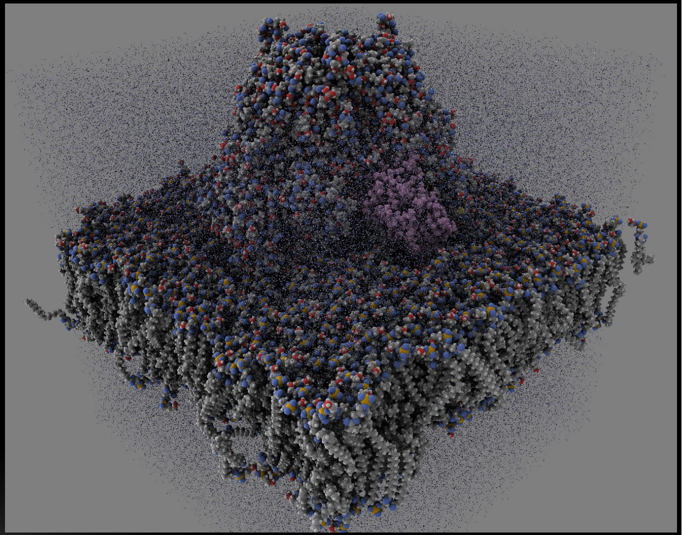


VIS. SYSTEM ON K-COMPUTER



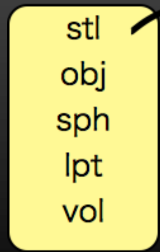
- Handle large-scale distributed data files
  - CIO library
- Direct rendering on K
  - Common rendering core for both on PC (/w GPU) and on K >> GLSL/GLES API, not OpenGL
  - Ray tracer and Volume renderer

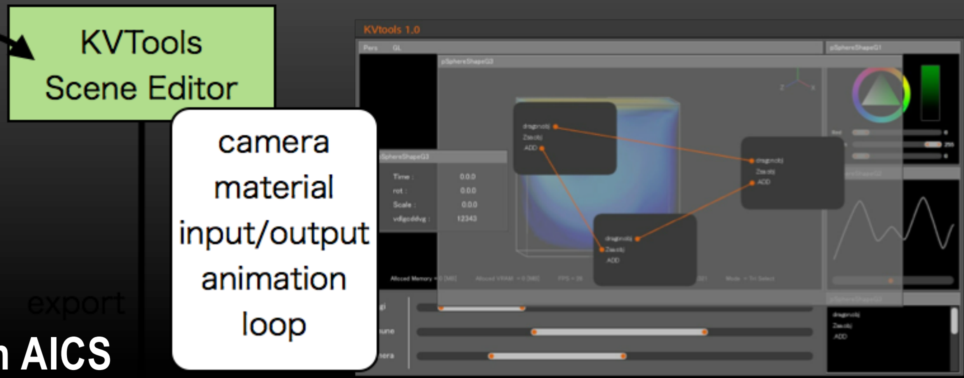
Rasterlizer -> O(N)
Ray tracer -> O(logN)



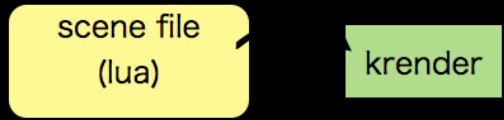
  - Sort-last type parallel renderer
  - For Cartesian, UNS, particles data structure
  - Currently, batch and interactive(x86 /w GPU)
  - Bring exascale into view

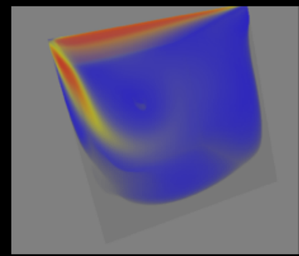
KVTOOLS





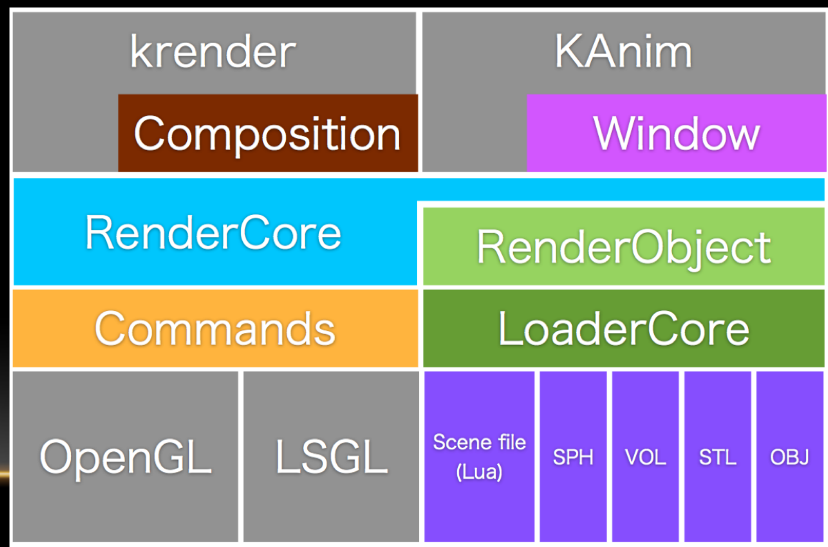
- Developing parallel visualization system in AICS
  - Scene Editor for visualization scenario
  - krender : image generation
  - Can be operated on local or remote machines
  - Batch job with visualization scenario





Performance of Volume renderer on K

32k Parallel, 8192^3 volume, 16k x 8k image
>> 6 min / image



Parallel renderer   Interactive renderer / GUI

FLOW WITH DEFORMED RBC

ZZ-EFSI

Prof. Takaki & Sugiyama @UT

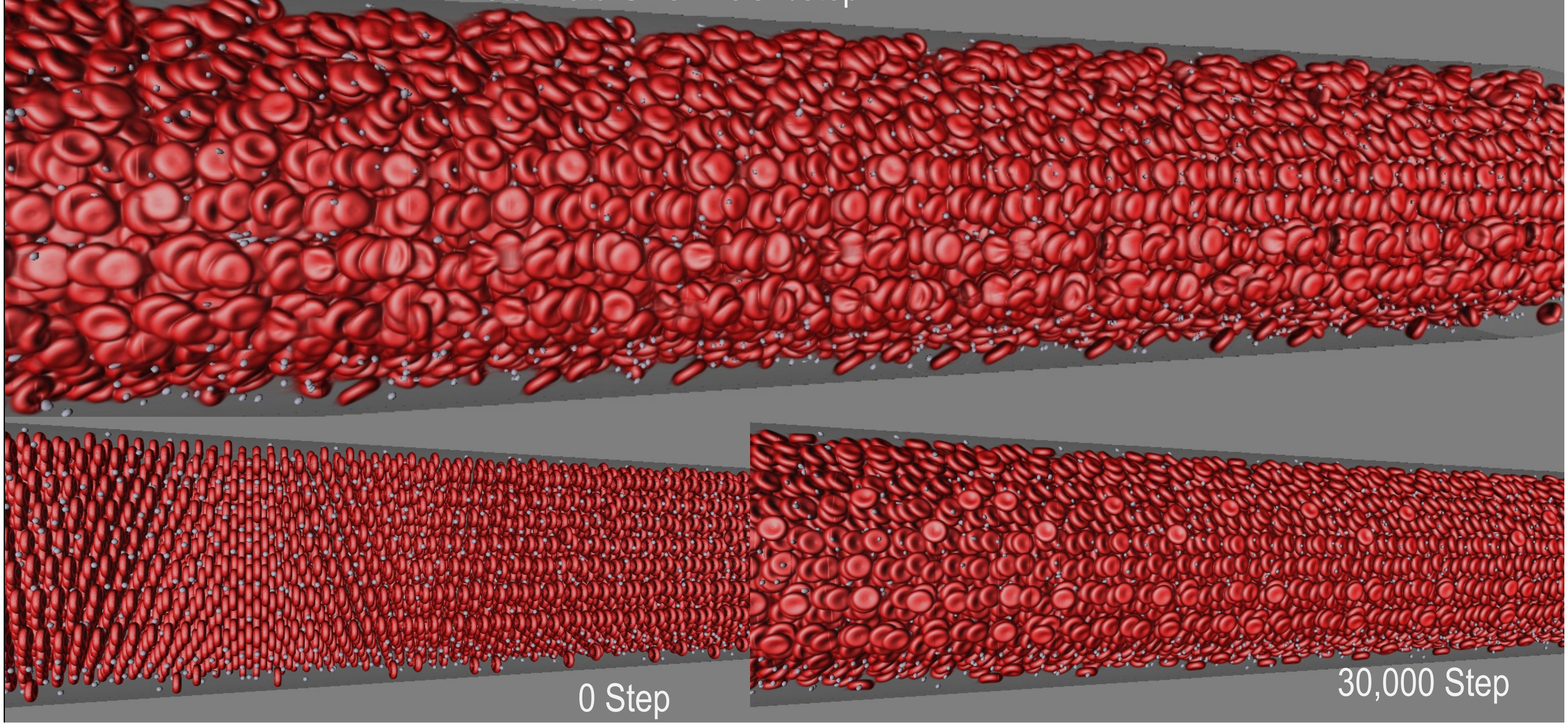
70,000 Step

Voxels per Domain : 66 x 66 x 66

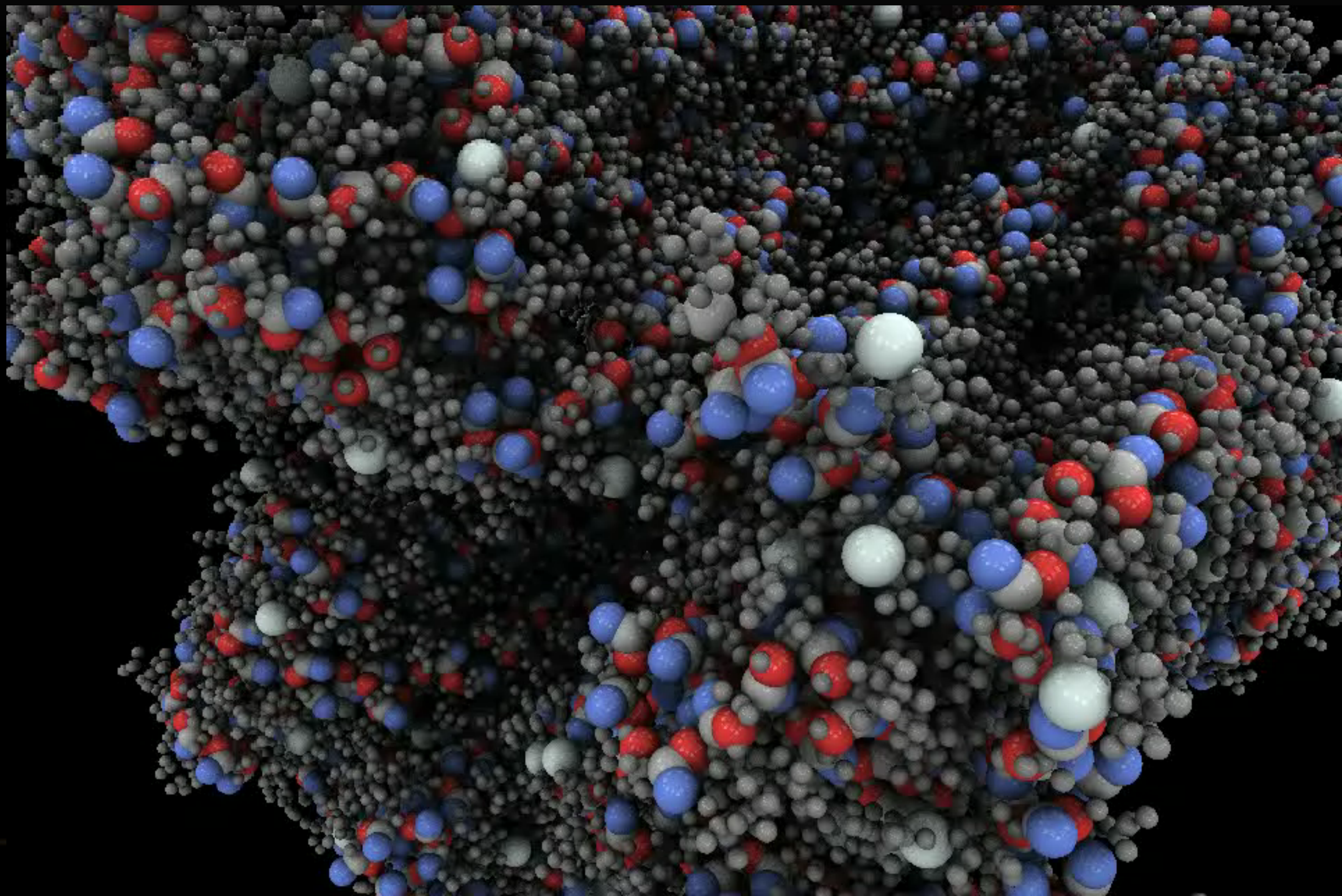
Num of Domain : 4,800

Num of Data : Scalar X 3(Red Blood Cell, Platelet, Blood Vessel Wall)

Total Data Size : 45GB/step



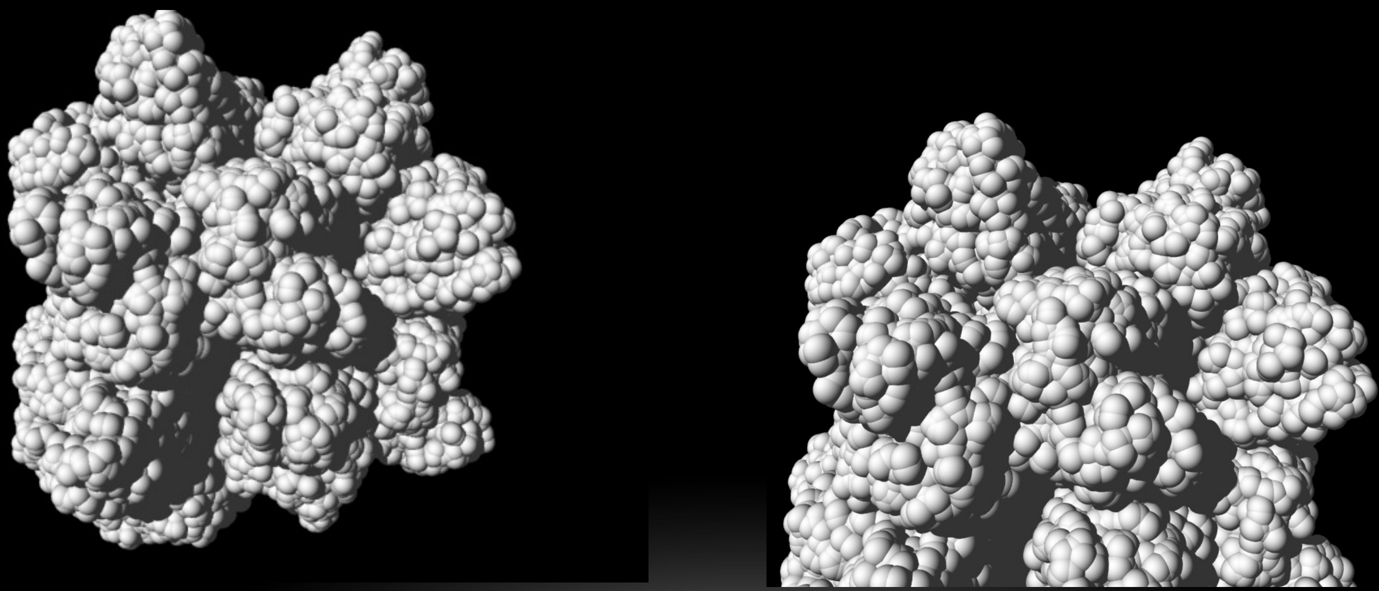
HIGH RES. RENDERING IMAGE



OFF-LINE RENDERING OF PDB DATA

Data :

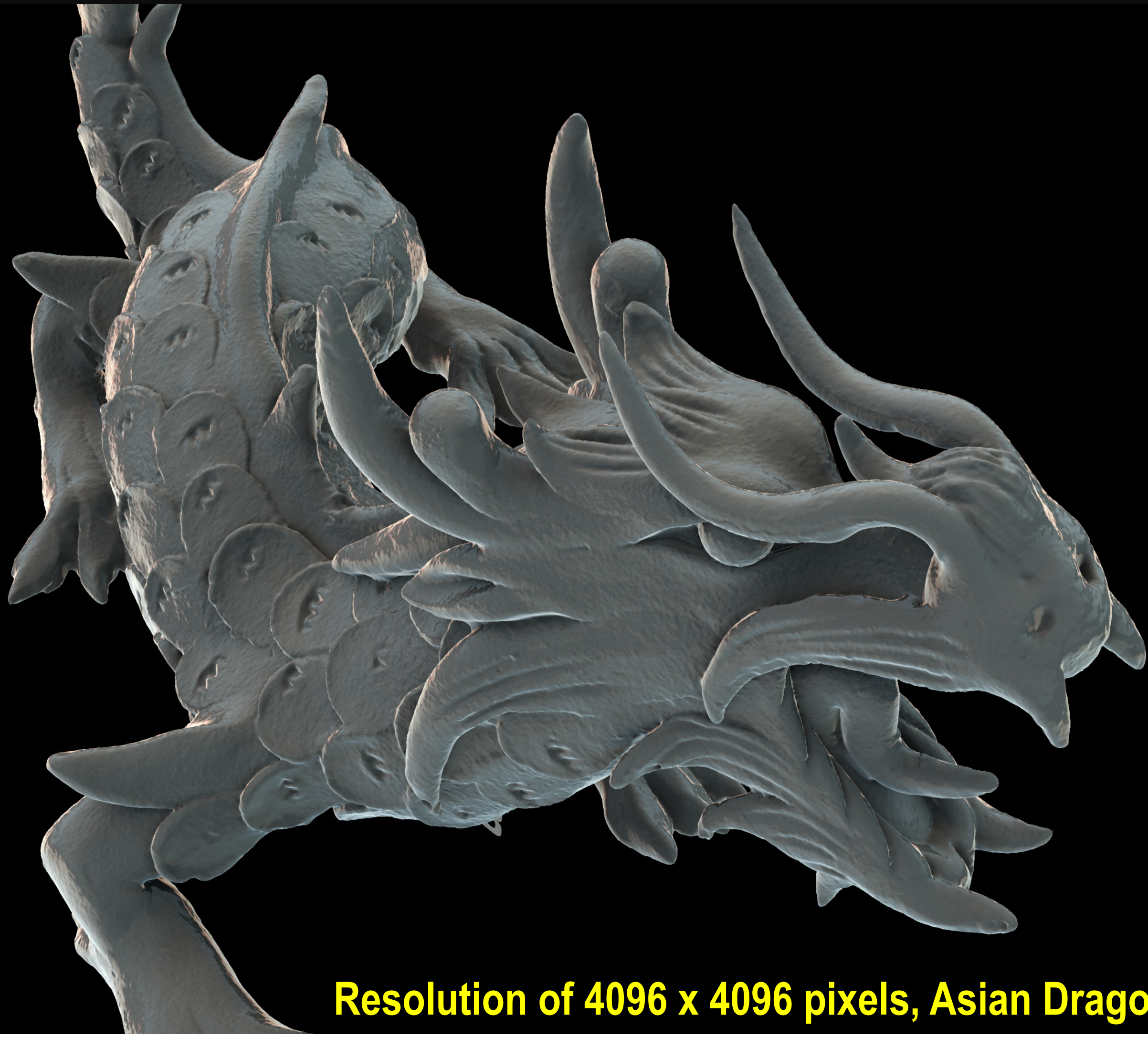
- http://www.rcsb.org/pdb/explore.do?structureId=1mt5
- **Only Atom, 1M**
- **Rendering point primitives with Lambert shader and ray casting**



Result on Intel PC　　　Result on K

Fujita (2013)

RENDERING GLSL ON K



Resolution of 4096 x 4096 pixels, Asian Dragon

CONCLUDING REMARKS

- Exascale computer project has just started.
  - The architecture is not fixed yet.
  - Power efficiency and co-design play a important role.
- Exascale application
  - Useful execution environment will be required for practical problems.
  - Data management and workflow plays an important role than ever.