Dynamic Domain Decomposition Using an eXtensible Data Model and Format

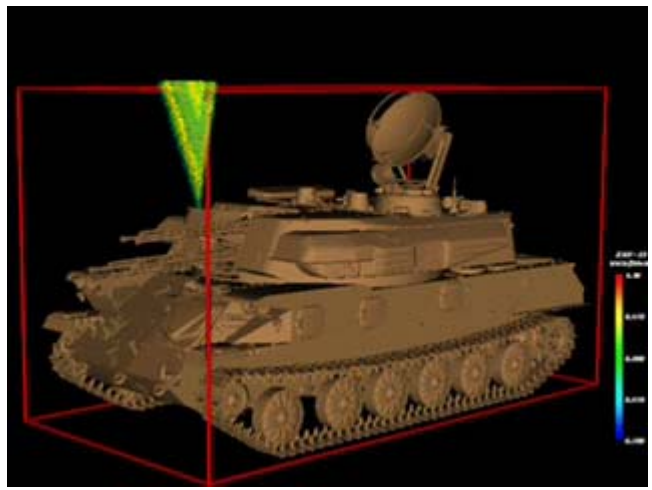
Simplifying Partitioning Complexities by using a Common Data Hub

Jerry Clarke Raju Namburu

ZSU23-4
(Zenitnaia Samokhodnaia Ustanovka) self-propelled anti-aircraft gun

Crew: 4
Length: 258 in
Height: 88.5 in, 149.5 in w/radar
Width: 116 in
Weight: 22.5 tons
Engine: V-6R 6 cyl. in-line diesel
Transmission: Manual 5 sp forward, 1 rev
Steering: Clutch and brake
Suspension: Christie
Smoke System: None
Fuel: 66.5 gal
Armor: Hull 0.6 in at 55 deg, Turret 0.35 in at 15 deg

Main Gun: Quad AZP-23 23mm cannons



Scalable Mesh Generation

Surface Scan of Existing Objects

CAD Models

Facets/Polygons

↓

Wedges/Thickness

↓

Define Implicit Structured Mesh

↓

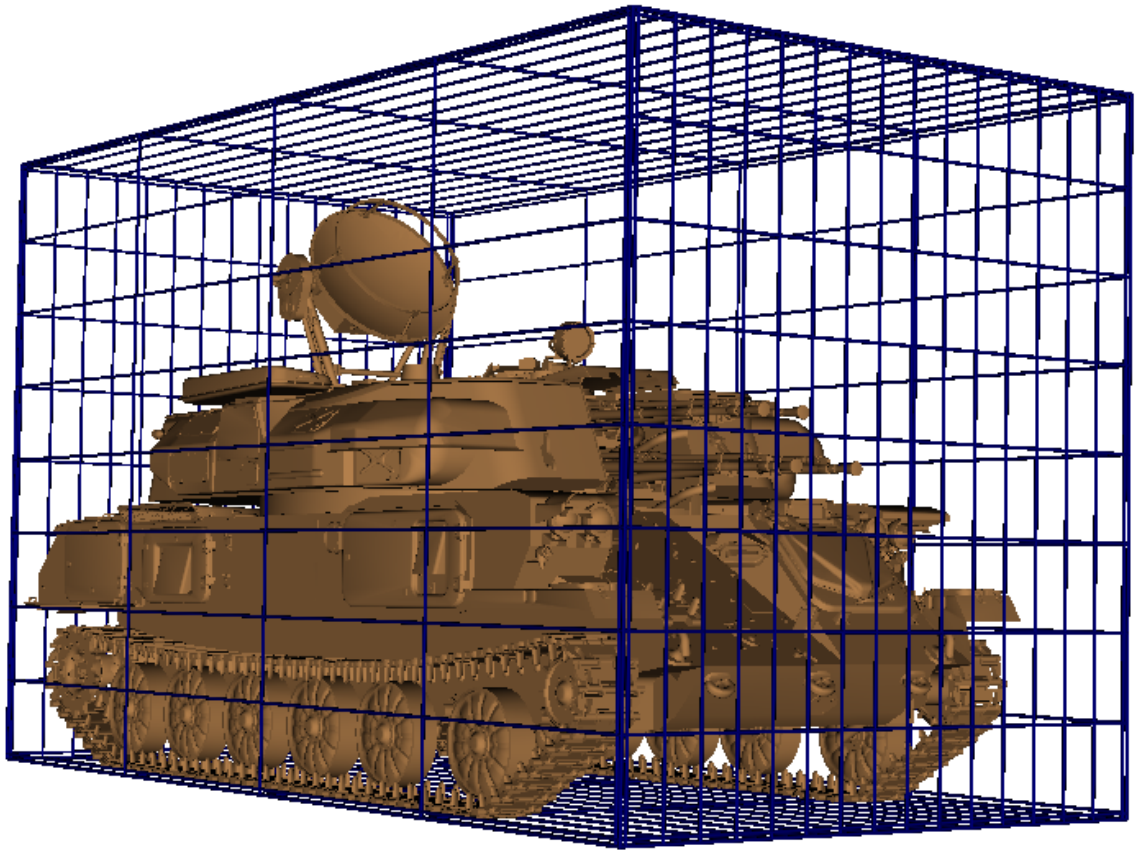
Domain Decomposition

↓

Parallel Generation of the Mesh

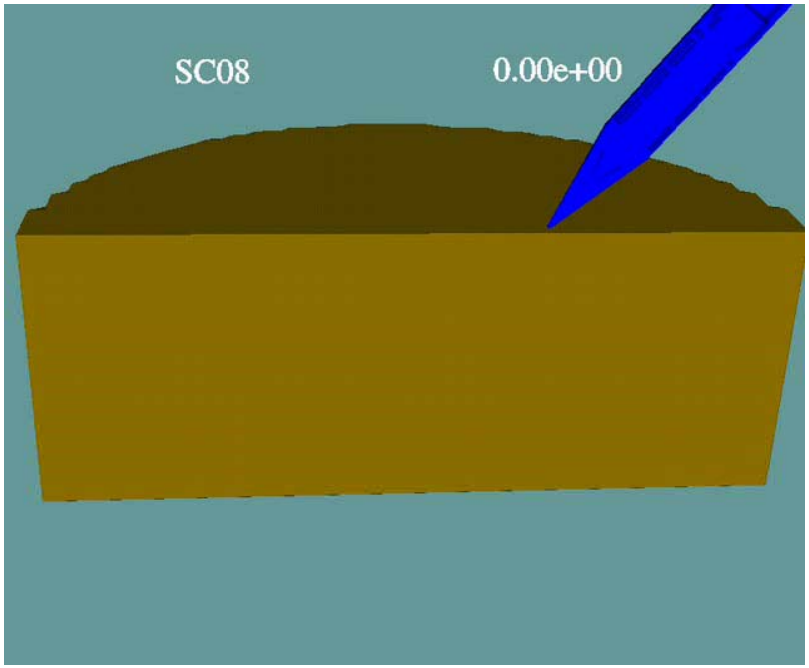
(number of PEs for mesh generation does NOT have equal the number of PEs for solver)

Partitioning / Decomposition



Partitions: 6 x 17 x 10
Cells: 1984 x 1043 x 1188
(2.46 billion)

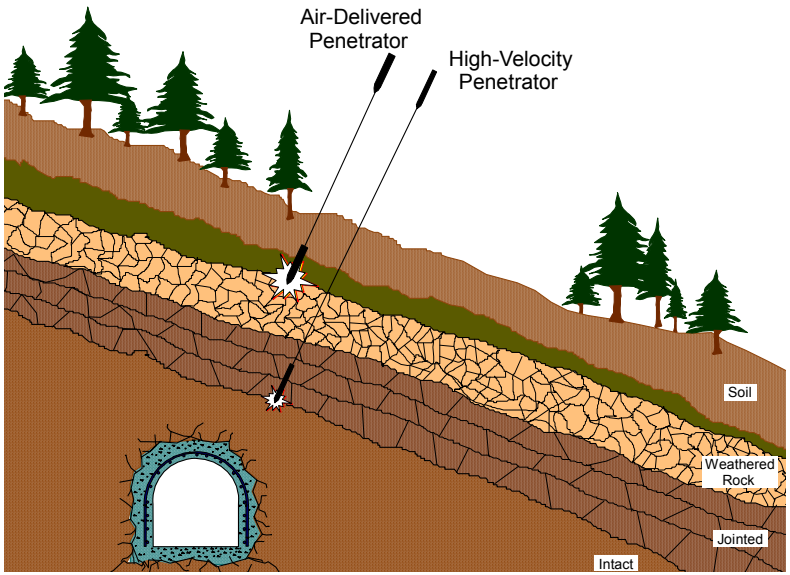
Run on 384 PE's on the ARL-MSRC IBM-SP3



Penetrator – Target Interaction

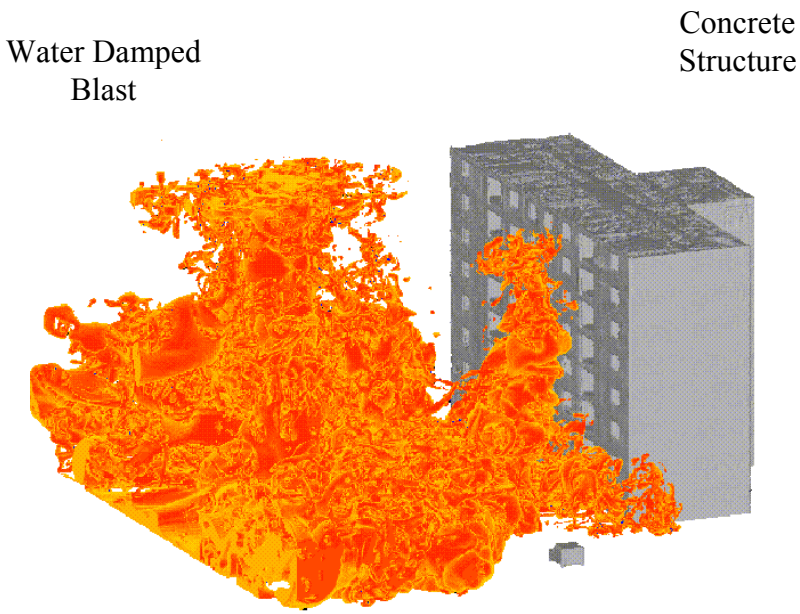
Blast – Structure Interaction

Penetrator – Target Interaction



Solving Same Physics
in Different Domains
Different Numerical Approach
in Each Domain
(CSM in Penetrator and Ground)

Blast – Structure Interaction



Solving Different Physics
in Different Domains
Different Numerical Approach
in Each Domain
(CFD for Blast – CSM for Structure)

WTI-1

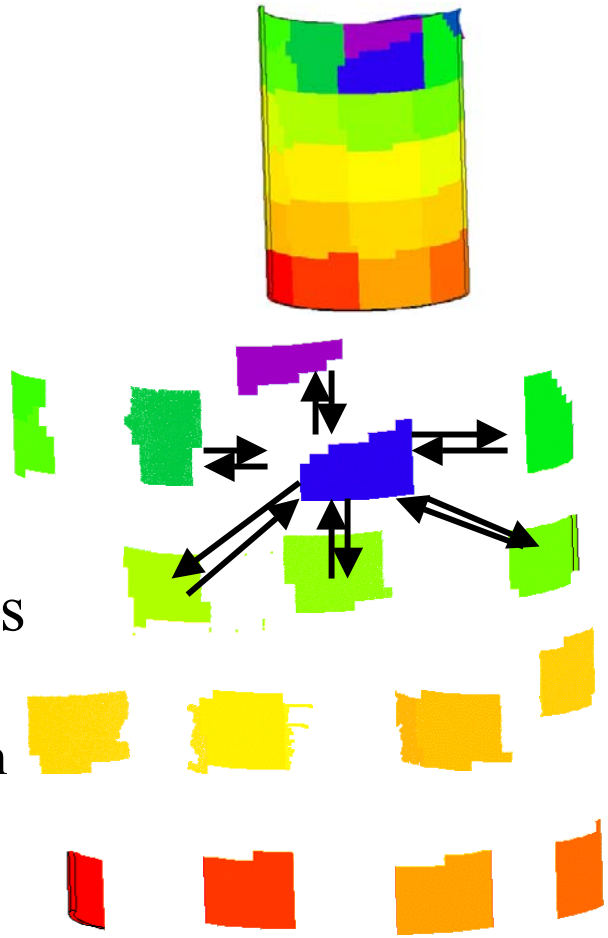
- Zapotec
- F90, F77, C
- Eulerian + Lagrangian
- Currently CTH + Pronto3D
- Inserts / Deletes CTH Material

WTI – 2

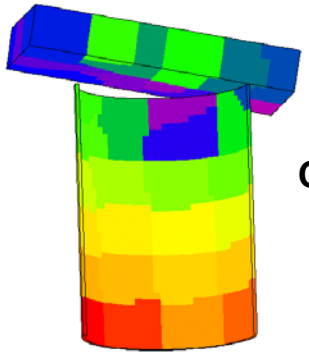
- Virtual Test Facility (VTF)
- Python Framework
- Fluid (AMR) + Solid (FE)
- Currently ARM3D + ADLIB
- Uses Ghost Region to Exchange Interface Information

PRONTO Message Passing

- Enablers
  - Mesh decomposition tools (Chaco)
  - Message passing interface standards (MPI)
  - Scalable to 1000s of processors
  - Standard parallel FEM techniques
    - Each processor gets a small piece of the problem to solve with information swapped along the processor boundaries

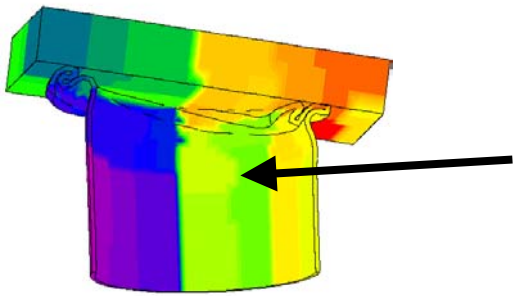


PRONTO - Parallel Contacts

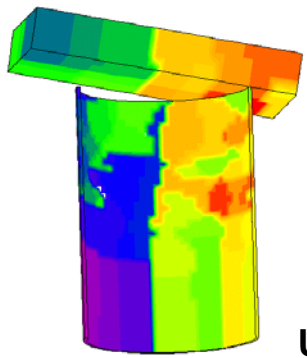


Chaco Element Decomposition

Two different decompositions are used: one for finite elements and one for surfaces

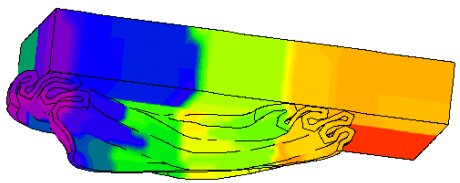


Meshes in these regions are distant in the finite-element decomposition and in geometric decomposition.

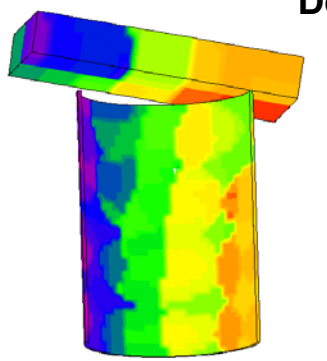


Undeformed Geometric Decomposition

Deformed Surface Geometric Decomposition

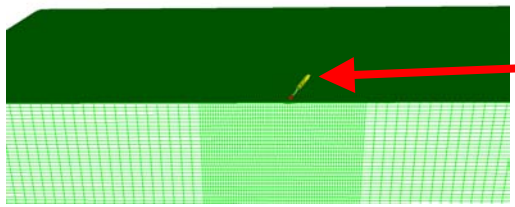


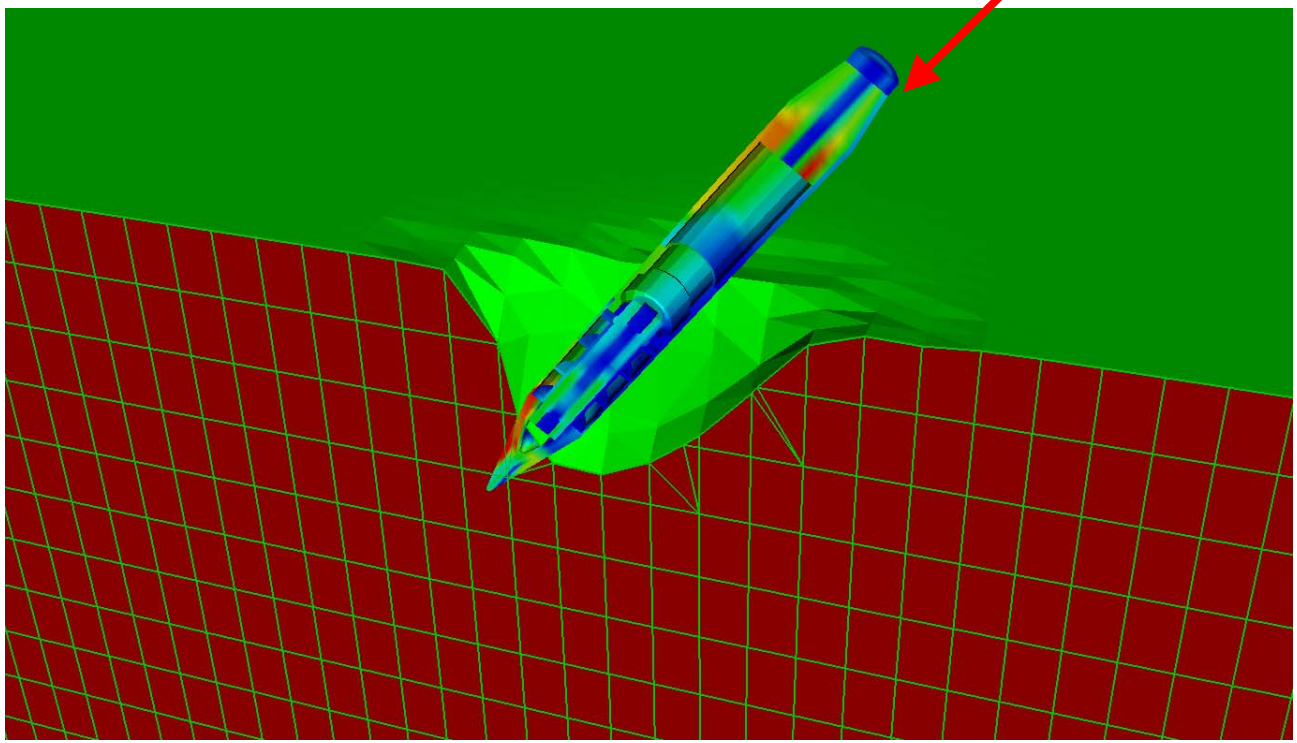
The surfaces are now geometrically close, but are still distant in the finite-element communications topology.



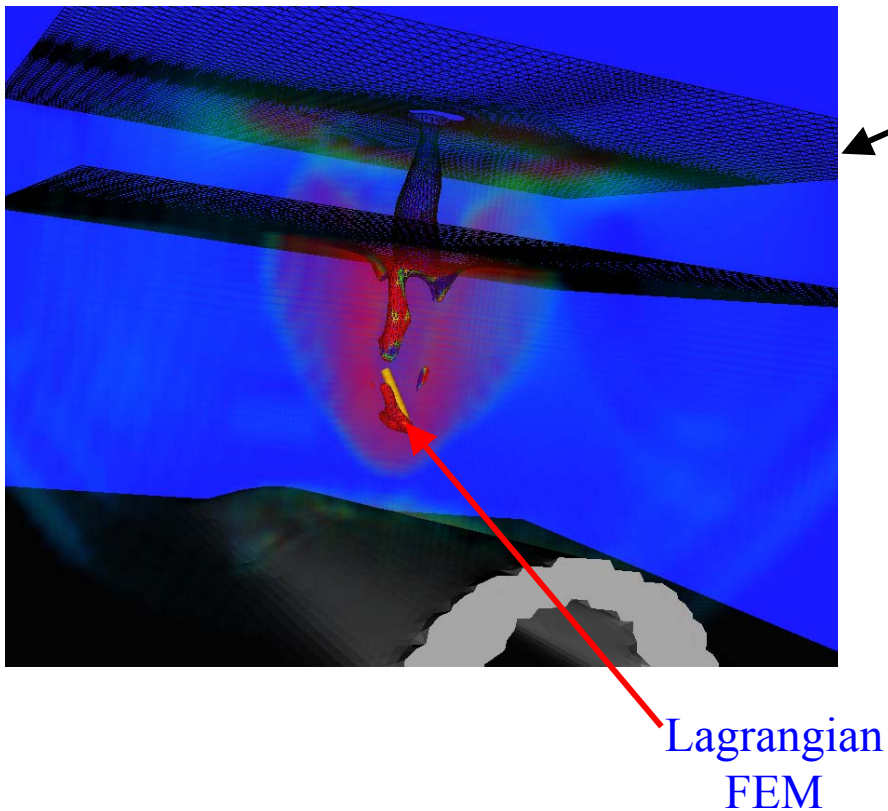
Clay
- Eulerian Finite Volume Code
- Large Structured Mesh

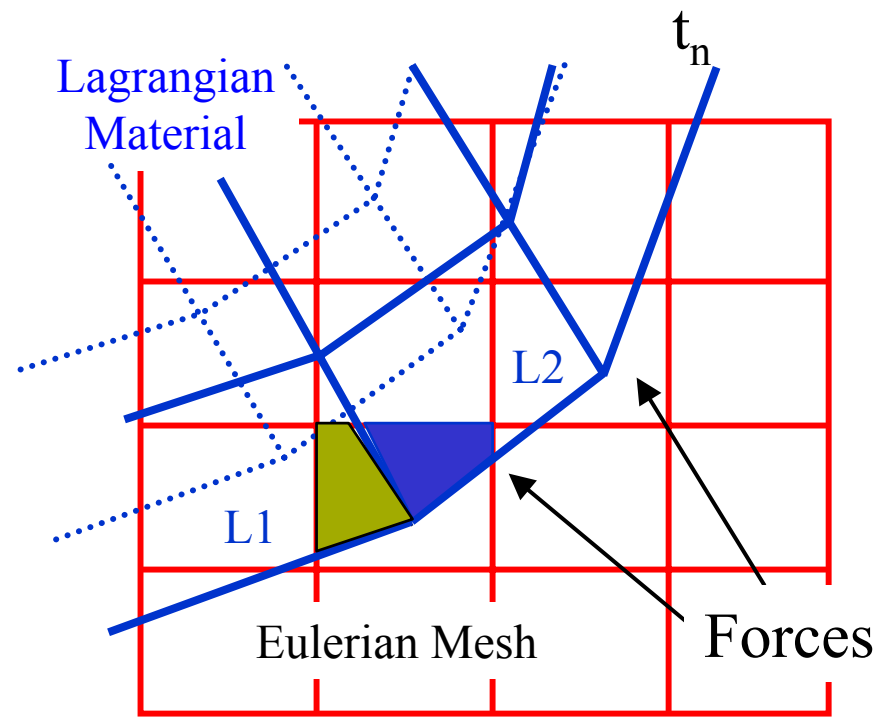
Projectile
- Lagrangian Finite Element Code
- Detailed Unstructured Mesh





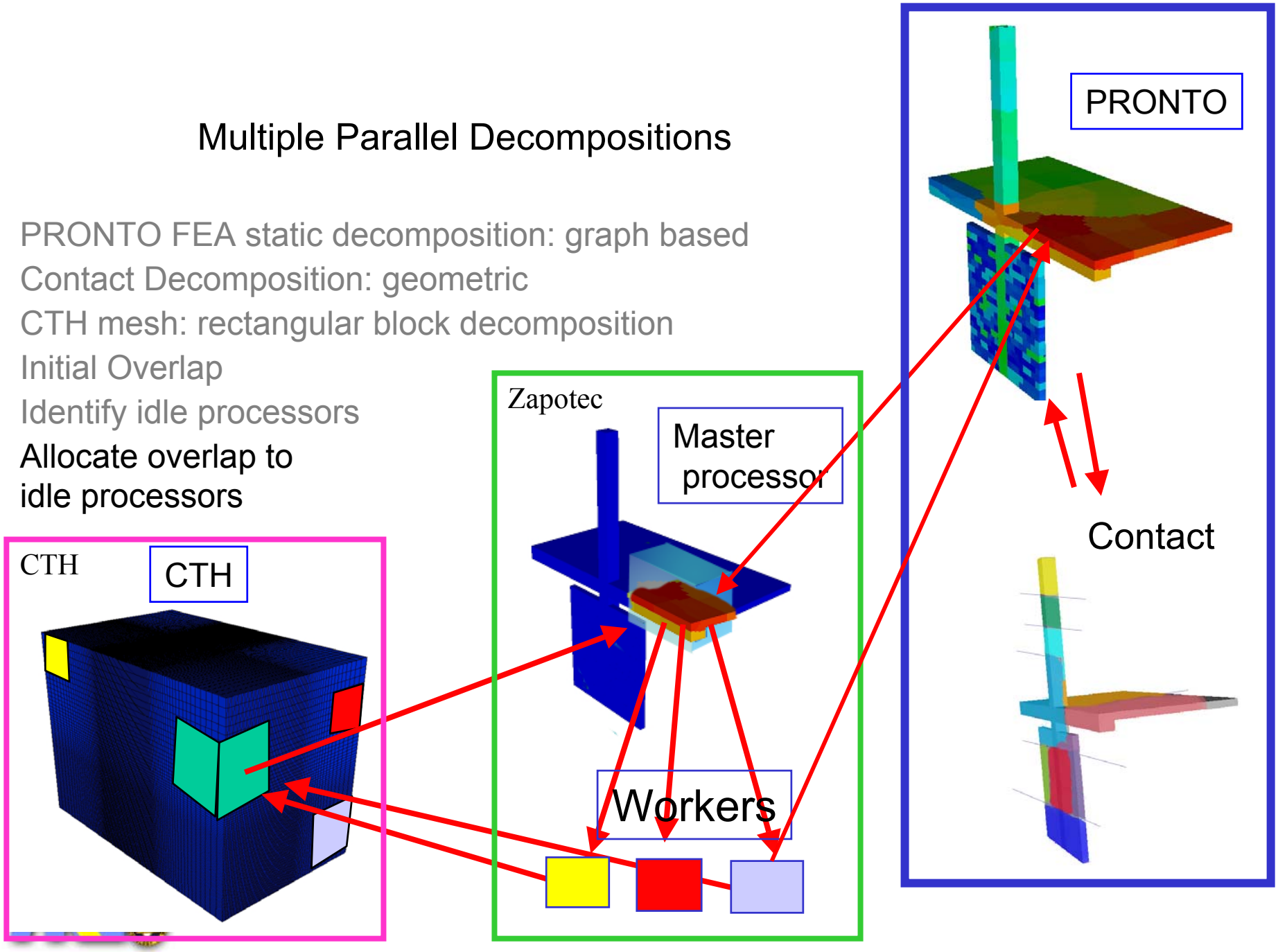
Two Way Coupling of Eulerian and Lagrangian Codes

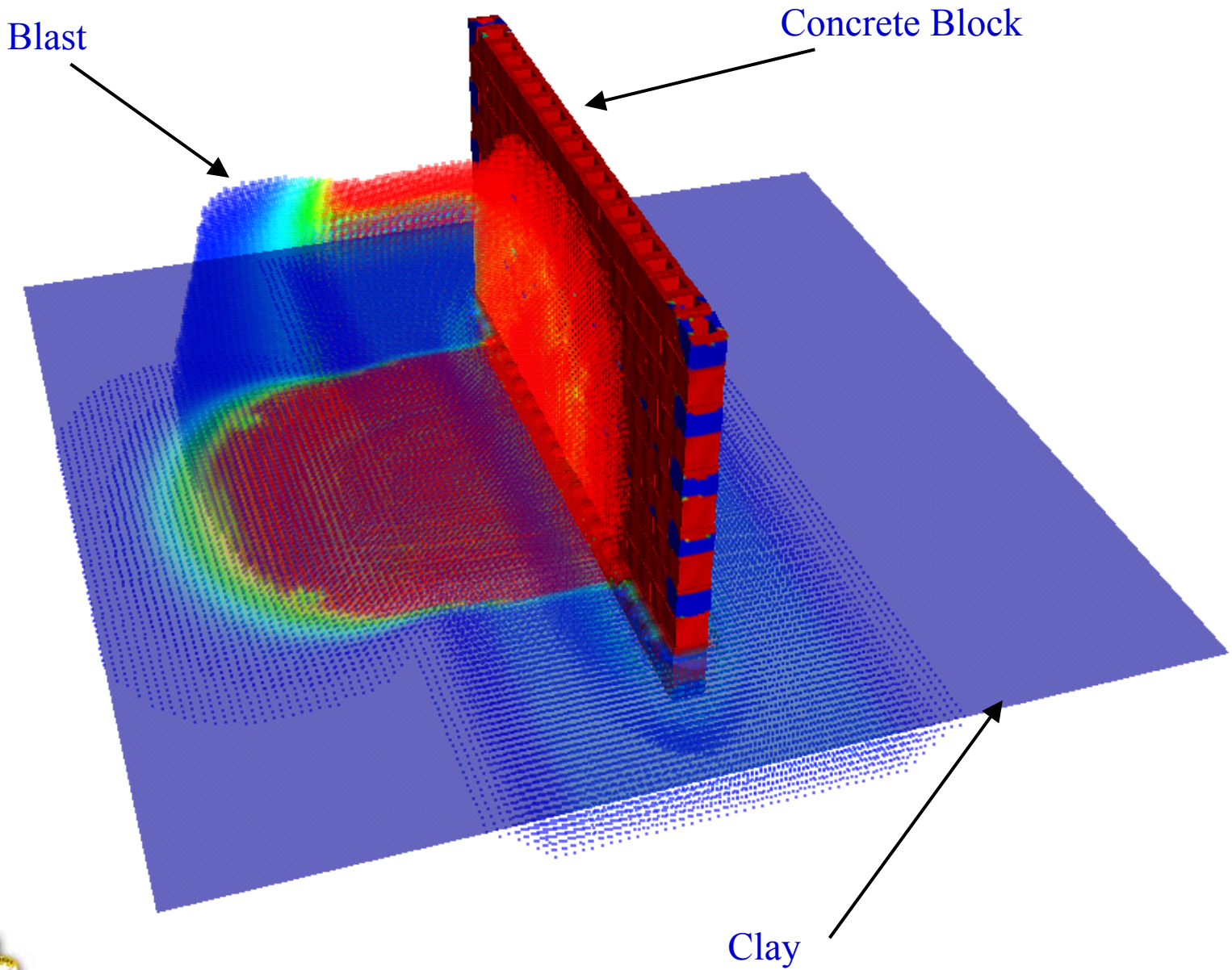


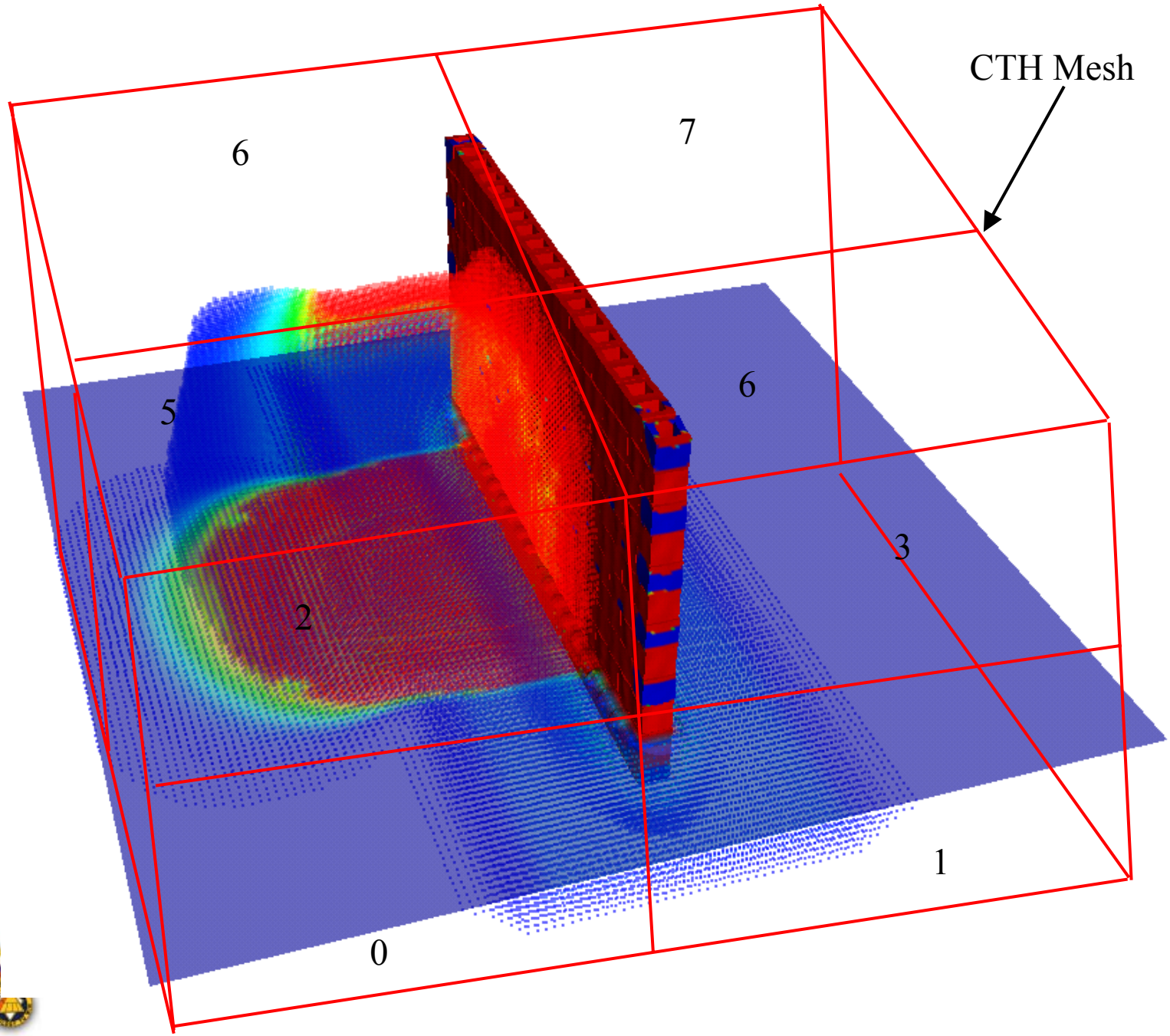


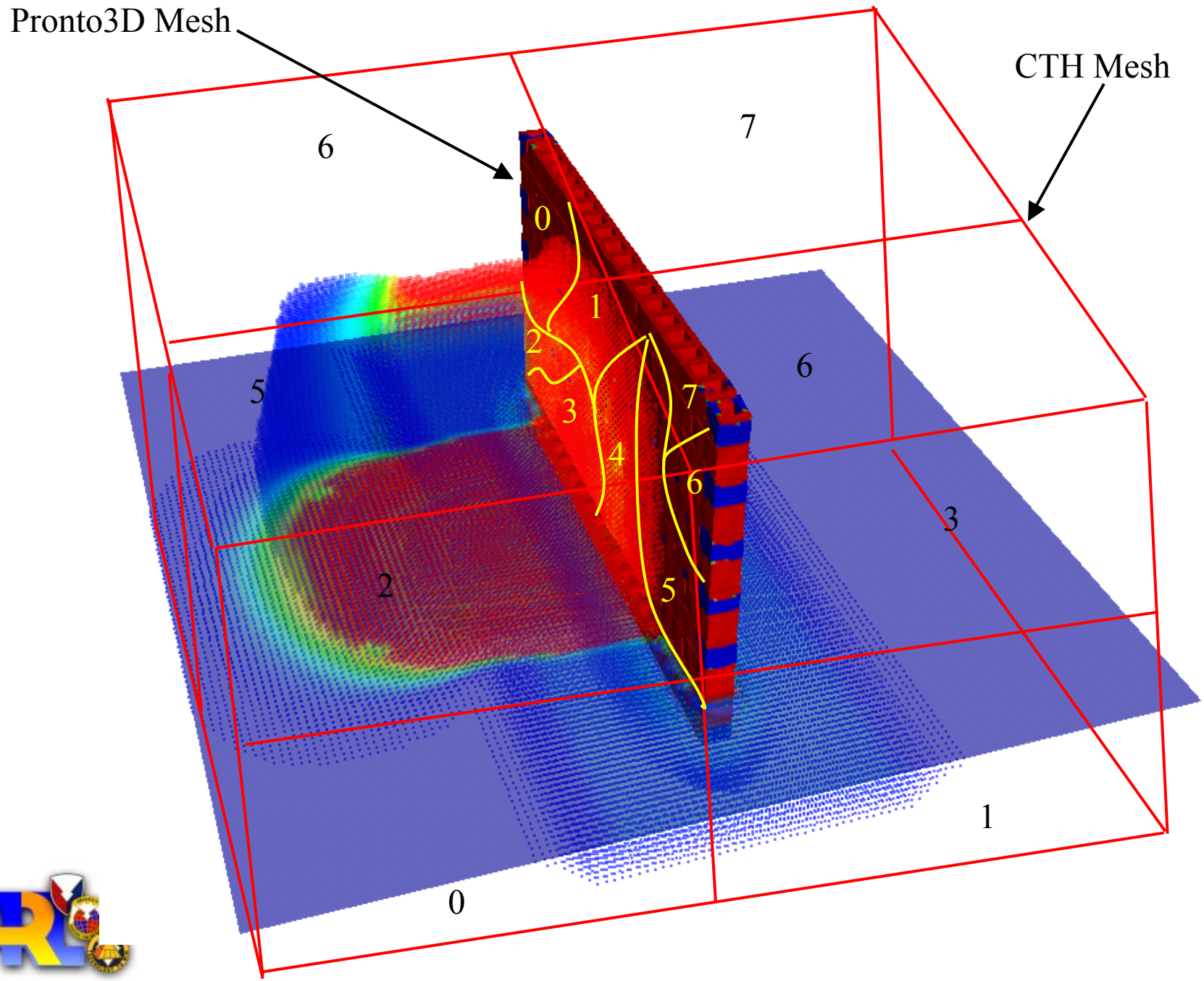
Multiple Parallel Decompositions

PRONTO FEA static decomposition: graph based
Contact Decomposition: geometric
CTH mesh: rectangular block decomposition
Initial Overlap
Identify idle processors
Allocate overlap to idle processors



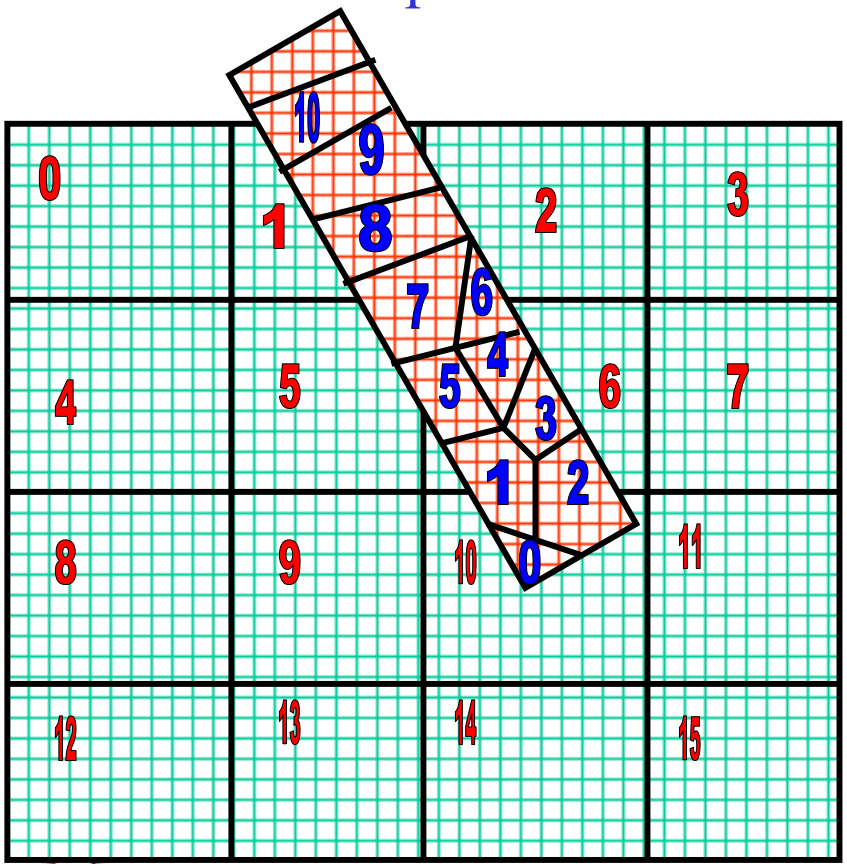

Blast
Concrete Block
Clay


CTH Mesh
6
7
5
6
3
2
1
0


Pronto3D Mesh
CTH Mesh
6
7
0
1
2
3
4
5
6
7
5
6
3
2
1
0

Interactions Between CTH and PRONTO

PRONTO decomposition



CTH decomposition

Processor Interactions

CTH / PRONTO	Total
• 1 - 7, 8, 9, 10	4
• 2 - 6, 7, 8, 9	4
• 5 - 5, 7	2
• 6 - 1, 2, 3, 4, 5, 6, 7	7
• 10 - 0, 1, 2	3

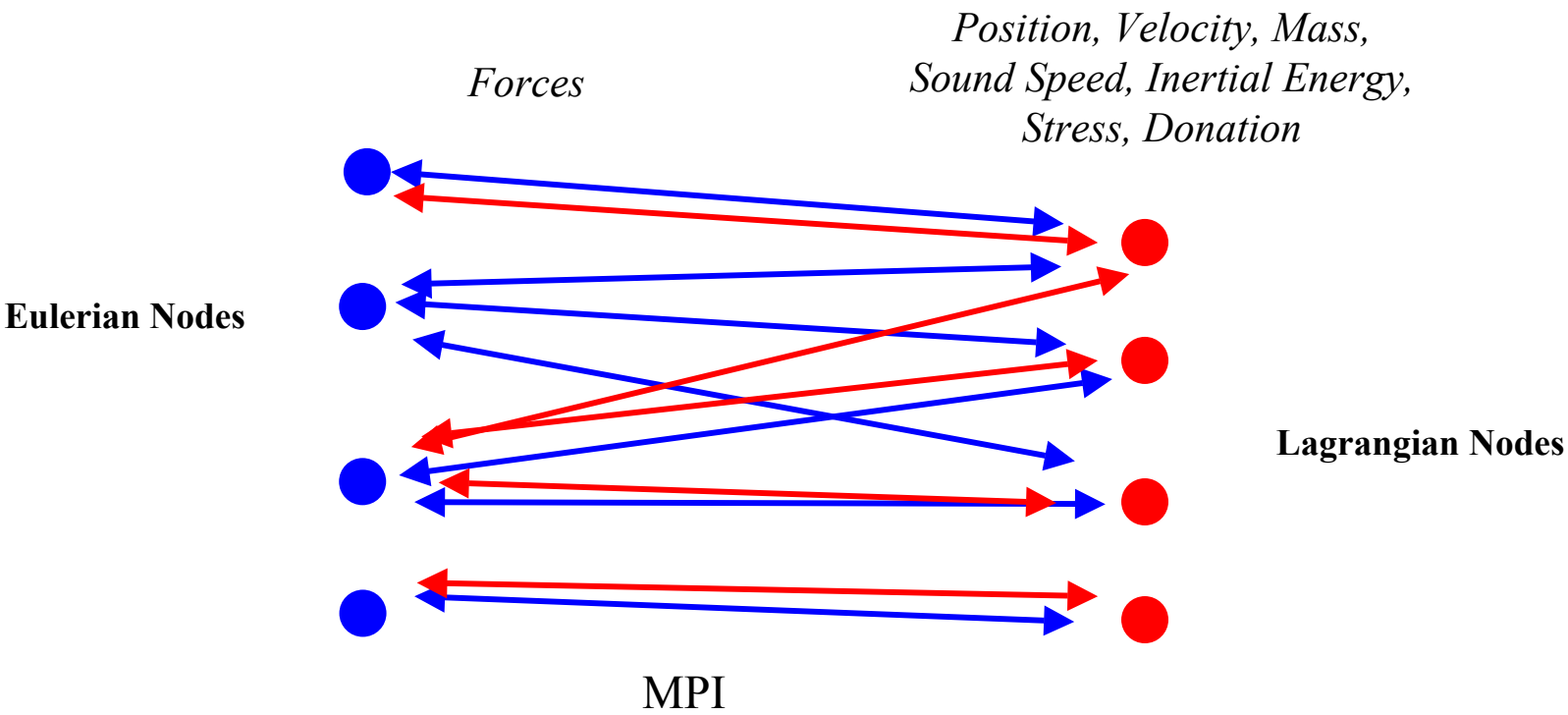
20 regions interact

5 processors with overlaps

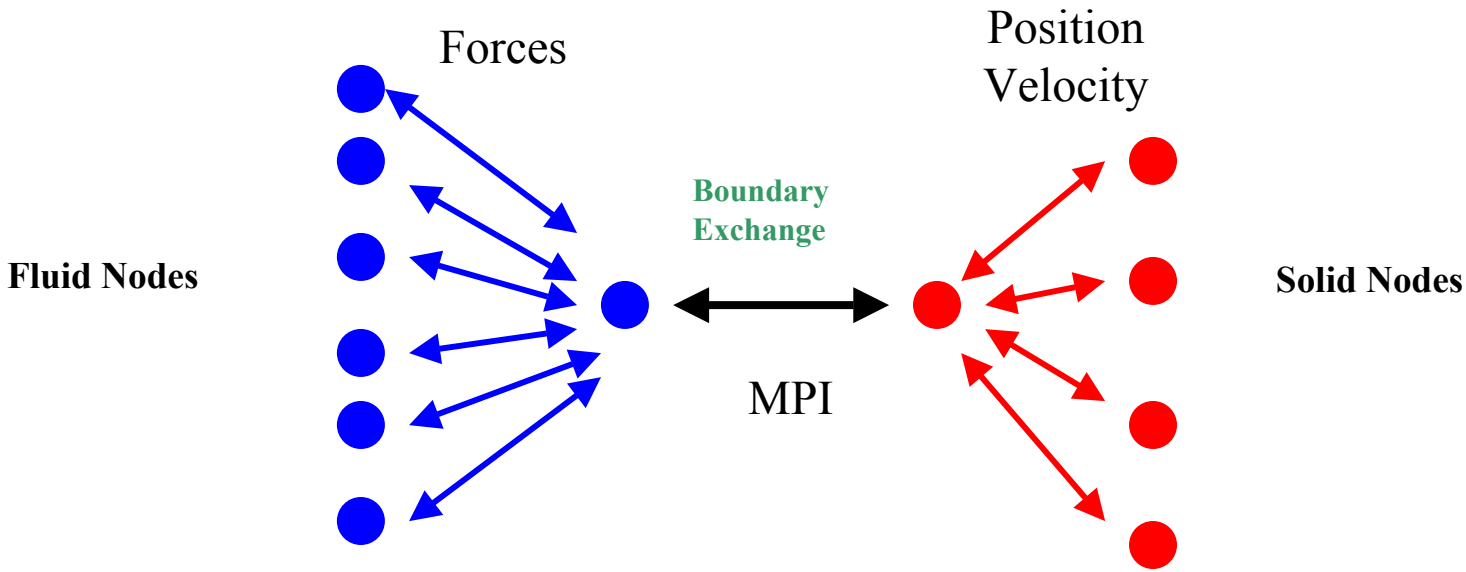
12 processors without overlaps

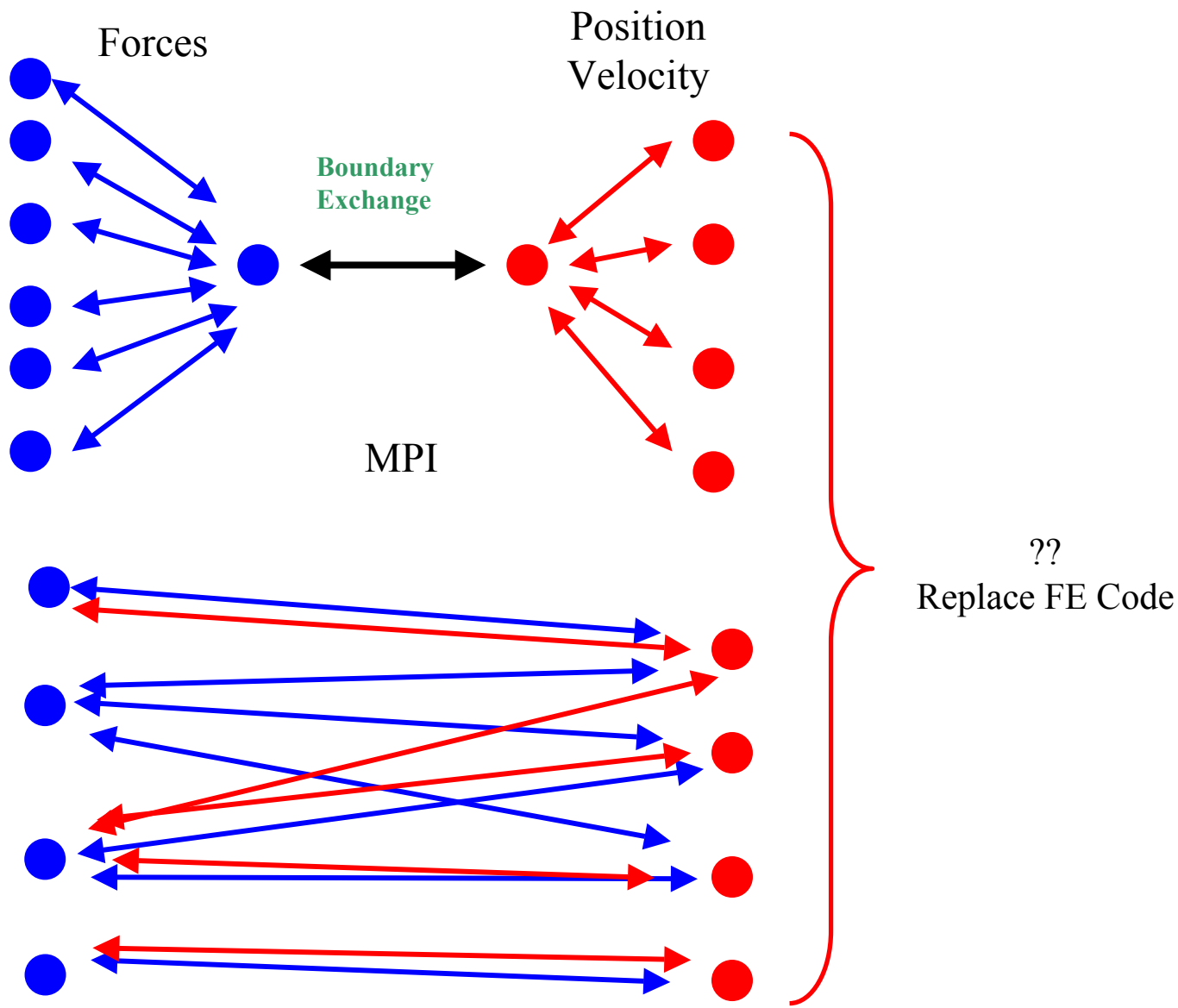
maximum number of overlaps = 7

WTI-1 Exchange



WTI-2 Exchange




Forces
Position
Velocity
Boundary
Exchange
MPI
??
Replace FE Code

Couple Two or More Previously Stand Alone Codes with an Interaction Software Layer

N Processors

```
SolveSystem(…) {
    SolvePhysics1(…);
    InterfaceResult(…);
    SolvePhysics2(…);
    InterfaceResults(…);
    }
```

N Processors

M Processors

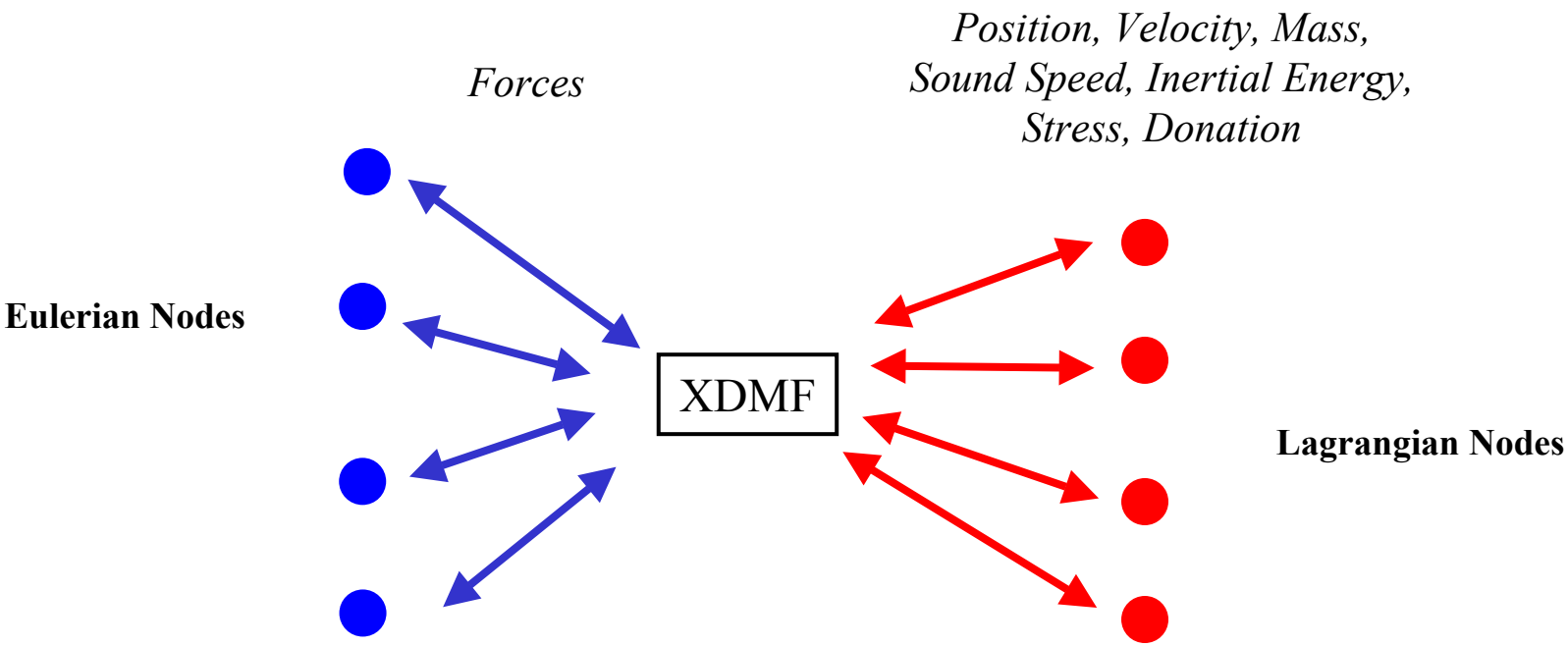
Disk Files
MPI Communicator
Shared Memory

SolvePhysics1(…) ↔ SolvePhysics2(…);

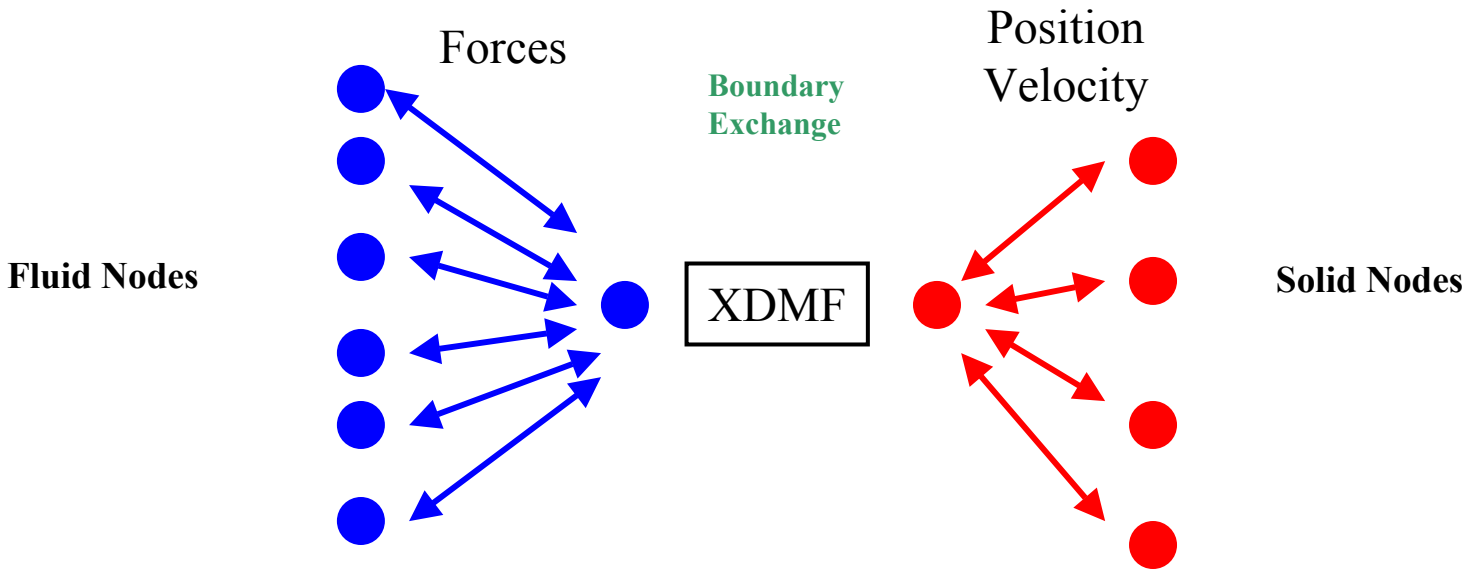
Scalable Active Data Hub
With
Common Data Model and Format

- Leverage Existing / Verified Codes with Minimal Modification
- Allows for Construction of Complex System with Modular Components
- Solves Difficult Domain Decomposition Issues (N to M Mapping)
- Allows for Runtime Analysis (Monitoring, Visualization, etc.)

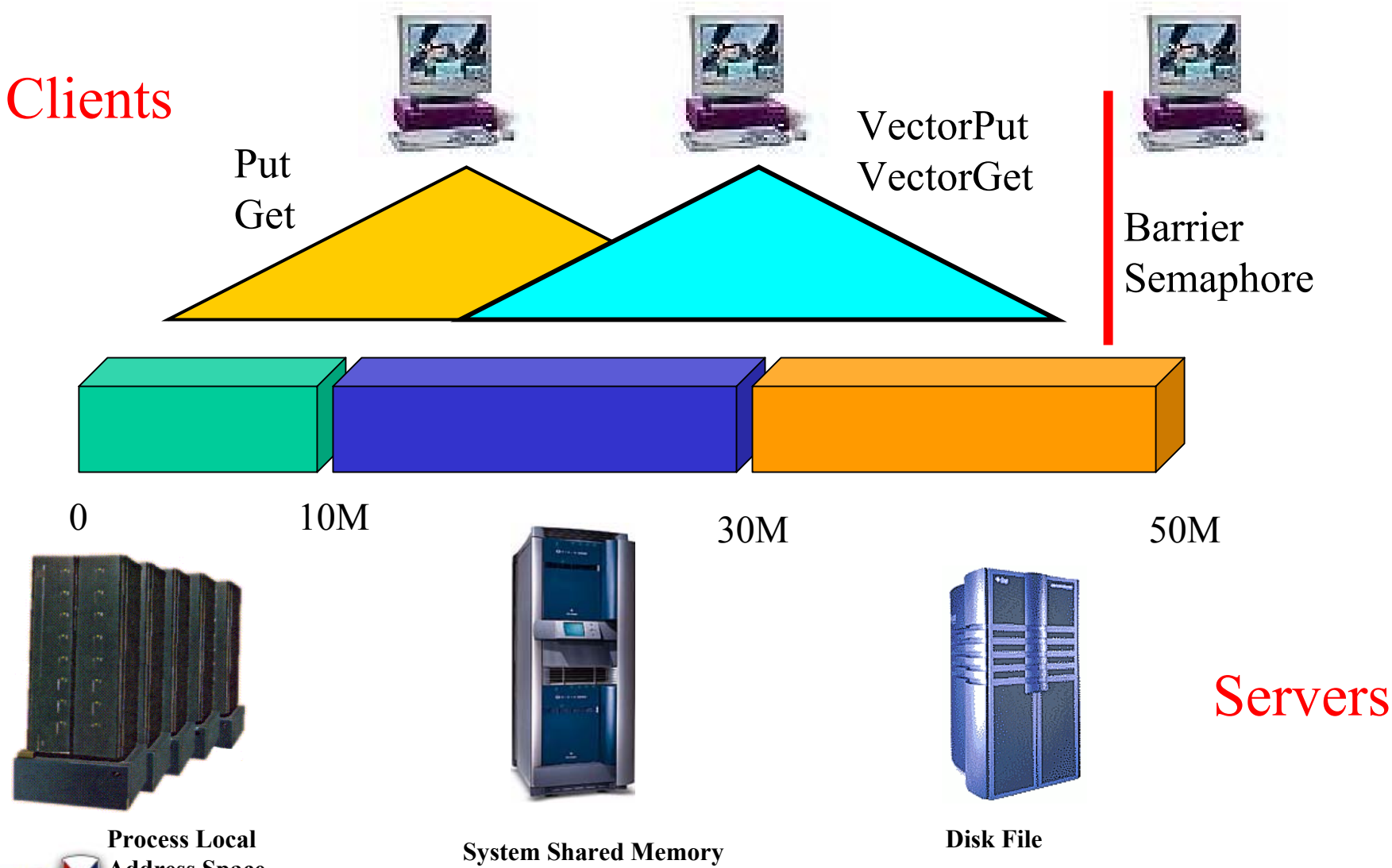
WTI-1 Exchange



WTI-2 Exchange



Network Distributed Global Memory



HDF-EOS

Hierarchical Data Format - Earth Observing System

- Provides the primary data standard for NASA's Earth Science Data
- Libraries and Analysis Tools for UNIX and Windows Platforms

Earth Science Data

Analysis Tools

HDF-EOS API

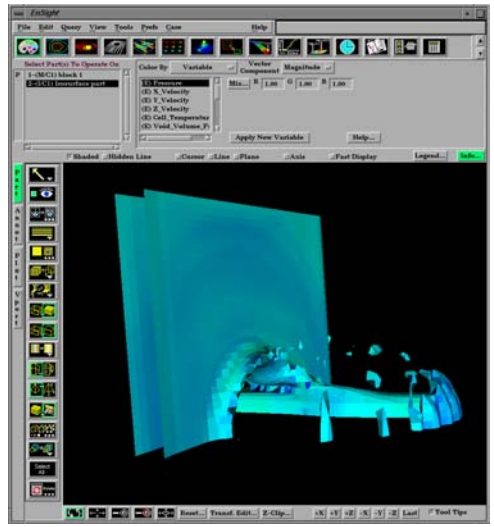
Hierarchical Data Format

Version 5

DOE ASCI Scientific Data Management

HPC Codes

Analysis Tools



Sets and Fields

Fiber Bundle Kernel

Hierarchical Data Format Version 5

NASA HDF-EOS

- Data access is targeted at *Earth Science* data thus inappropriate for many DOD HPC Applications
- All data and metadata accessed via HDF
- Access via custom API ONLY

DOE Scientific Data Management

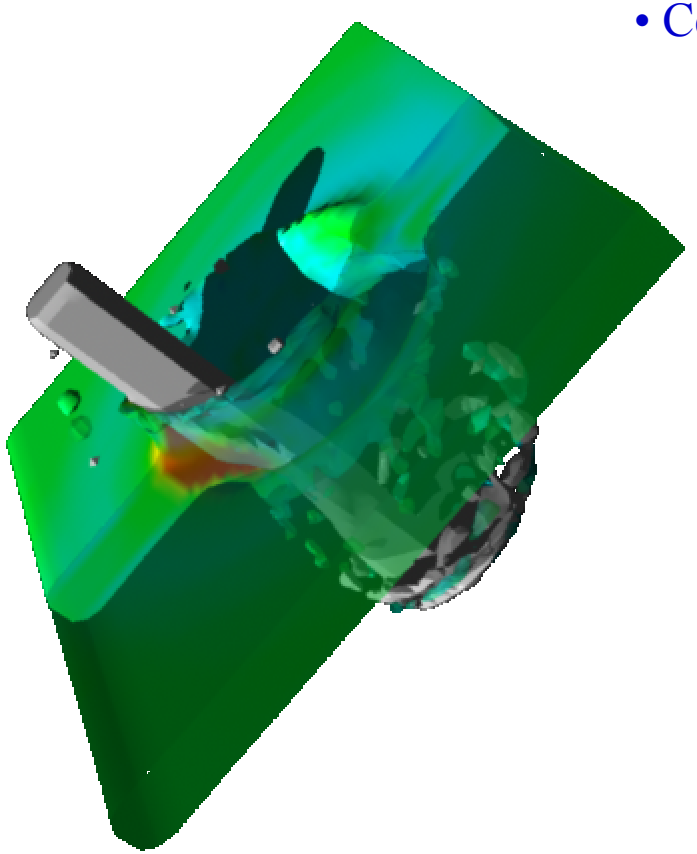
- Targeted at Teraflop scale machines and datasets
- All data and metadata accessed via HDF5
- Access via custom API ONLY
- DOE maintains full control of source and features

Light Data

- 3D Structured Mesh
- 300 x 200 x 300
- Number of Materials
- Material Names
- Total Residual Mass

Heavy Data

- X, Y, Z Values
- Material Volume Fractions
- Cell Temperatures



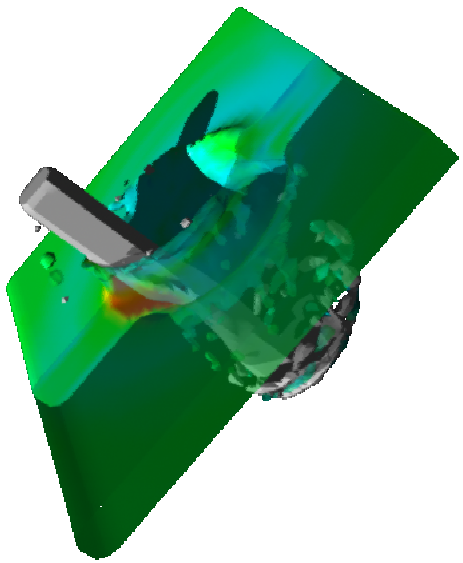
Data Model

Cell Centered Scalars for a
300 x 200 x 300
Structured Mesh

Data Format

299 x 199 x 299
Float32 Values

Light Data



Portion of Larger Dataset in HDF5

Data.h5:/AllScalars

Heavy Data

Shared Memory

The Data

Organization

HPC code Input / Output

Model

Finite Element, Finite Volume, Finite Difference

XYZ Points, Connectivity, Scalars, Vectors, Tensors, ...

Arrays and Structures

Float, Integer, Char, ...

Bytes

Format

Physical Storage of Bits

eXtensible Data Model and Format

In-Memory Application

External Data

API in C++
Wrapped for
FORTRAN, Tcl,
Java, etc.

- Light Data Description
- Heavy Data Formats

eXtensible Data Model and Format

In-Memory Application

XdmfArray
- Rank
- Dimensions
- Number Type
- Selection
  - HyperSlab
  - Coordinates

External Data

- Light Data Description
  - XML
- Heavy Data Formats
  - HDF5
  - Oracle
  - Plot3D
  - etc.

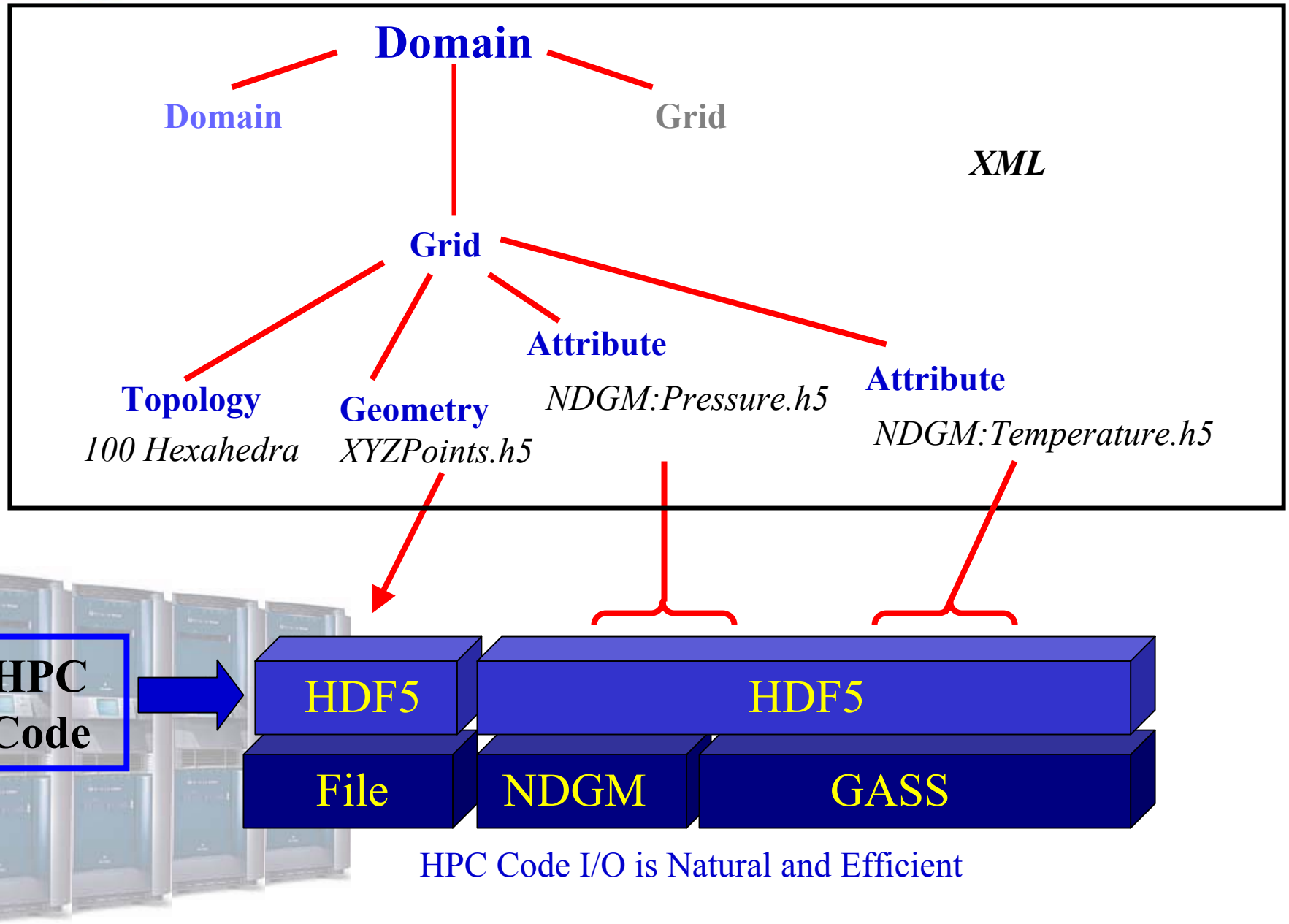
XDMF Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

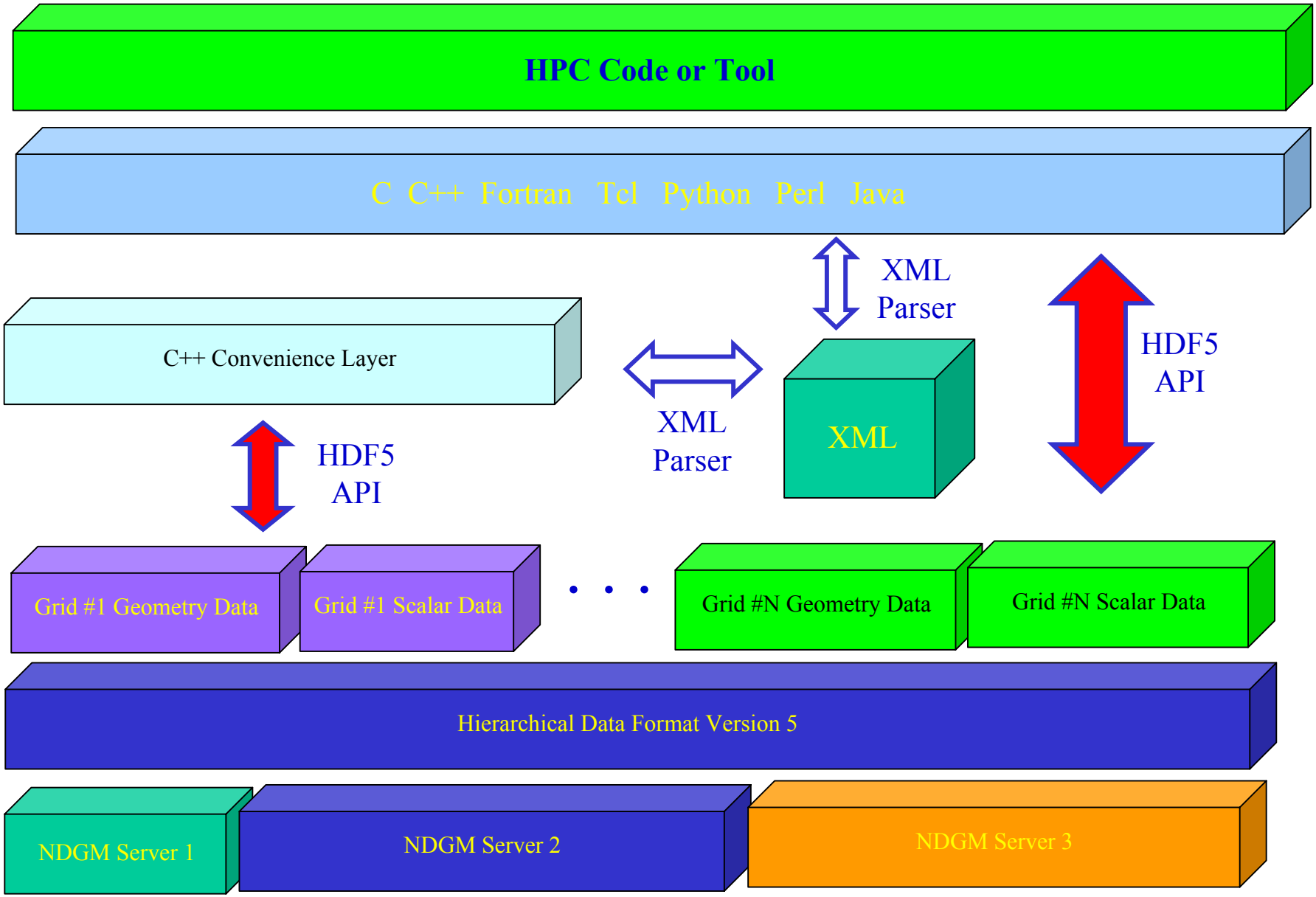
- XdmfObject
  - XdmfDOM
  - XdmfDataDesc
    - XdmfArray
      - XdmfCharArray
    - XdmfHeavyData
      - XdmfHDF
    - XdmfNDGM
      - XdmfRuntime
  - XdmfLightData
    - XdmfAttribute
    - XdmfFormat
      - XdmfFormatHDF
      - XdmfFormatMulti
      - XdmfFormatXML
    - XdmfGeometry
    - XdmfTopology
      - XdmfGrid
    - XdmfTransform
  - XdmfMpi
  - XdmfParameter
  - XdmfXNode

eXtensible Data Model and Format

Complexity of Model Defined in Light Data



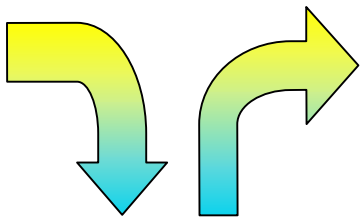
HPC Code I/O is Natural and Efficient


HPC Code or Tool
C C++ Fortran Tcl Python Perl Java
XML Parser
HDF5 API
C++ Convenience Layer
XML Parser
XML
HDF5 API
Grid #1 Geometry Data
Grid #1 Scalar Data
Grid #N Geometry Data
Grid #N Scalar Data
Hierarchical Data Format Version 5
NDGM Server 1
NDGM Server 2
NDGM Server 3

User Application

Scripting (Python, Tcl) and Java Interface

High Performance Computing Code



Analysis
Visualization
XY Plots
Calculated Quantities
etc.

Object Oriented Support for Data Model, XML, ...

Visualization Toolkit (**vtk**)

Hierarchical Data Format (HDF5)
Data Format : Number Type, Data Space ...

OpenGL

Network Distributed Global Memory (NDGM)
with GSSAPI

Mesa

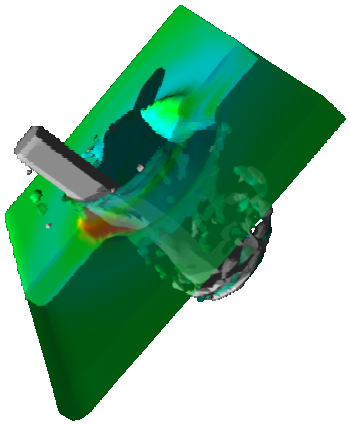
Shared Memory Facility
TCP/IP Sockets

Operating System

File I/O
Process Control

Distributed Resources

Data Classification



By Size

Light Data

- **Descriptions**
- **Names**
- **Rank and Dimensions**
- **Small Computed Values**

Heavy Data

- **Large Collection of Values**

By Function

Data Model

Cell Centered Scalars for a 300 x 200 x 300 Structured Mesh ← Intent

Data Format

Number Shape → 299 x 199 x 299

Number Type → Float32 Values

↓

Selection → Portion of Larger Dataset in HDF5

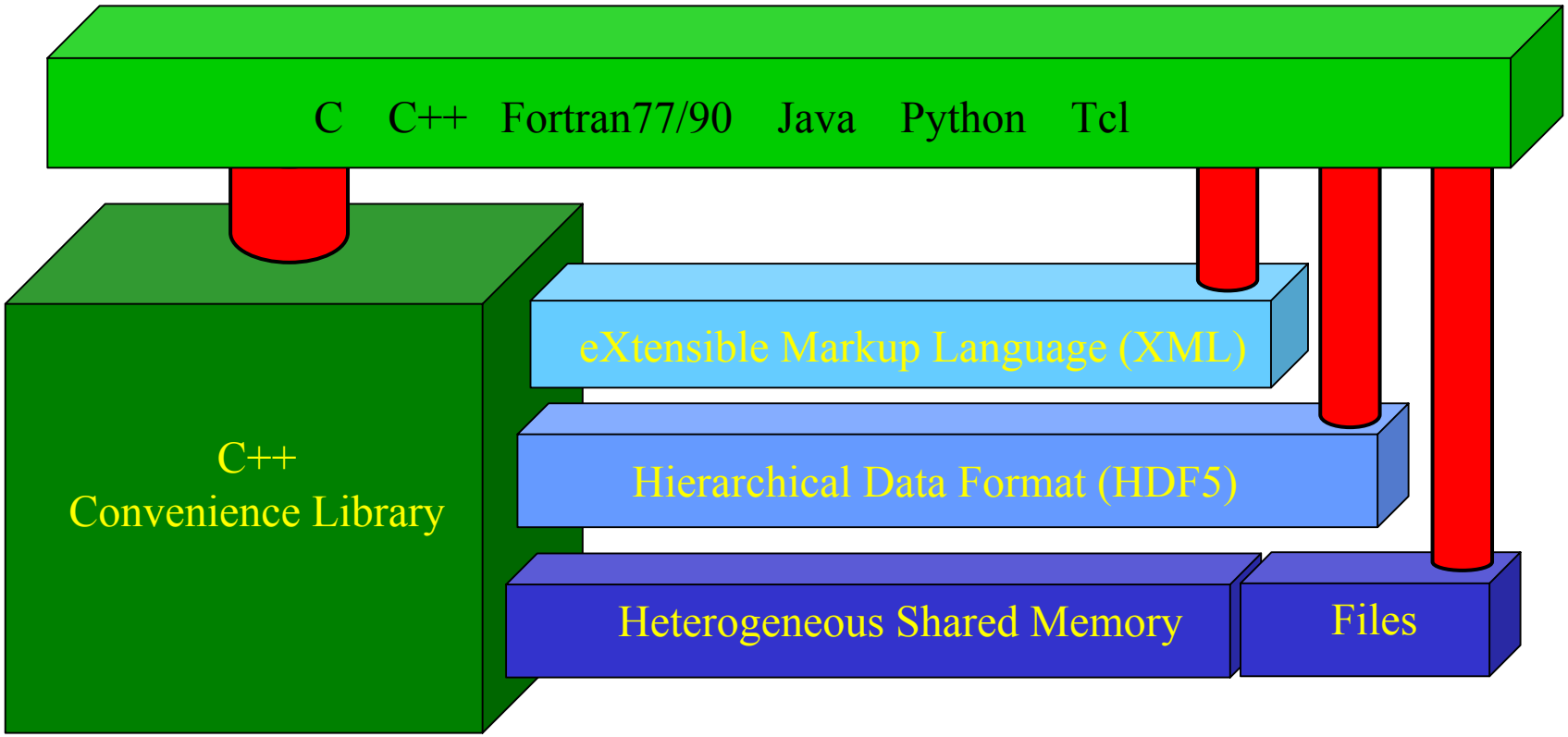
↓

Data.h5:/AllScalars

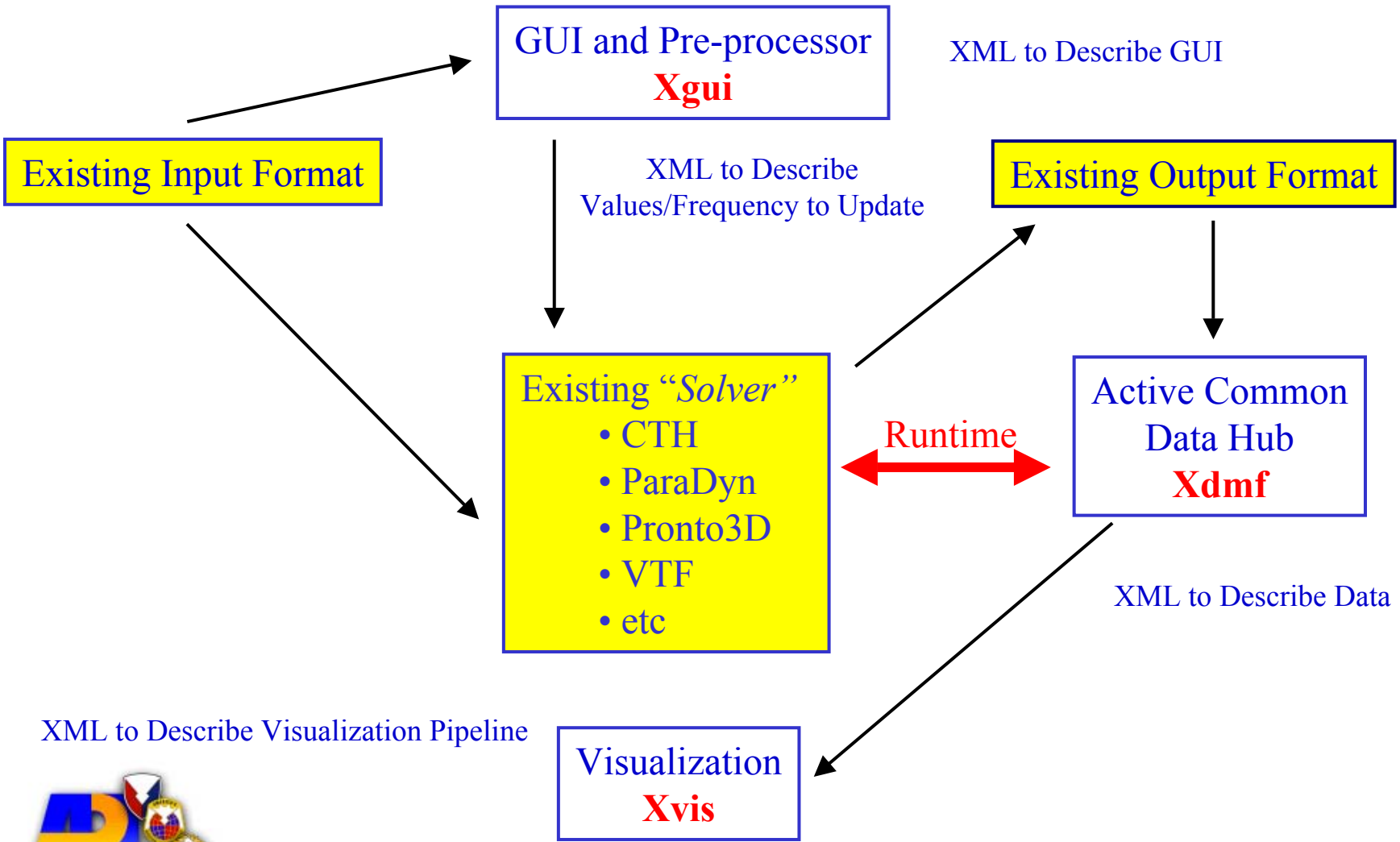
↓

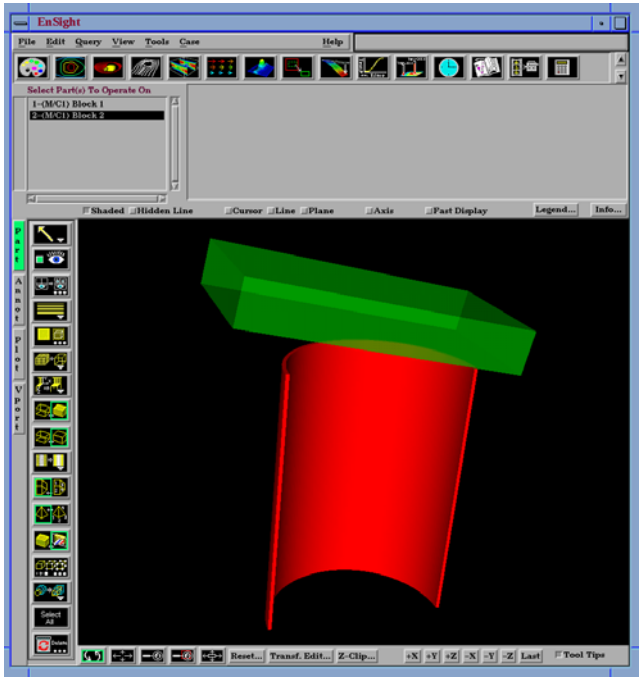
Shared Memory

eXtensible Data Model and Format

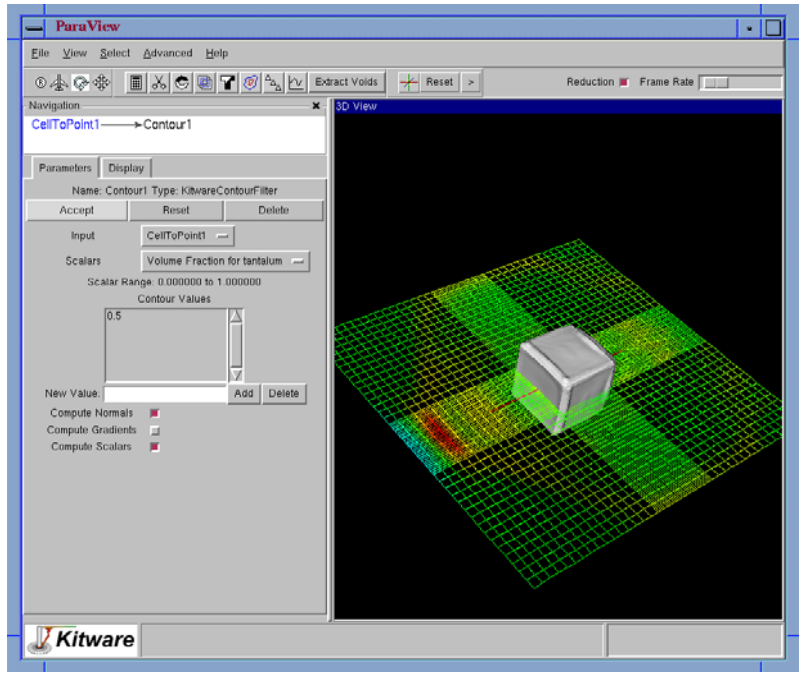


Pervasive use of eXtensible Markup Language (XML)

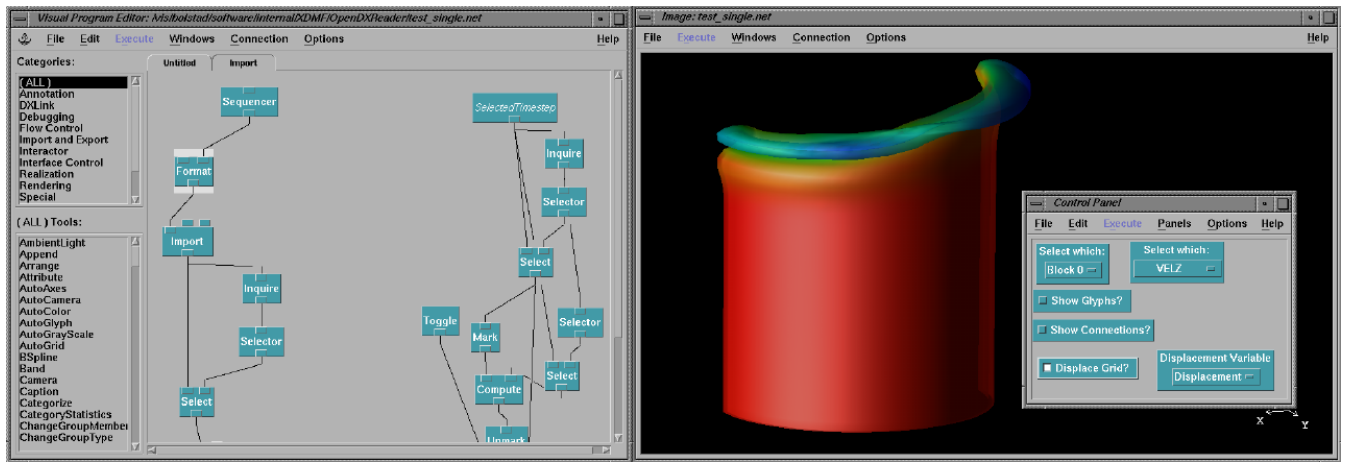




EnSight

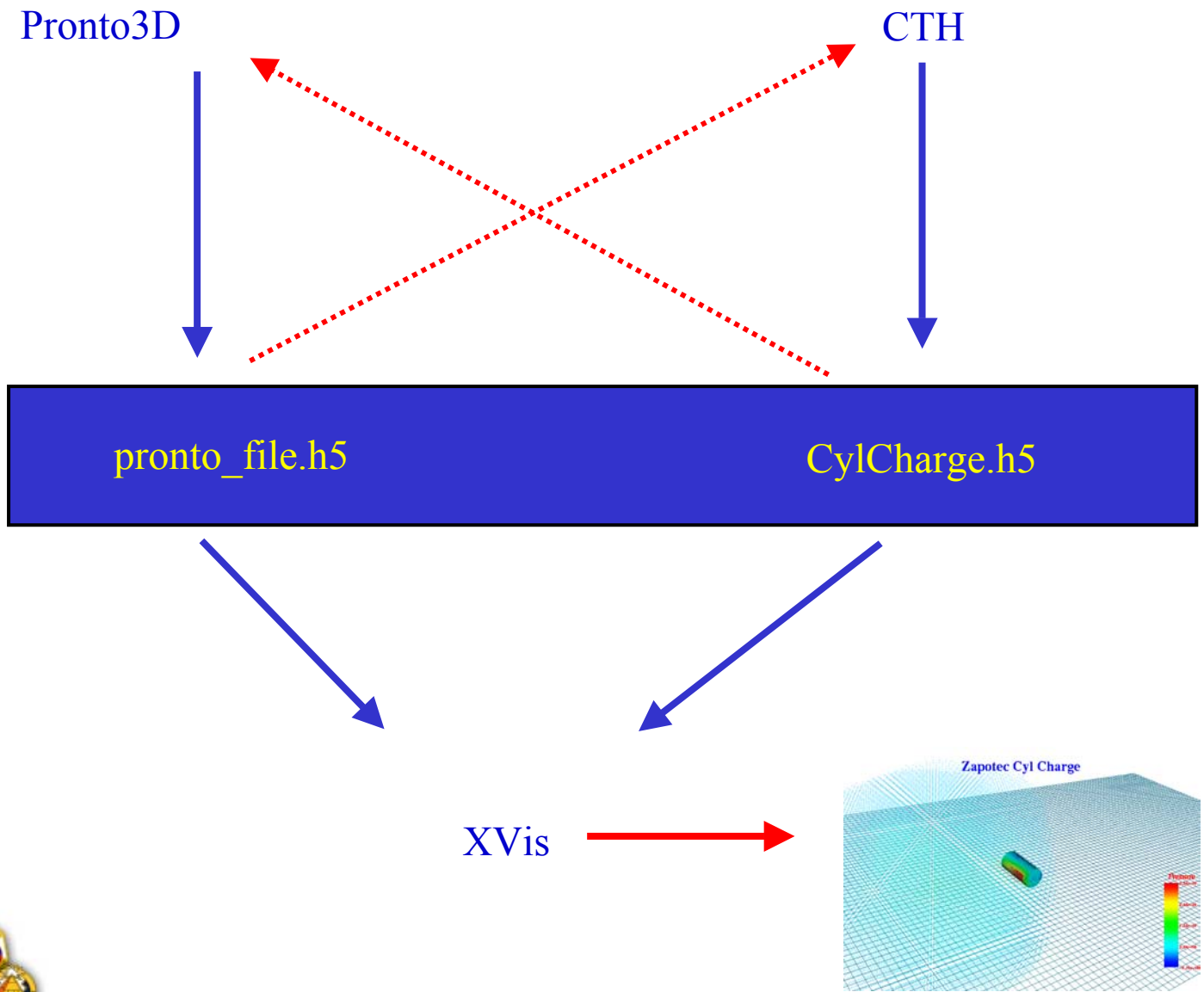


ParaView



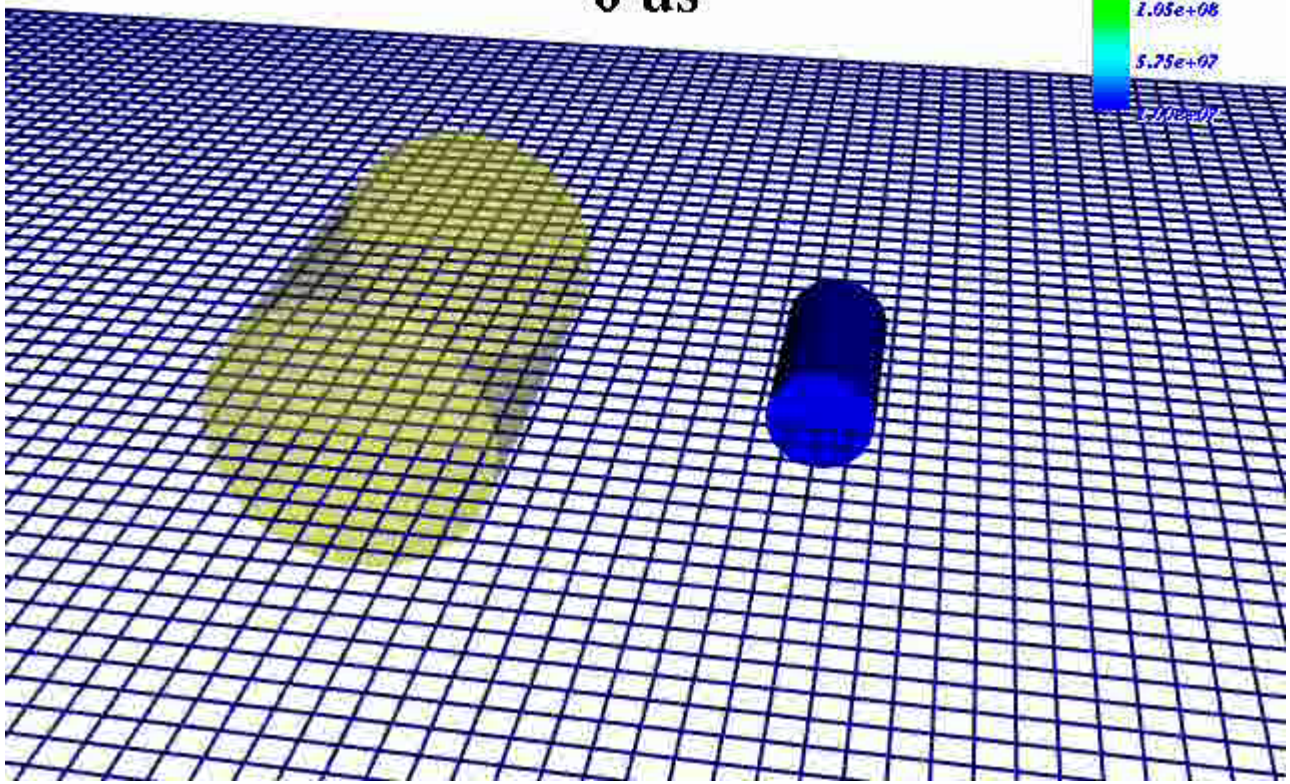
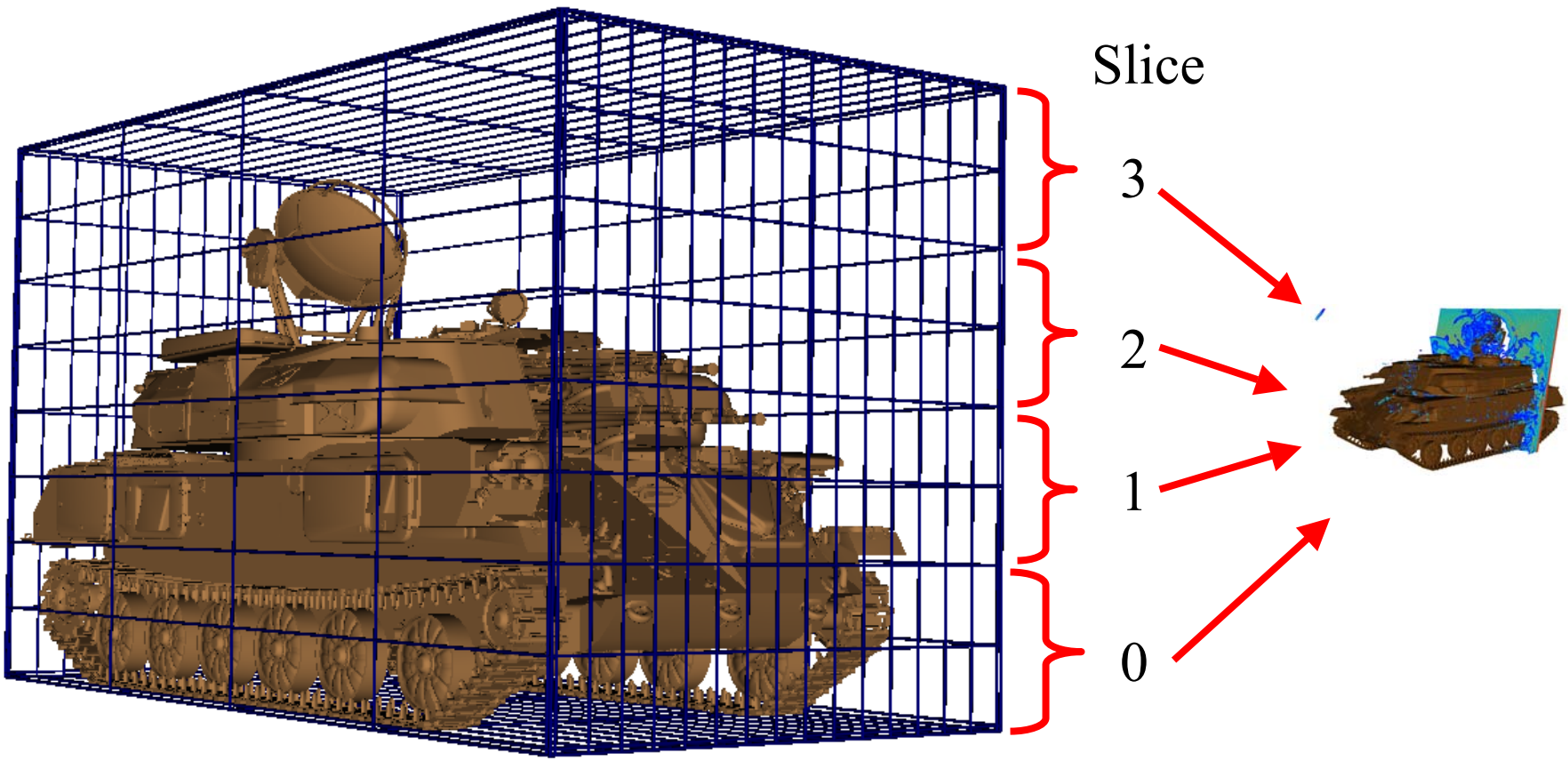
OpenDX

Zapotec


Pronto3D
CTH
pronto_file.h5
CylCharge.h5
XVis
Zapotec Cyl Charge

Cyl Charge vs. Cyl Target

0 us



Repartitioning for Parallel Visualization