Open Source Software to Visualize Complex Data on Remote CEA's Supercomputing Facilities

Release 0.5

Fabien Vivodtzev[1], Thierry Carrard[1]

April 2nd, 2014

[1]CEA (French Alternative Energies and Atomic Energy Commission)

Abstract

In order to guaranty performances of complex systems using numerical simulation, CEA is performing advanced data analysis and scientific visualization with open source software using High Performance Computing (HPC) capability. The diversity of the physics to study produces results of growing complexity in terms of large-scale, high dimensional and multivariate data. Moreover, the HPC approach introduces another layer of complexity by allowing computation amongst thousands of remote cores accessed from sites located hundreds of kilometers away from the computing facility.

This paper presents how CEA deploys and contributes to open source software to enable production class visualization tools in a high performance computing context. Among several open source projects used at CEA, this presentation will focus on Visit, VTK and Paraview.

In the first part we will address specific issues encountered when deploying VisIt and Paraview in a multi-site supercomputing facility for end-users. Several examples will be given on how such tools can be adapted to take advantage of a parallel setting to explore large multi-block dataset or perform remote visualization on material interface reconstructions of billions of cells. Then, the specific challenges faced to deliver Paraview's Catalyst capabilities to end-users will be discussed.

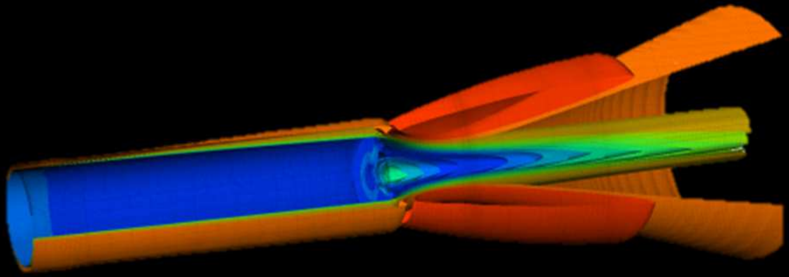
In the second part, we will describe how CEA contributes to open source visualization software and associated software development strategy by emphasizing on two recent development projects. The first is an integrated simulation workbench providing plugins for every step required to achieve numerical simulation independently on a local or a remote computer. Embedded in an Eclipse RCP environment, VTK views allow the users to perform data input using interaction or mesh preview before running the simulation code. Contributions to VTK have been made in order to smoothly integrate these technologies. The second details how recent developments at CEA have helped to visualize and to analyze results from ExaStamp, a parallel molecular dynamics simulation code dealing with molecular systems ranging from a few millions up to a billion atoms. These developments include a GPU intensive rendering method specialized for atoms and specific parallel algorithms to process molecular data sets.

DE LA RECHERCHE À L'INDUSTRIE

cea

www.cea.fr

Open Source Software to Visualize Complex Data on Remote CEA's Supercomputing Facilities



Fabien VIVODTZEV

Thierry CARRARD

Conference EuroVis 2014
Workshop on Open Source in Visualization (OSVIS)
Swansea, UK - June 2014

Overview

- Deploy open source visualization software on CEA facilities
  - Architecture, development and configuration of VisIt
  - Paraview Catalyst

- Contribute to open source visualization software
  - Evolution of VTK in order to integrate interactive views in a RCP Eclipse application
  - Improvement of VTK in order to visualize large molecular dynamic simulations

CONTEXT

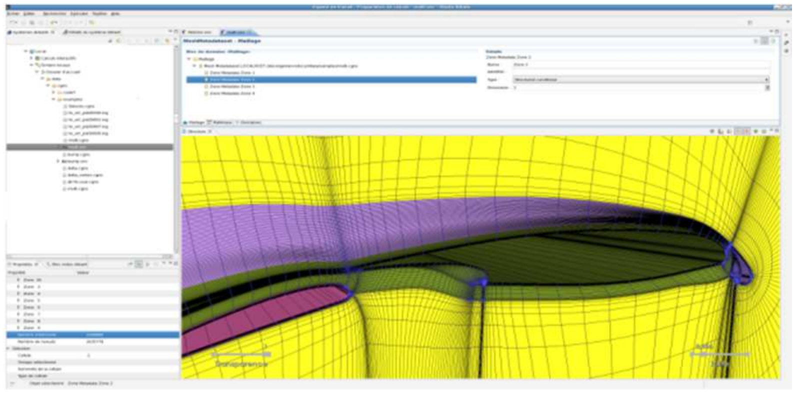
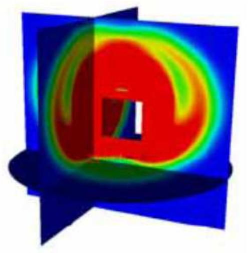
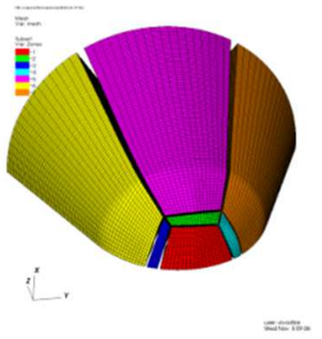
From simulation to the understanding of physical phenomena

The simulation at CEA

- In a context of guarantee of the performances of a complex system using simulation
- Many physics to study : mechanics, aerodynamics, electromagnetism …

Goals in data analysis and visualization

- Enhance understanding of physical phenomena to study
- Explore large volume of data (computed or measured) by reducing complexity
- Setting a software environment adapted, simple to use and homogeneous
- Capitalize solutions in a context of many projects

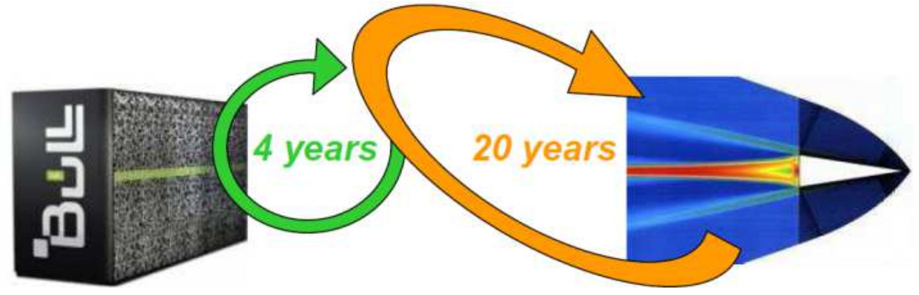
Challenges in data exploration

Complexity of the data

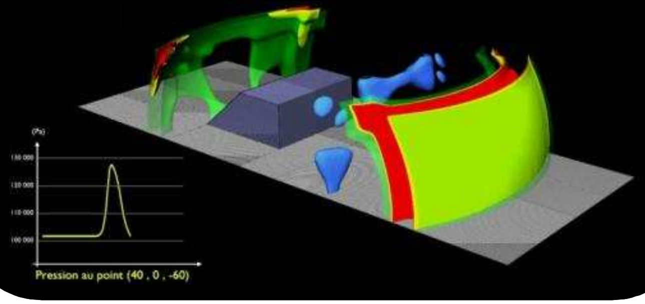
- Origin : HPC, Exascale, many physics, multiscale, many users
- Size : Billions of irregular cells, multi blocks, particles …
- Dimension : 3D time varying, tensors…
- Number : multivariate data, parametric studies,

Constrained environment

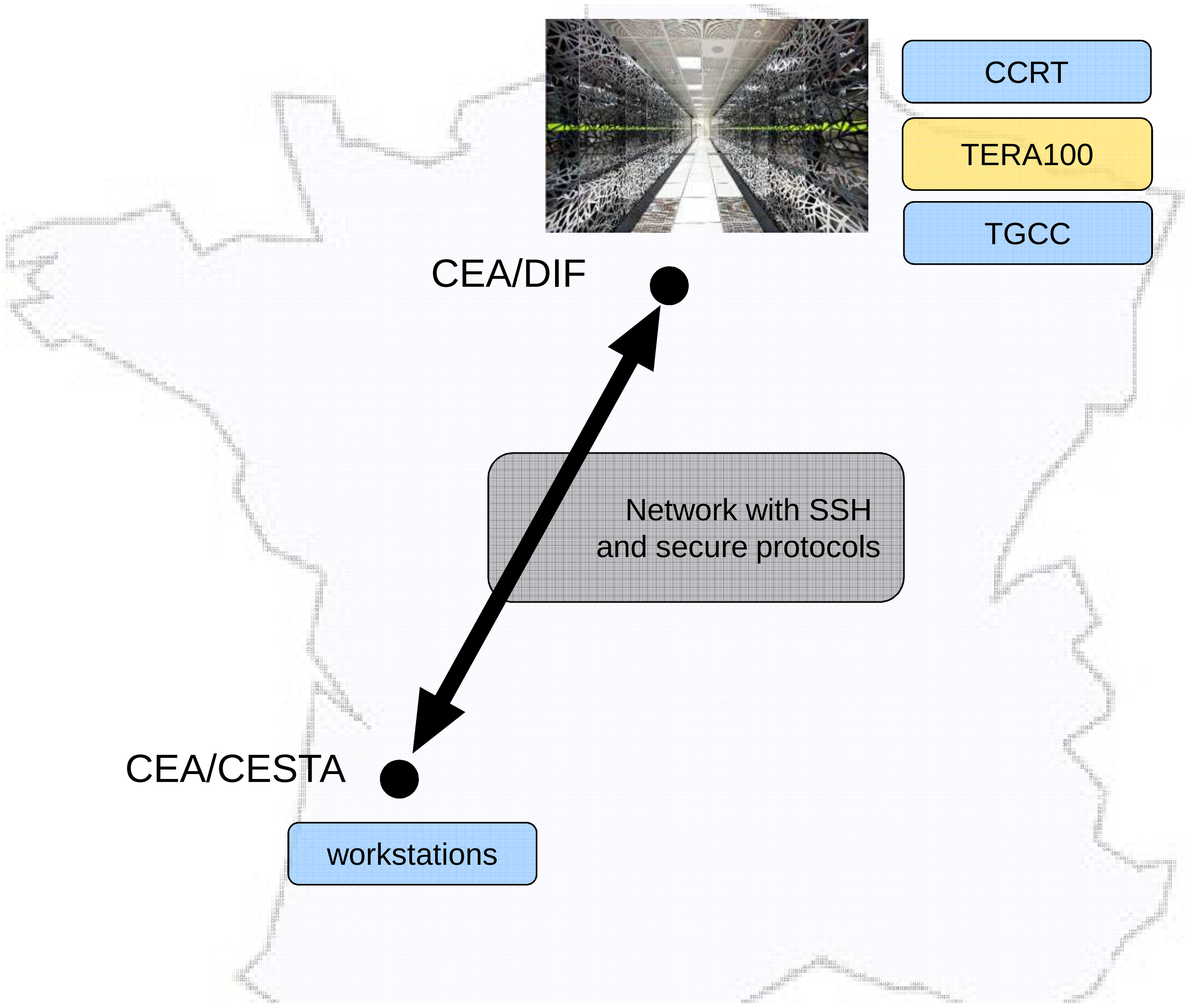
- Petaflop supercomputers : TERA100, TGCC, CCRT
- Remote and secure machine
- Heterogeneous systems (Linux, Windows)
- Non equal software/hardware lifecycle



Need VIS tools !



CEA Facilities at CEA/DIF and CEA/CESTA



DEPLOY OPEN SOURCE VISUALIZATION SOFTWARE ON CEA FACILITIES

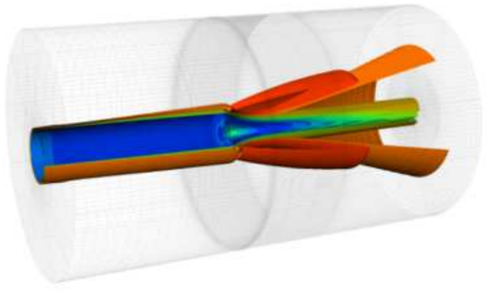
-

VISIT

VisIt at CEA

Open source visualization software

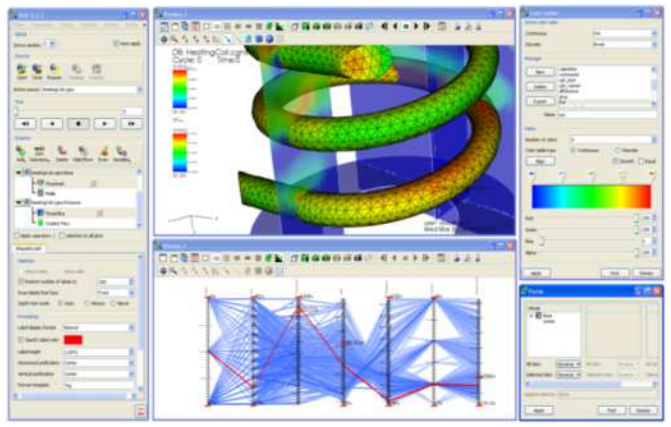
- About 50 users at CEA/CESTA
- First deployment in 2007 … 7 years of experience !
- Efficient for 3D complex data computed or measured
- Parallel and remote visualization amongst CEA sites
- Several layers of adaptation : reader, plot, macro …



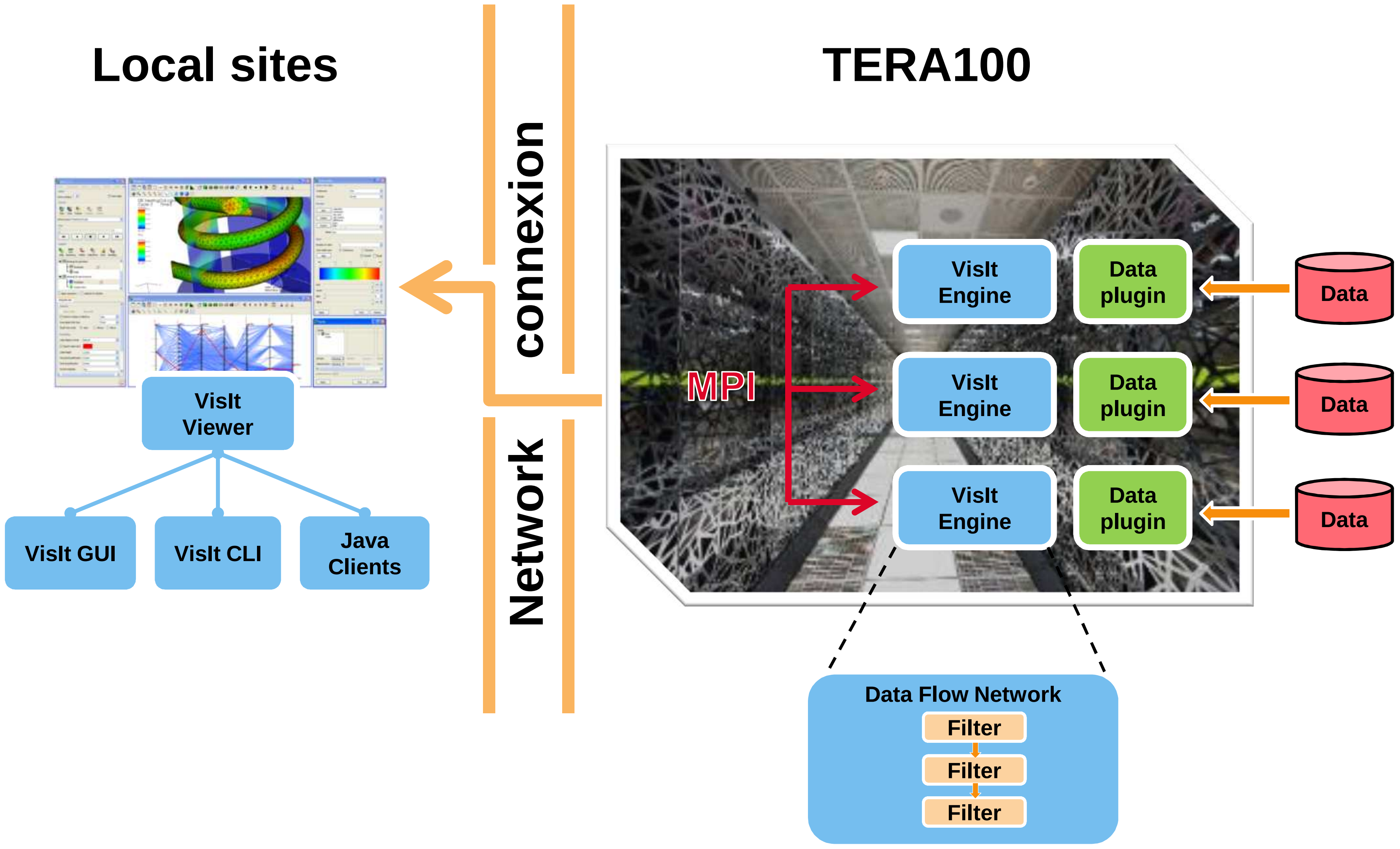
VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data
H.Childs, E.Brugger†, B.Whitlock, J.Meredith, S. Ahern, K.Bonnell, M.Miller†, G.Weber, C.Harrison, D.Pugmire, T.Fogal, C. Garth, A.Sanderson, E.Wes Bethel, M.Durant, D. Camp, J.Favre, O.Rubel, P. Navratil, M. Wheelera, P. Selbya et F.Vivodtzev
Proceedings of SciDac 2011 http://press.mcs.anl.gov/scidac2011 - 2011

Several strategies at CEA

- Direct visualization of simulation data
- Evolution of the simulation code data model (e.g. write SILO files)
- Development of conversion tools
- Development of specific database readers into VisIt
- Development of specific plot

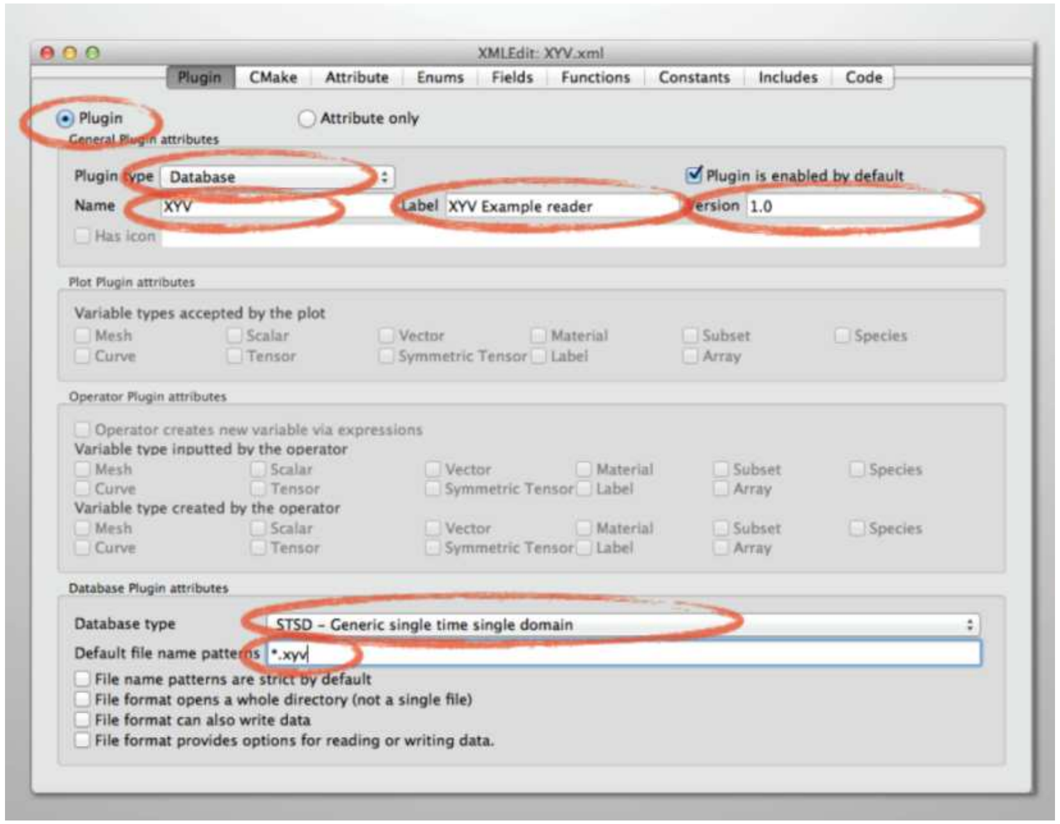


VisIt parallelized client-server architecture



Database reader

- About 12 specific database reader in production at CEA
  - → using VisIt code development tools (xmlEdit)
  - → deployed on both local and remote machine
  - → closely related to the simulation data
  - → computation on demand of derived data


XMLEdit: XYV.xml

Plugin | CMake | Attribute | Enums | Fields | Functions | Constants | Includes | Code

[✓ Plugin] Attribute only

General Plugin attributes

Plugin type: Database
[✓] Plugin is enabled by default
Name: XYV
Label: XYV Example reader
Version: 1.0
[ ] Has icon

Plot Plugin attributes

Variable types accepted by the plot
[ ] Mesh [ ] Scalar [ ] Vector [ ] Material [ ] Subset [ ] Species
[ ] Curve [ ] Tensor [ ] Symmetric Tensor [ ] Label [ ] Array

Operator Plugin attributes

[ ] Operator creates new variable via expressions
Variable type inputted by the operator
[ ] Mesh [ ] Scalar [ ] Vector [ ] Material [ ] Subset [ ] Species
[ ] Curve [ ] Tensor [ ] Symmetric Tensor [ ] Label [ ] Array
Variable type created by the operator
[ ] Mesh [ ] Scalar [ ] Vector [ ] Material [ ] Subset [ ] Species
[ ] Curve [ ] Tensor [ ] Symmetric Tensor [ ] Label [ ] Array

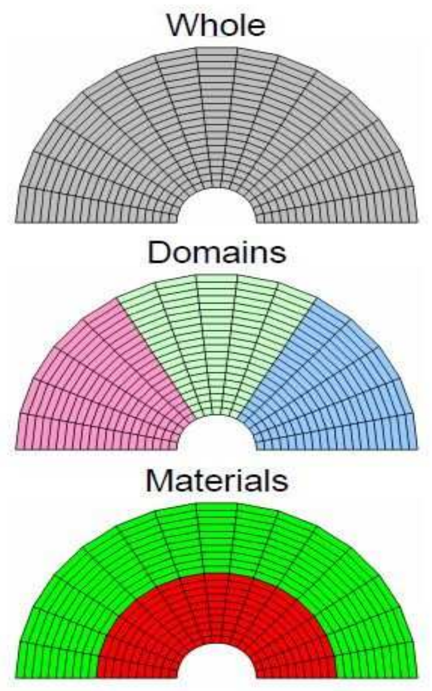
Database Plugin attributes

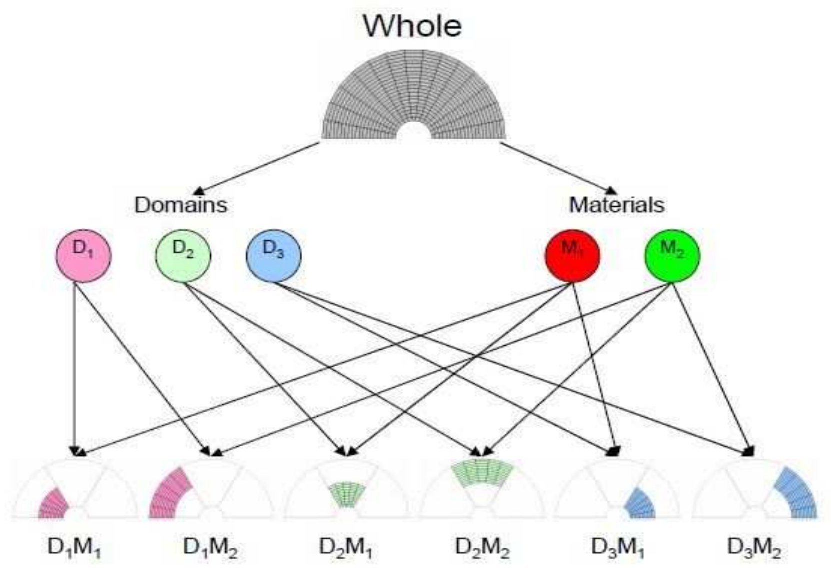
Database type: STSD – Generic single time single domain
Default file name patterns: *.xyv
[ ] File name patterns are strict by default
[ ] File format opens a whole directory (not a single file)
[ ] File format can also write data
[ ] File format provides options for reading or writing data.

Database reader : lesson learned

Best practices

- Think hard to implement a light *PopulateMetaData*
- Use STMD as much as possible to benefit from the native parallelism
- For files without description store positions such as connectivity
- Mesh type : using NetCDF library (getNcDim, nc_inq_varid …)
    - Have to know the structure of the file
- Handling of the materials with the *GetAuxiliaryData*



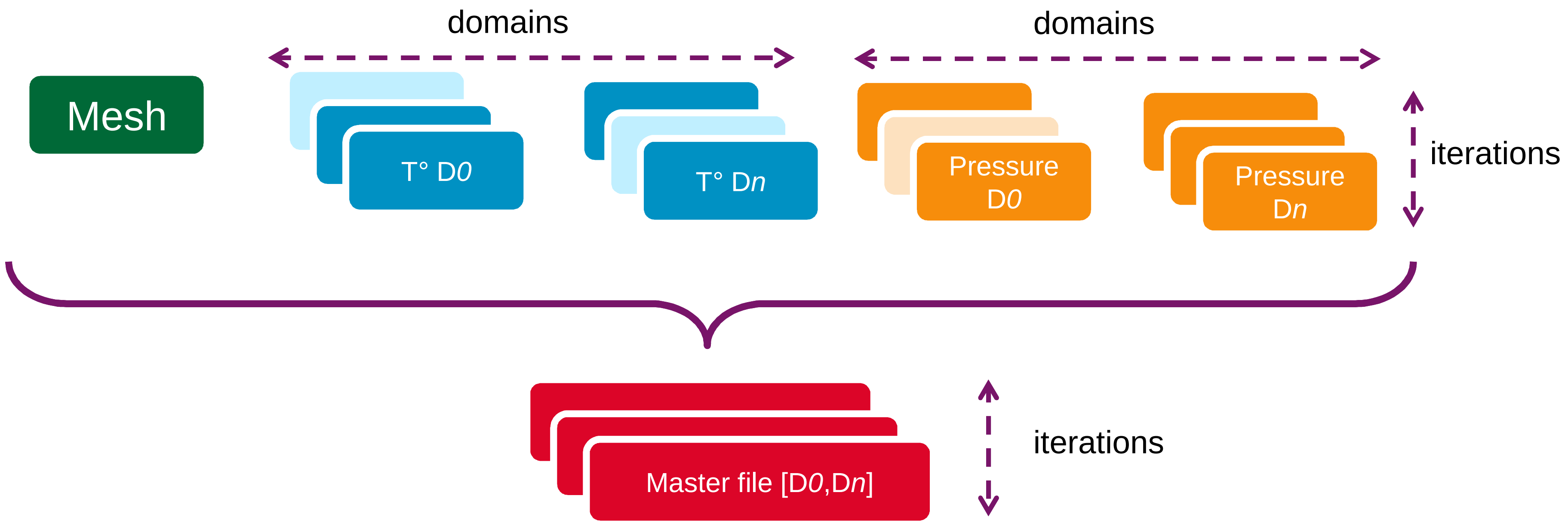


Creating compatible files (SILO)

SILO API

- Lots of mesh types (cells, particles …)
- Multiblocks, empty domains expressed in master files
  - Can be used to handle processor crashes
- Data expressions for derived scalars

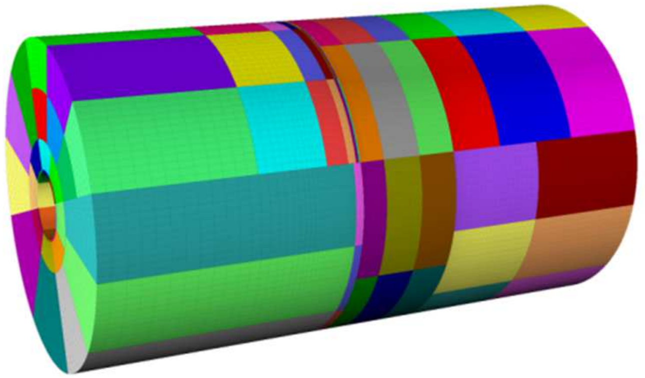
```
/* Processors 3,4 did not contribute so use EMPTY. */
char *meshnames[] = {"proc-1/file000/mesh", "proc-2/file000/mesh","EMPTY", "EMPTY"};
/* Write the multimesh. */
DBPutMultimesh(dbfile, "mesh", nmesh, meshnames, meshtypes, NULL);
```

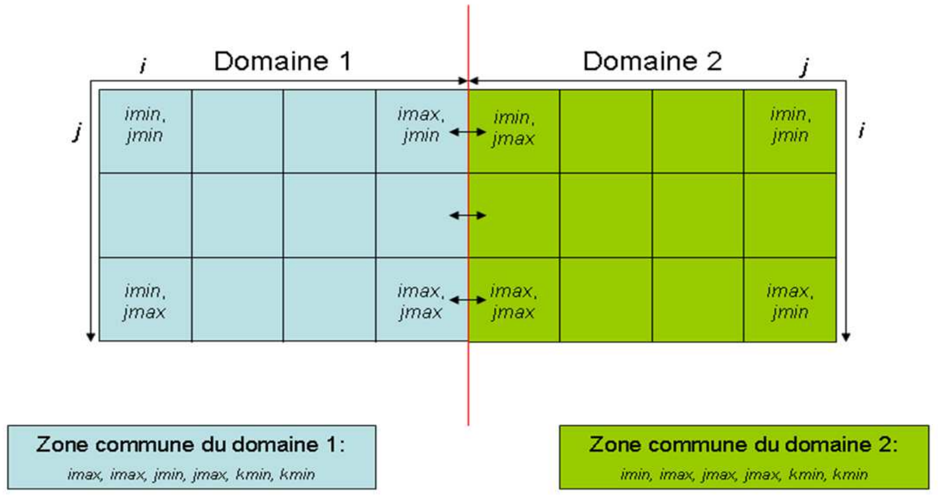


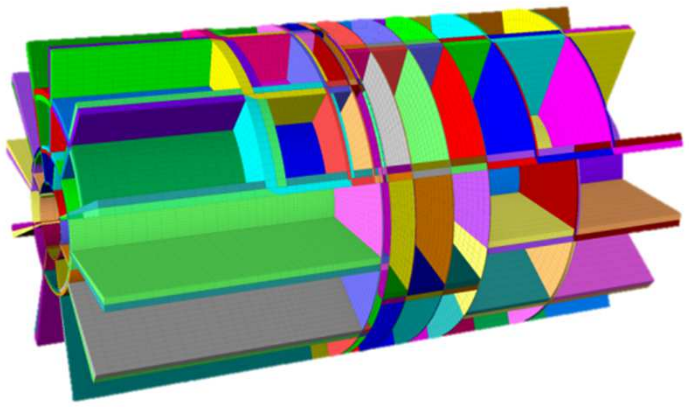
Handling ghost zones

- Using knowledge from the simulation code to generate the ghost cells
  - Block connectivity in the input mesh

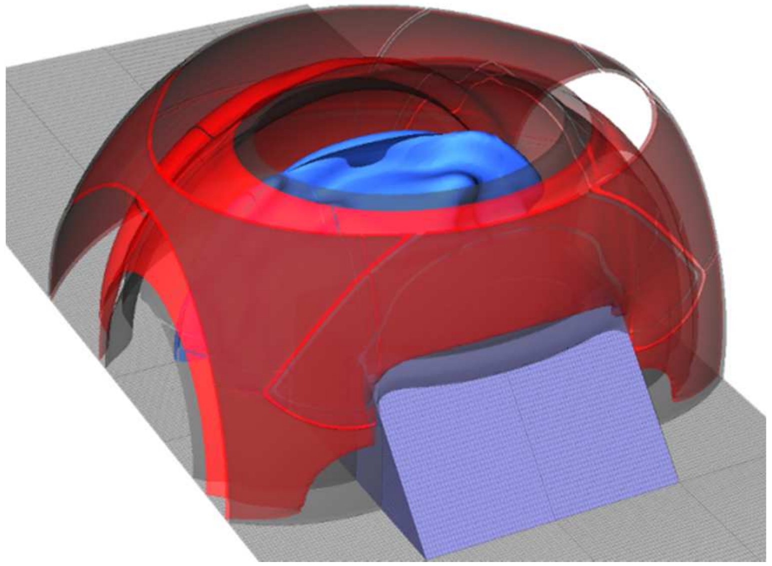
- Data to use
  - Dimensions of the current bloc
  - Min and max indices for each axis
  - Orientation of the axis
  - Zone of interval with the neighboring blocks
  - Example code used :
    - *avtCurvilinearDomainBoundaries*
    - *AddNeighbor (ID, neighborID, neighboringIndices)*

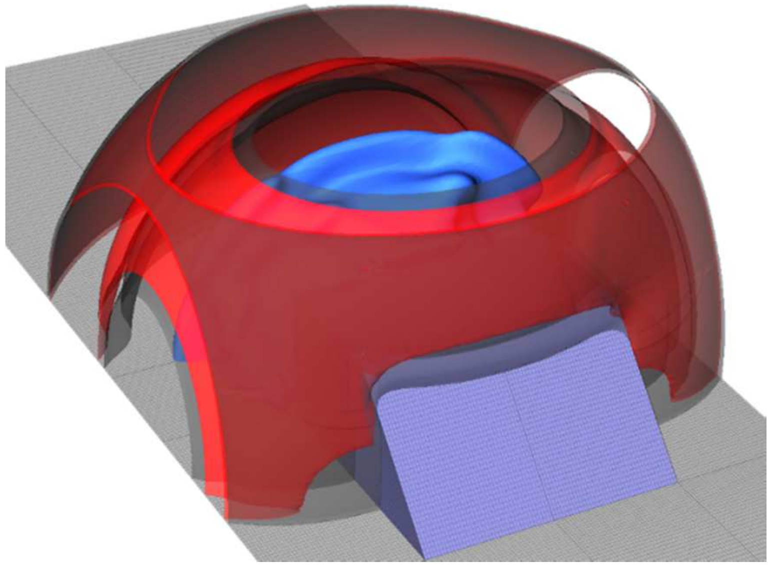


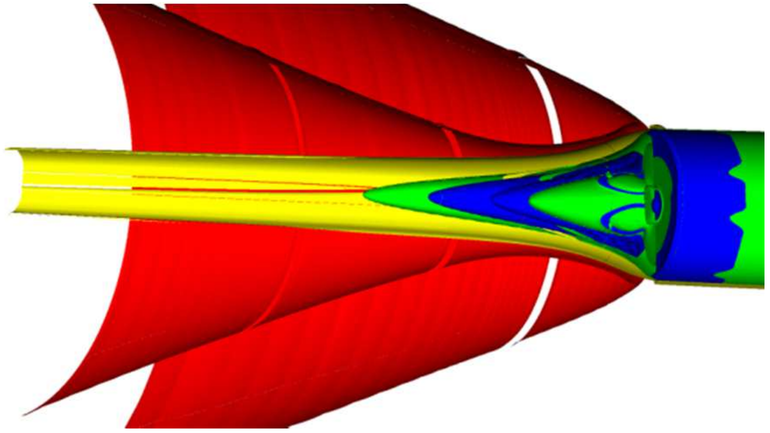

Domaine 1
Domaine 2
i
j
j
i
imin, jmin
imax, jmin
imin, jmax
imin, jmin
imin, jmax
imax, jmax
imax, jmax
imax, jmin
Zone commune du domaine 1:
imax, imax, jmin, jmax, kmin, kmin
Zone commune du domaine 2:
imin, imax, jmax, jmax, kmin, kmin

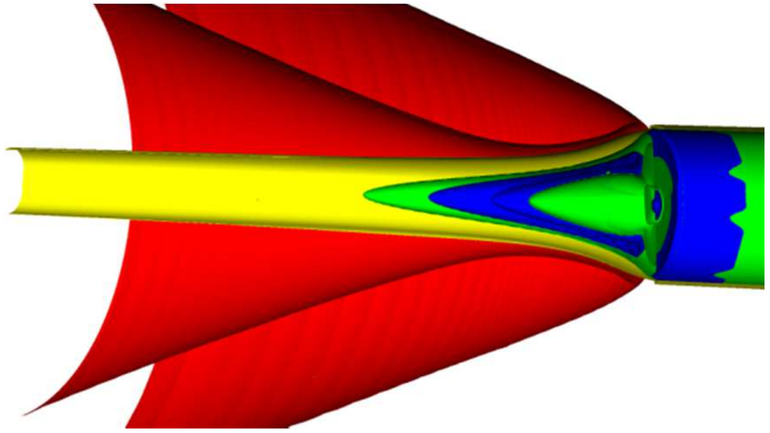


Handling ghost zones : Results









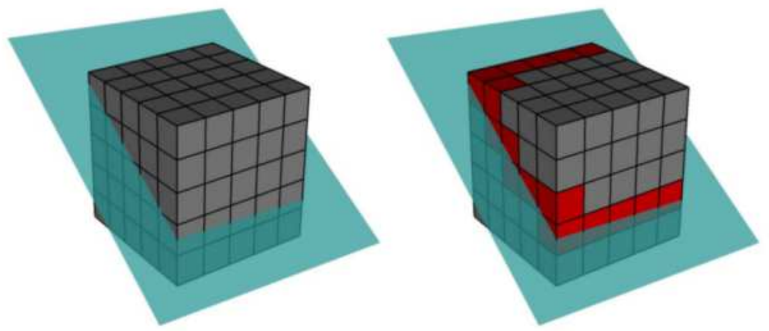
Optimization with data and spatial extents

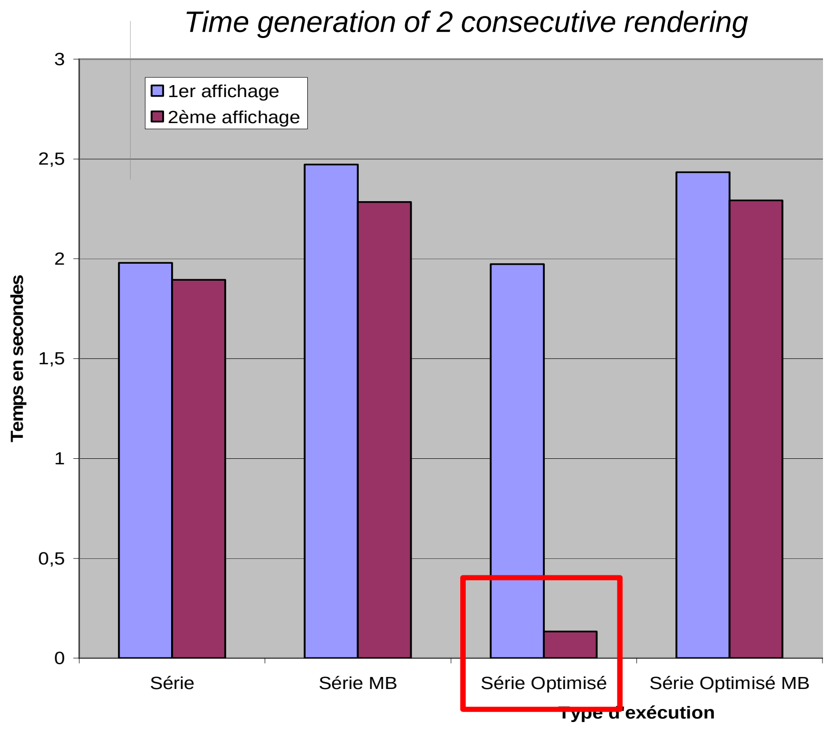
Generate extends

- From the header
- On the fly

Results

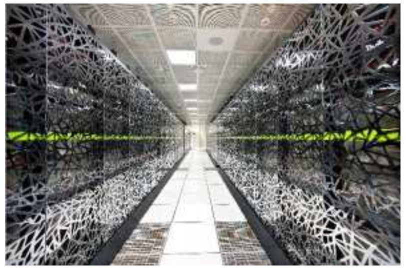
- Speed-up the second rendering in a monobloc setting

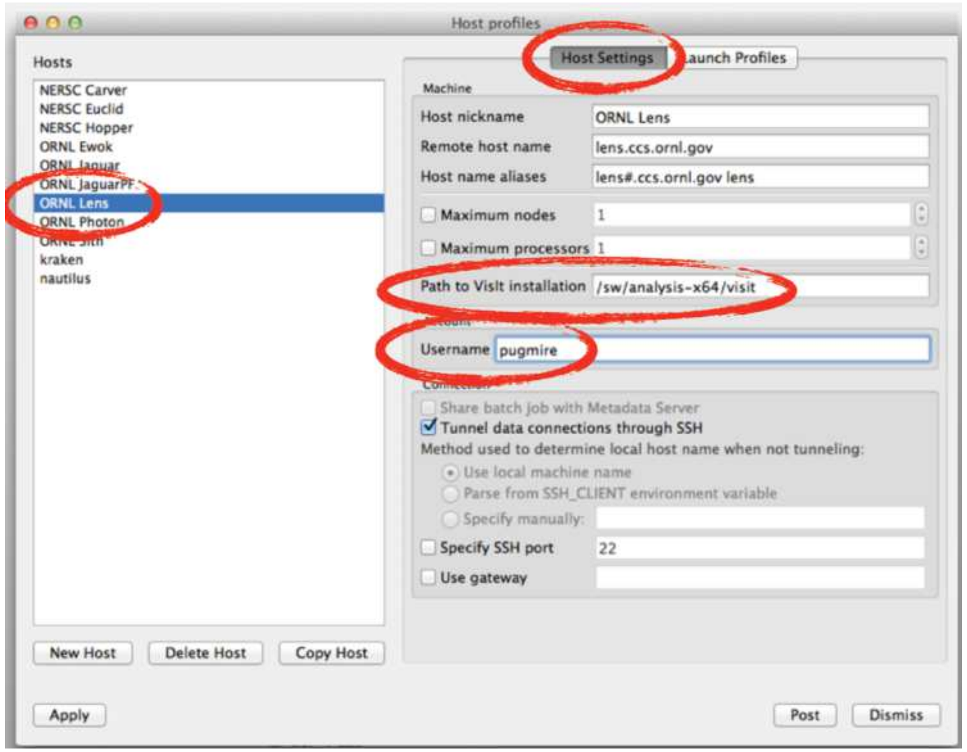




Remote visualization

- Connexion to the super computer TERA
  - → Communication with SSH (both ways)
  - → Remote data and compute engines
  - → Simple to use for a user from the VisIt interface





Hosts

NERSC Carver
NERSC Euclid
NERSC Hopper
ORNL Ewok
ORNL Jaguar
ORNL JaguarPF
ORNL Lens
ORNL Photon
ORNL Sith
kraken
nautilus

Machine

Host nickname	ORNL Lens
Remote host name	lens.ccs.ornl.gov
Host name aliases	lens#.ccs.ornl.gov lens
☐ Maximum nodes	1
☐ Maximum processors	1
Path to VisIt installation	/sw/analysis-x64/visit

Account

Username	pugmire

Connection

☐ Share batch job with Metadata Server
☑ Tunnel data connections through SSH
Method used to determine local host name when not tunneling:
● Use local machine name
○ Parse from SSH_CLIENT environment variable
○ Specify manually:
☐ Specify SSH port 22
☐ Use gateway

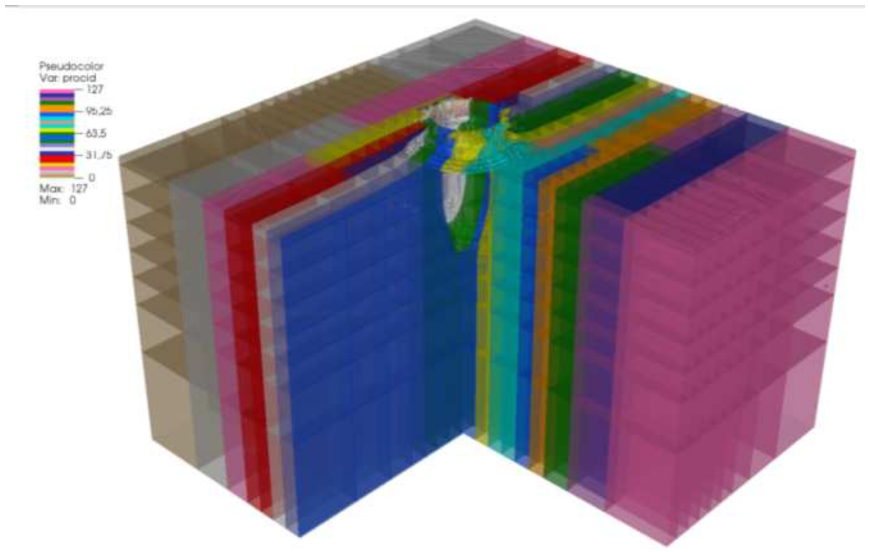
Large remote data analysis

Production data

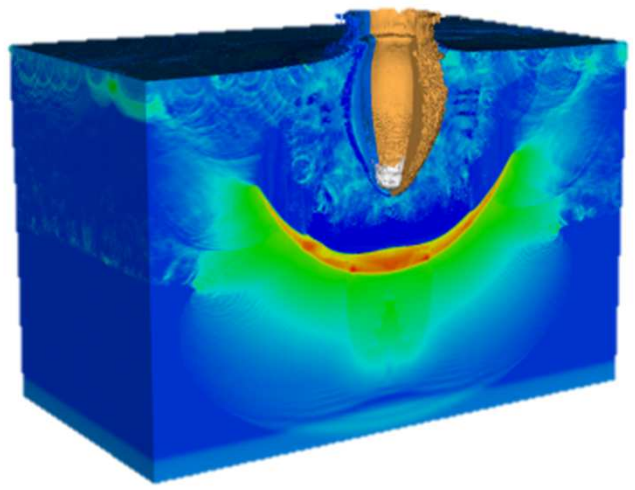
- 315 millions of cells in 2048 domains
- 2048 binary files : 25 GB for 1 time step

Caracteristics

- Specific MTMD binary reader
- Visualized on TERA with 128 processors



Step	Time
Connexion to the remote partition	30 seconds
Initialization of 128 visit parralel engines	30 seconds
Read 2048 binary files	30 seconds
Parallel visualization computation	10 seconds
Interaction	< 1 second



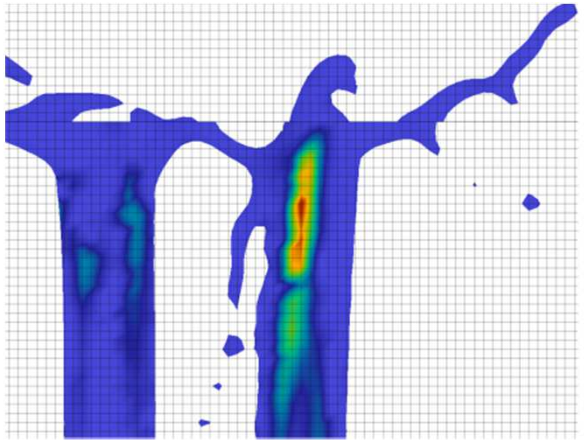
Material Interface reconstruction

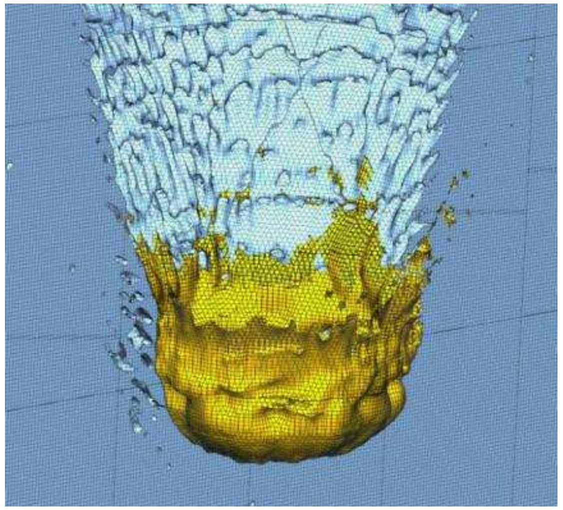
Method

- J.S. Meredith, « *Material Interface Reconstruction in VisIt* », Lawrence Livermore National Laboratory, technical report UCRL-CONF-209351, 2004.

Implementation

- *PopulateMetaData* : Read only the first file
  - ReadMetaData(0) // All the info in each file
  - GenerateAllReader
  - GenerateAllFileName
- *GetMesh(n)*
  - ReadMetaData(n)
  - MeshMetaData
  - MaterialMetaData
  - ScalarMetaData
  - OtherMetaData
  - Read and build X Y Z
- *AuxilaryData (n) // use metadata loaded in the getMesh*
  - Pure cells
  - Mixted cells (#cellule , [(mat1,FP1%),(mat2,FP2%) …])
  - Empty cells

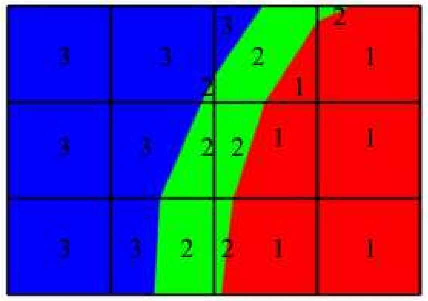




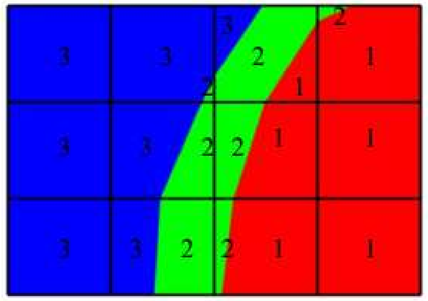
Material Interface reconstruction

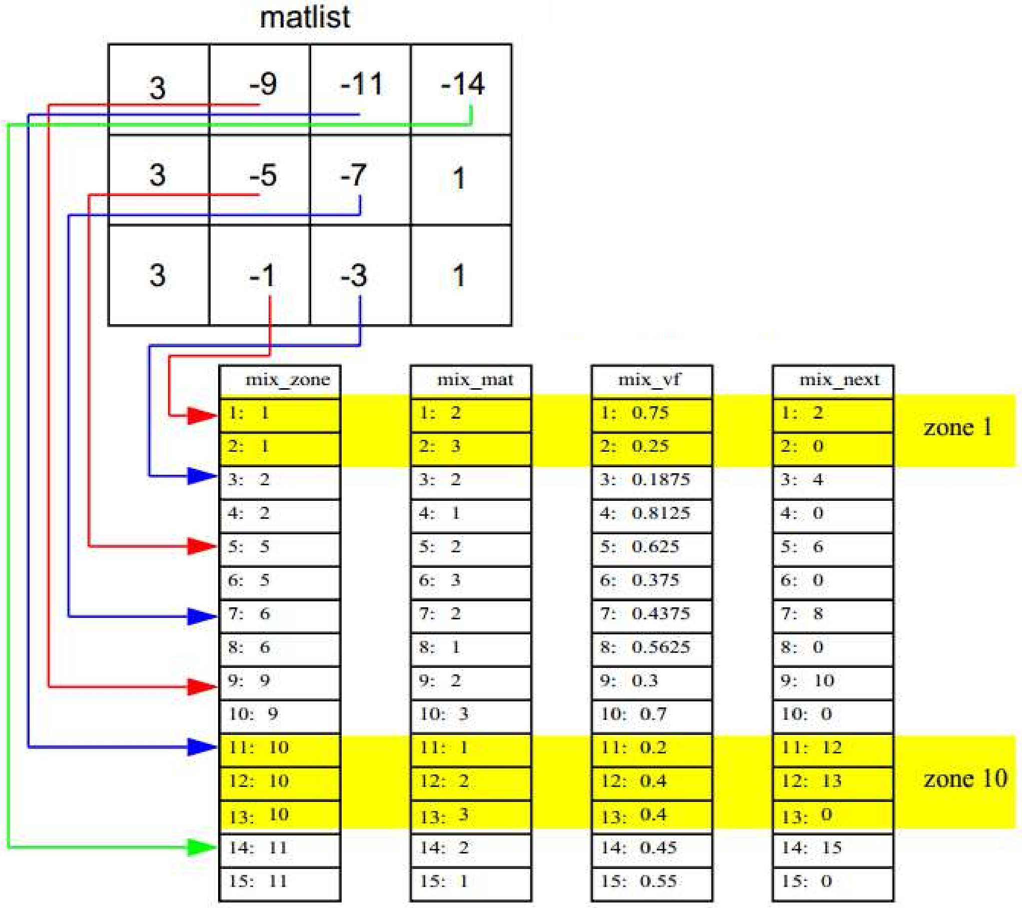
Implementation

Material numbers



Material numbers





3	-9	-11	-14
3	-5	-7	1
3	-1	-3	1

mix_zone	mix_mat	mix_vf	mix_next
1: 1	1: 2	1: 0.75	1: 2
2: 1	2: 3	2: 0.25	2: 0
3: 2	3: 2	3: 0.1875	3: 4
4: 2	4: 1	4: 0.8125	4: 0
5: 5	5: 2	5: 0.625	5: 6
6: 5	6: 3	6: 0.375	6: 0
7: 6	7: 2	7: 0.4375	7: 8
8: 6	8: 1	8: 0.5625	8: 0
9: 9	9: 2	9: 0.3	9: 10
10: 9	10: 3	10: 0.7	10: 0
11: 10	11: 1	11: 0.2	11: 12
12: 10	12: 2	12: 0.4	12: 13
13: 10	13: 3	13: 0.4	13: 0
14: 11	14: 2	14: 0.45	14: 15
15: 11	15: 1	15: 0.55	15: 0

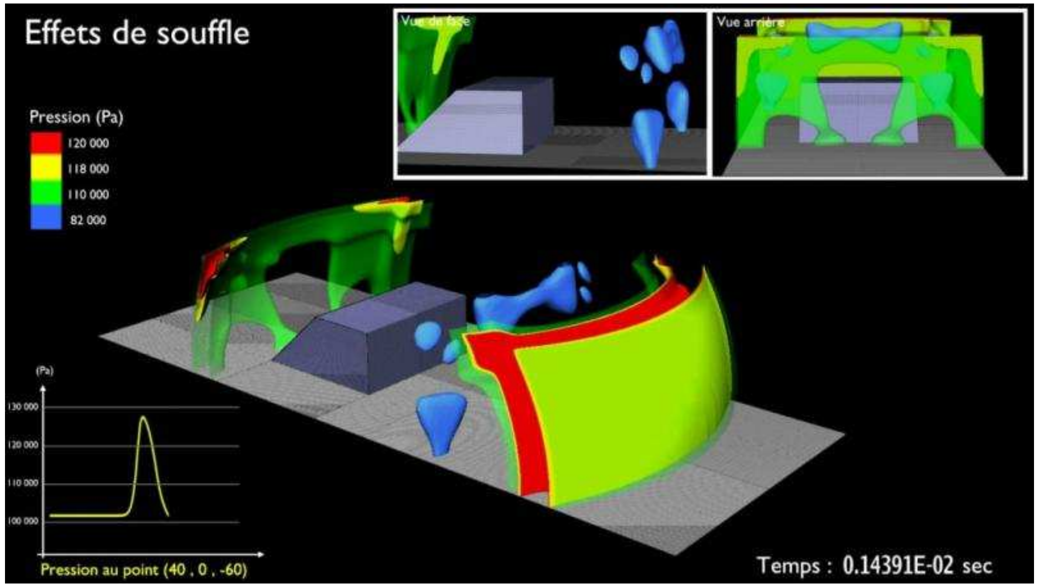
Example : Simulation of the breath effect

Simulation

- Analysis of the pressure by the resolution of Navier-Stokes equations
- Experiment in reduce scale
- Time varying multi-blocs curvilinear mesh
- *NetCdf* based data and corellation with measured pressure (curve)

Results

- Comparison in between the computed and mesured data on the same visualization



DEPLOY OPEN SOURCE VISUALIZATION SOFTWARE ON CEA COMPUTING FACILITIES

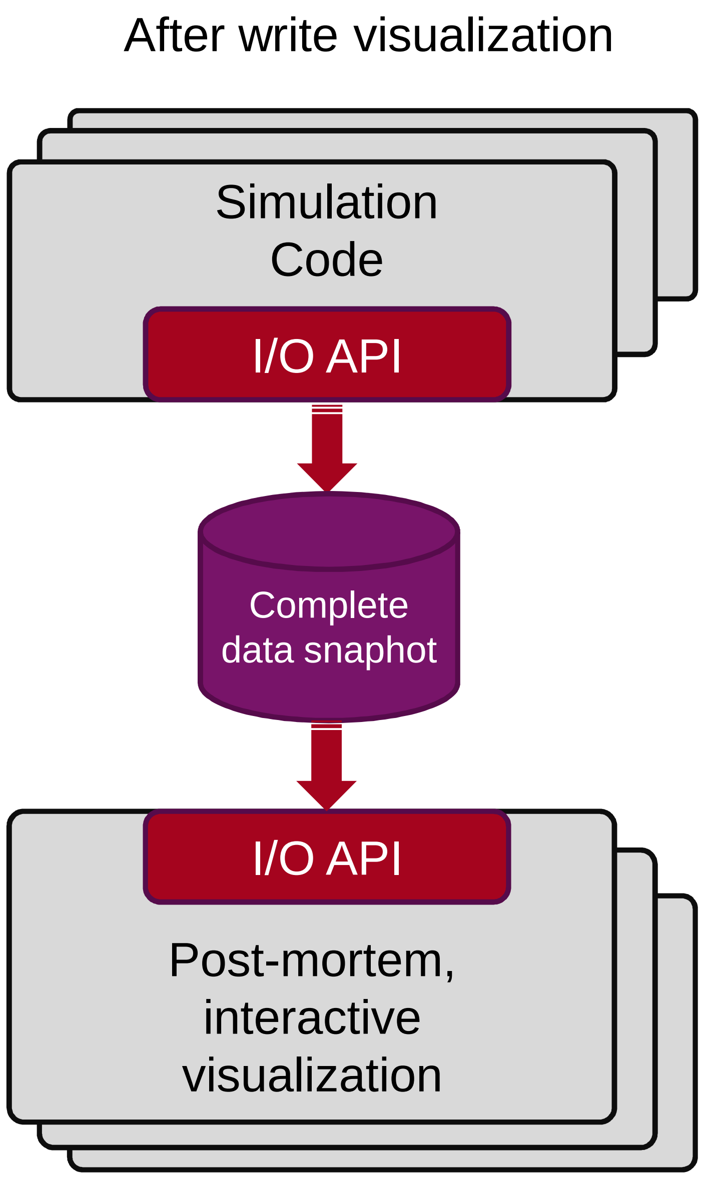
-

CATALYST

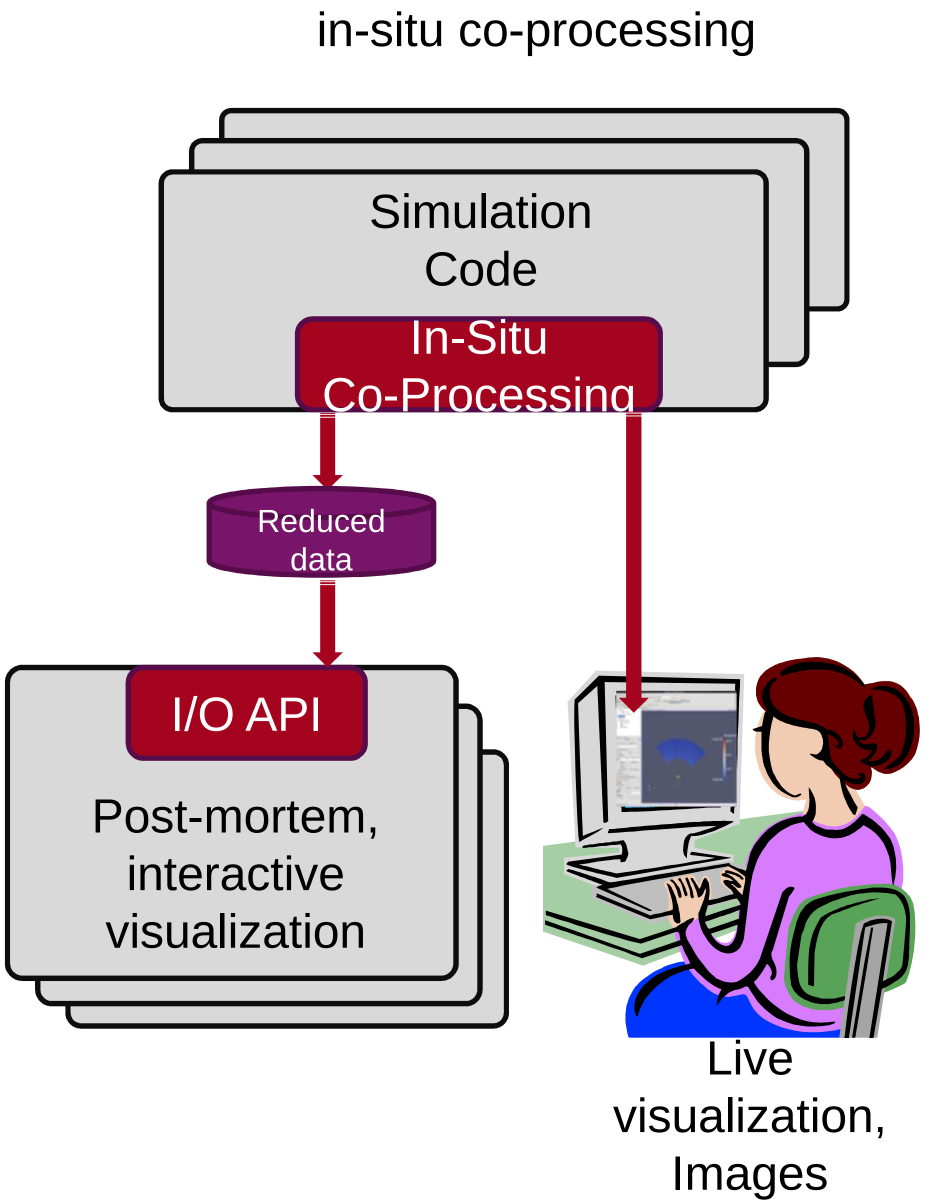
Outline

- Principles
- Catalyst software architecture
- Use case with Arcane simulation platform
- Live visualization
- Results & conclusion

Principles



After write visualization
Simulation Code
I/O API
Complete data snaphot
I/O API
Post-mortem, interactive visualization



in-situ co-processing
Simulation Code
In-Situ Co-Processing
Reduced data
I/O API
Post-mortem, interactive visualization
Live visualization, Images

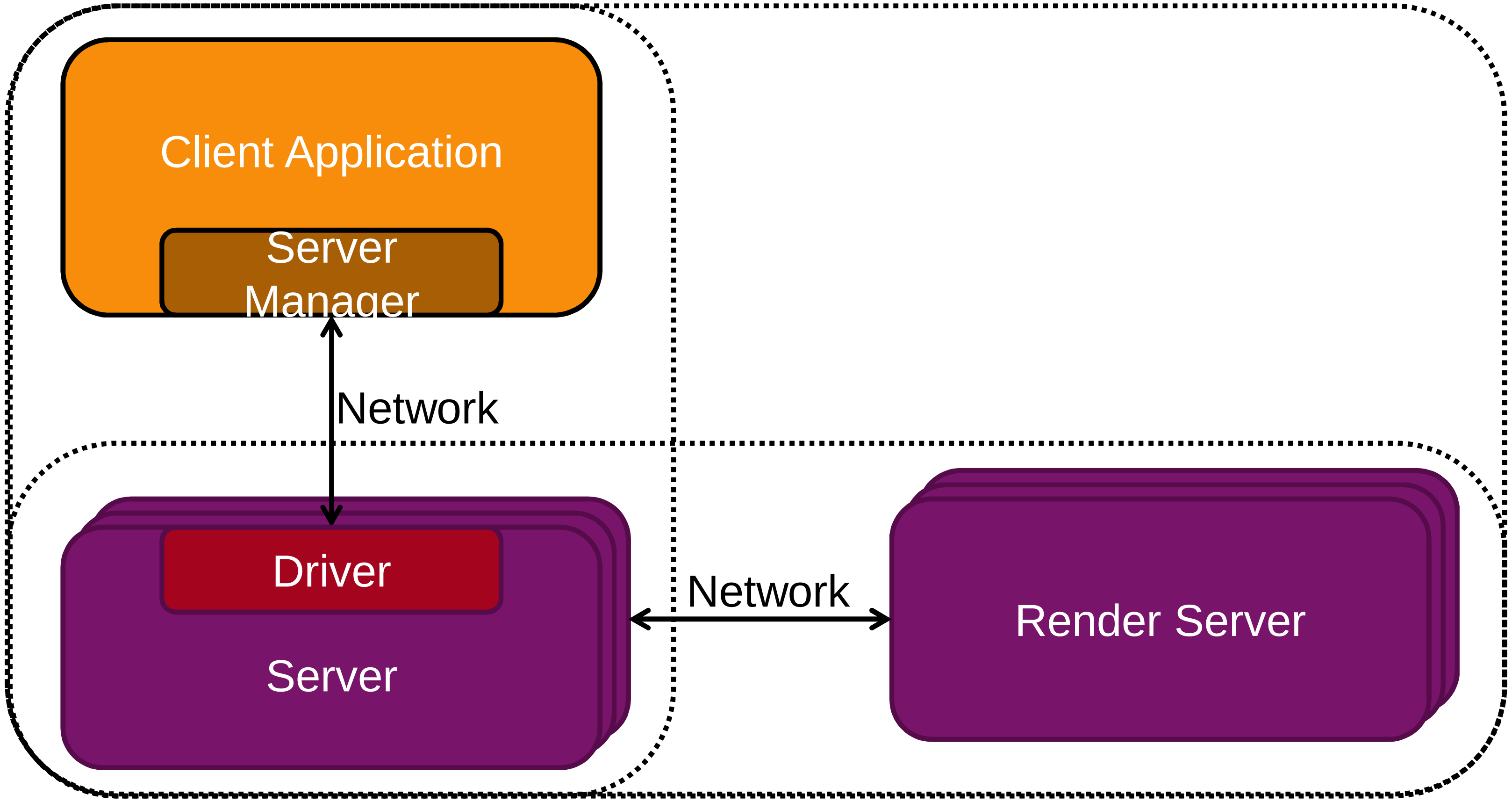
Available Tools

ICARUS : ParaView + Modified HDF5

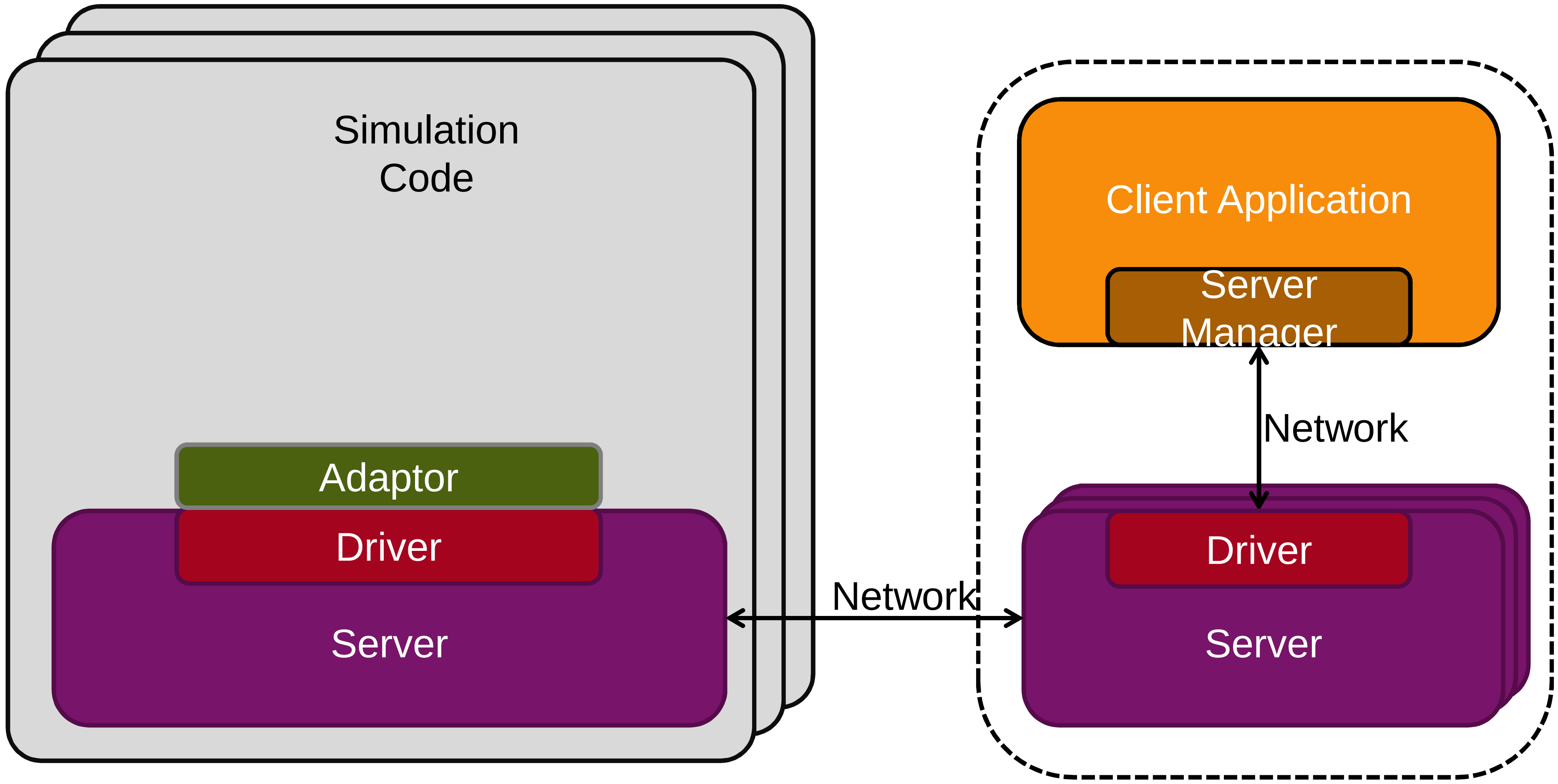
Libsim : VisIt’s In-Situ and steering API

Catalyst : ParaView + Modified driver (so-called adpator)

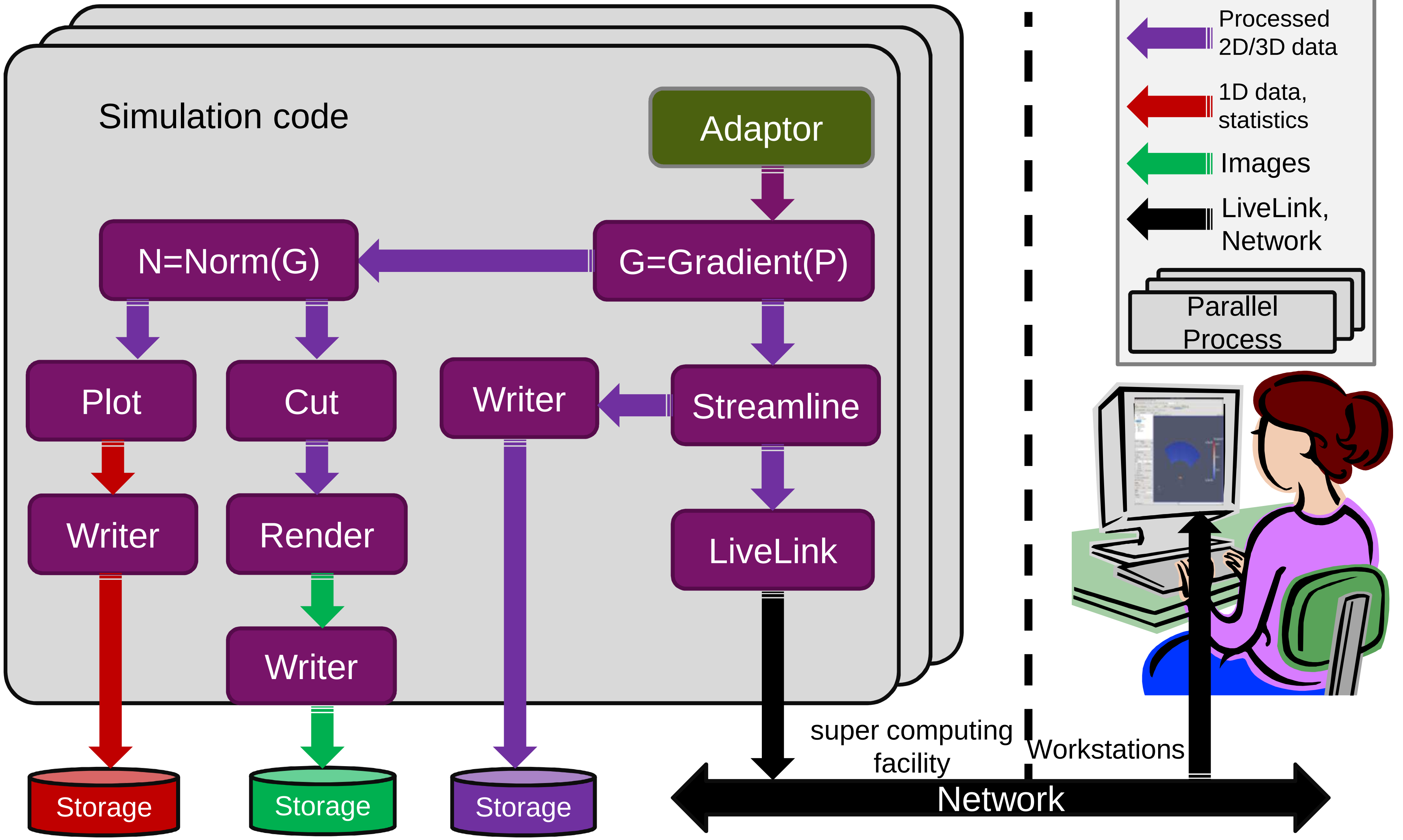
Paraview software components



Catalyst software coupling scheme



Catalyst data flow



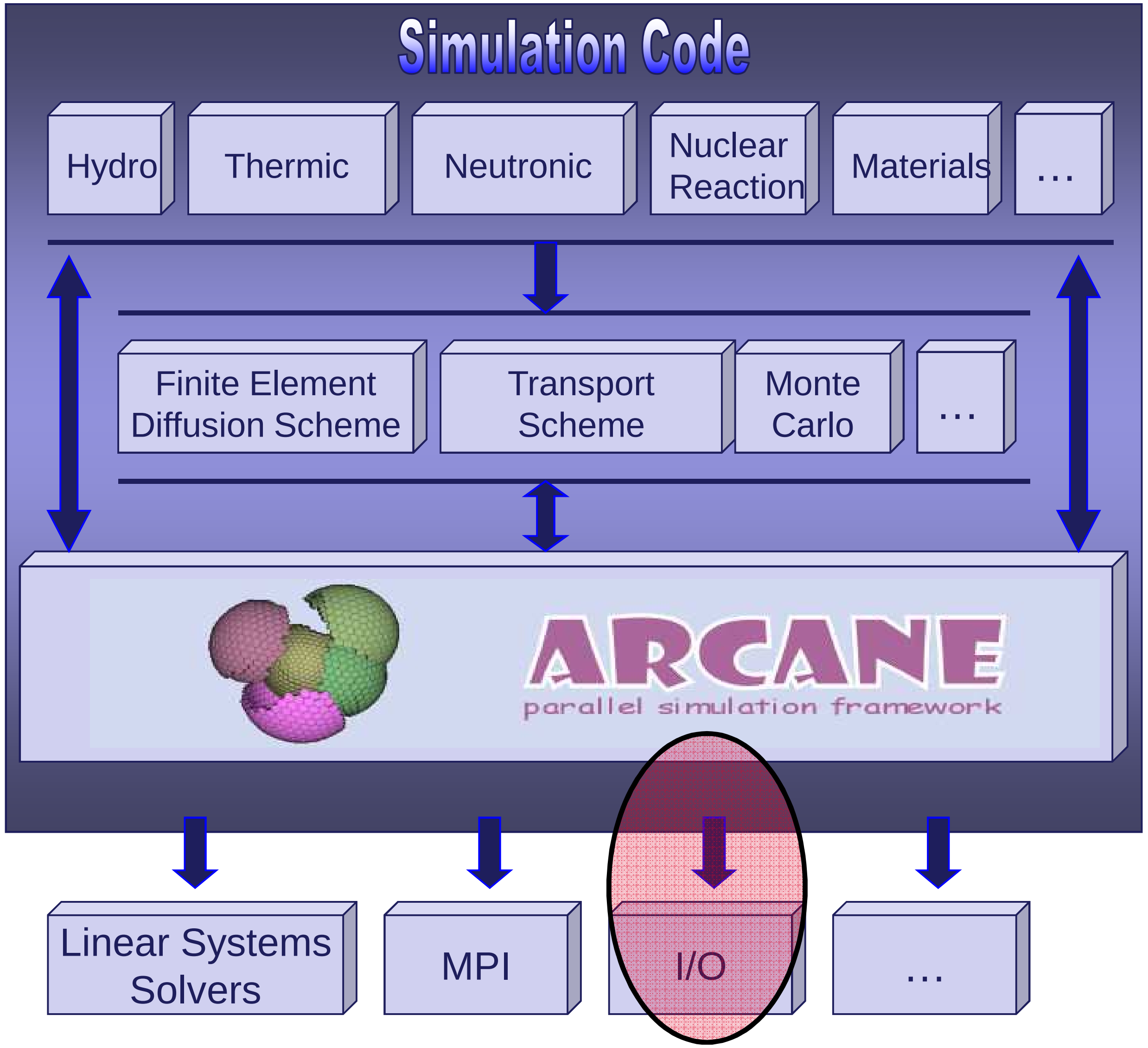
Arcane simulation code platform

Physics Modules

Numerical services

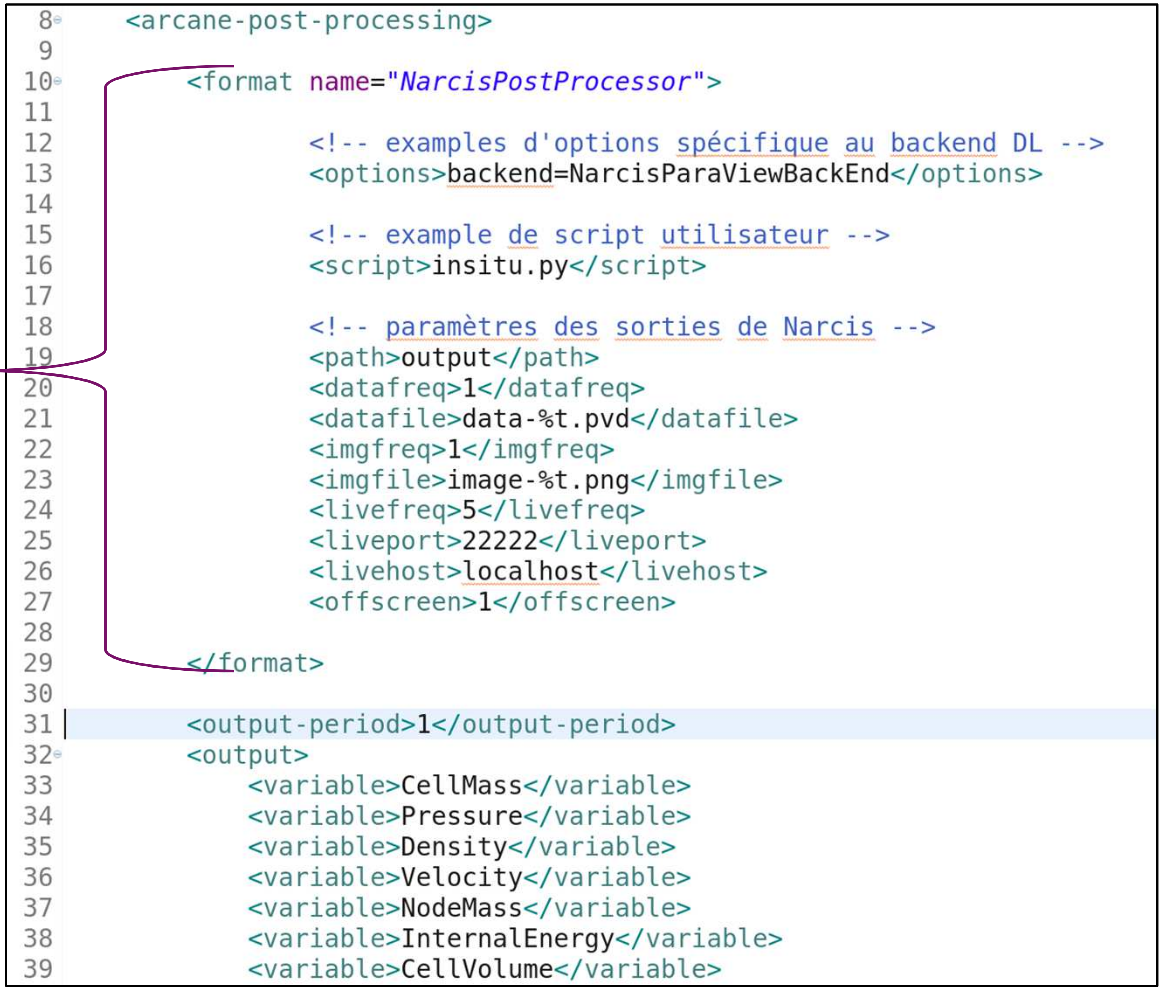
Arcane Framework

External libraries



The Arcane development framework. *2009 G. Grospellier and B. Lelandais In Proceedings of the 8th workshop on Parallel/High-Performance Object-Oriented Scientific Computing (POOSC '09).*

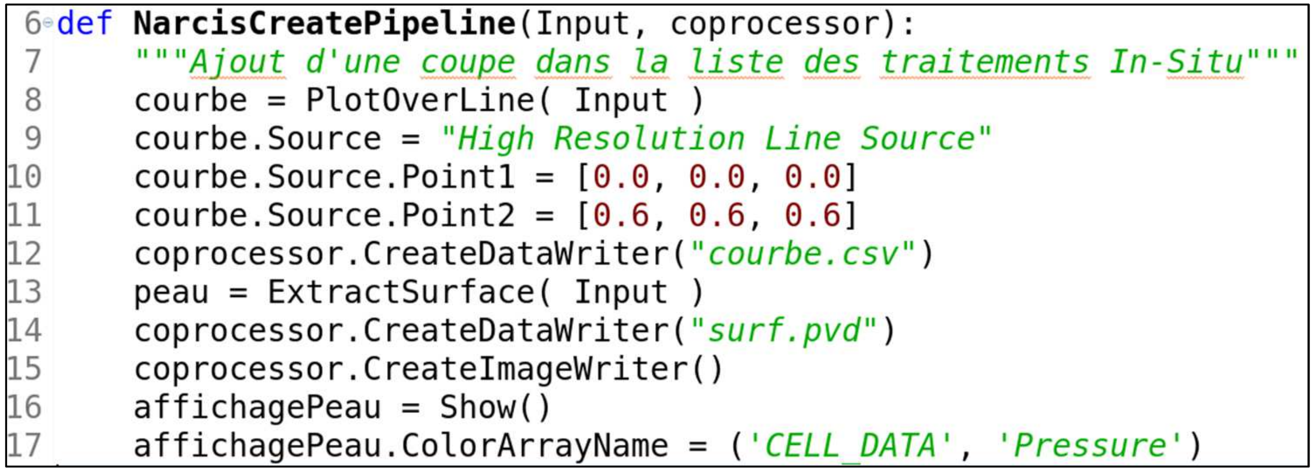
Arcane use case



```
<arcane-post-processing>

    <format name="NarcisPostProcessor">

            <!-- examples d'options spécifique au backend DL -->
            <options>backend=NarcisParaViewBackEnd</options>

            <!-- example de script utilisateur -->
            <script>insitu.py</script>

            <!-- paramètres des sorties de Narcis -->
            <path>output</path>
            <datafreq>1</datafreq>
            <datafile>data-%t.pvd</datafile>
            <imgfreq>1</imgfreq>
            <imgfile>image-%t.png</imgfile>
            <livefreq>5</livefreq>
            <liveport>22222</liveport>
            <livehost>localhost</livehost>
            <offscreen>1</offscreen>

    </format>

    <output-period>1</output-period>
    <output>
        <variable>CellMass</variable>
        <variable>Pressure</variable>
        <variable>Density</variable>
        <variable>Velocity</variable>
        <variable>NodeMass</variable>
        <variable>InternalEnergy</variable>
        <variable>CellVolume</variable>
```

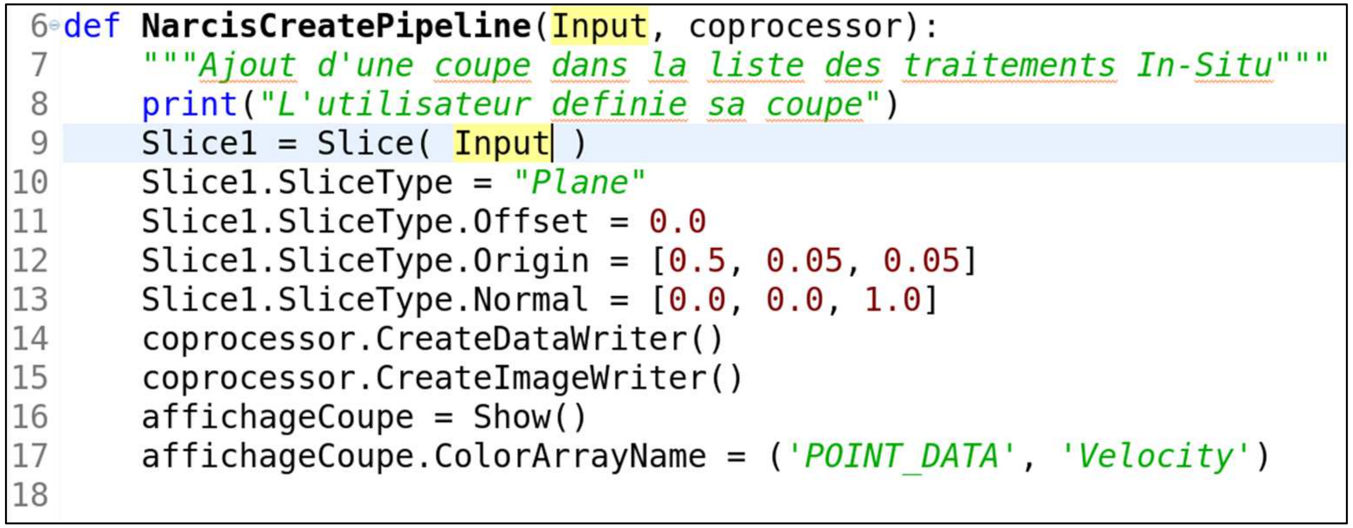
output to catalyst binding rathen than to disk

Arcane use case



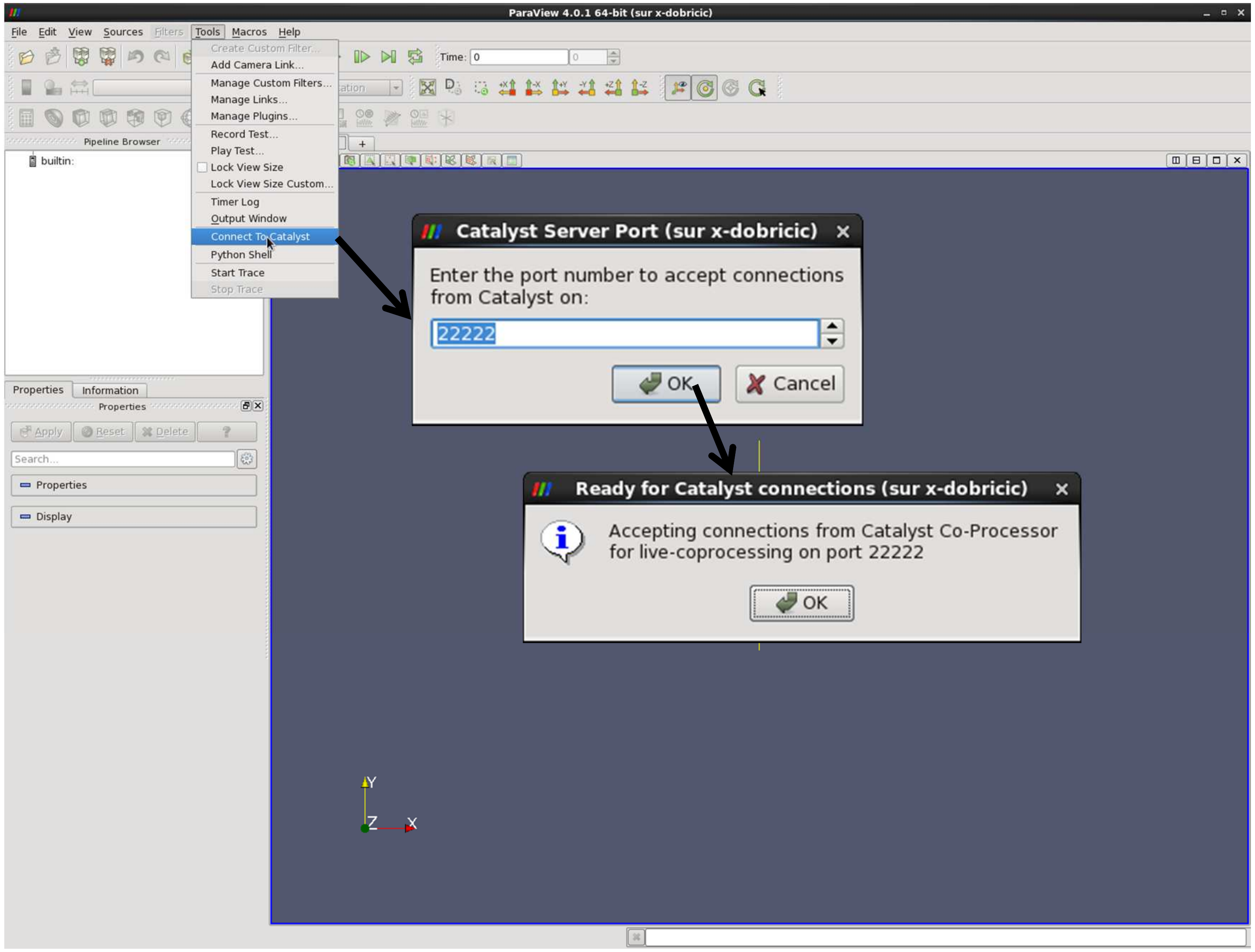
```python
def NarcisCreatePipeline(Input, coprocessor):
    """Ajout d'une coupe dans la liste des traitements In-Situ"""
    courbe = PlotOverLine( Input )
    courbe.Source = "High Resolution Line Source"
    courbe.Source.Point1 = [0.0, 0.0, 0.0]
    courbe.Source.Point2 = [0.6, 0.6, 0.6]
    coprocessor.CreateDataWriter("courbe.csv")
    peau = ExtractSurface( Input )
    coprocessor.CreateDataWriter("surf.pvd")
    coprocessor.CreateImageWriter()
    affichagePeau = Show()
    affichagePeau.ColorArrayName = ('CELL_DATA', 'Pressure')
```

co-processing script defines pipeline to be executed during simulation run



```python
def NarcisCreatePipeline(Input, coprocessor):
    """Ajout d'une coupe dans la liste des traitements In-Situ"""
    print("L'utilisateur definie sa coupe")
    Slice1 = Slice( Input )
    Slice1.SliceType = "Plane"
    Slice1.SliceType.Offset = 0.0
    Slice1.SliceType.Origin = [0.5, 0.05, 0.05]
    Slice1.SliceType.Normal = [0.0, 0.0, 1.0]
    coprocessor.CreateDataWriter()
    coprocessor.CreateImageWriter()
    affichageCoupe = Show()
    affichageCoupe.ColorArrayName = ('POINT_DATA', 'Velocity')

```

Catalyst live link



Catalyst Server Port (sur x-dobricic)

Enter the port number to accept connections from Catalyst on:

22222

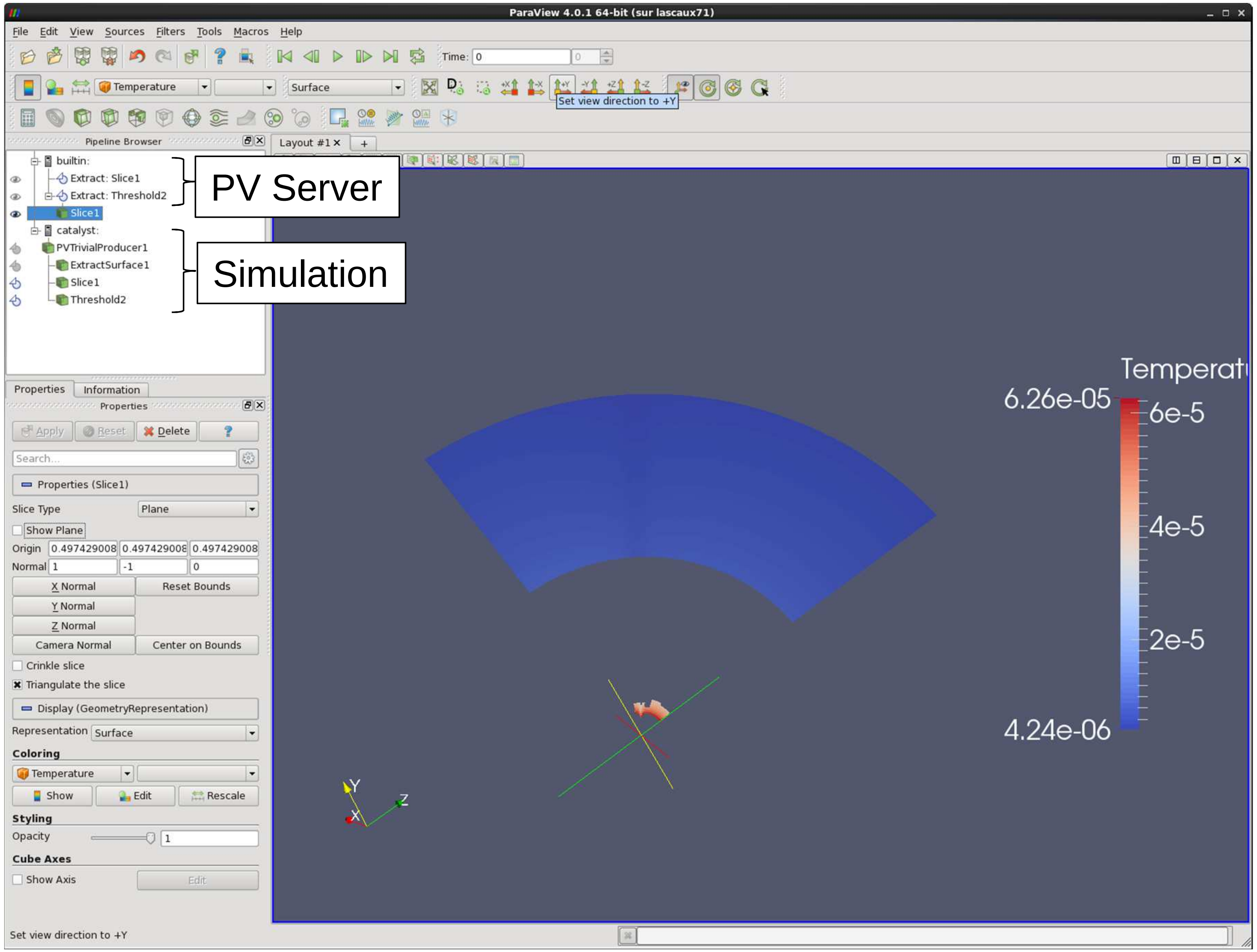
OK Cancel

Ready for Catalyst connections (sur x-dobricic)

Accepting connections from Catalyst Co-Processor for live-coprocessing on port 22222

OK

Catalyst live link



Catalyst: conclusion

- Opensource benefits
  - →Reuse existing, turn key application
  - →customizable, extendable with domain specific developments
  - →interaction with partners facing the same issues
- Time to results
  - →fast integration. most of the time spent on linking the libraries
  - →co-processing scripts generated or written using paraview.simple syntax
- Performance
  - →can be improved : zero copy, performance optimization
  - →already comparable to optimized I/O in most cases
  - →much faster for small outputs with about 100 cpus
- Adaptability
  - →many different grid types supported
- drawbacks
  - →still evolving architecture and API

CONTRIBUTE TO OPEN SOURCE VISUALIZATION SOFTWARE

-

VISION

Integrated simulation workbench : Needs

Data input and setup computation case

- Many simulation codes
- Both local and remote files : mesh, material …
- Ease data input and validation

Submit computations

- Many simulation machines local and remote
- Interactive and batch submissions
- Help to size the resources to allocate (time, memory, CPUs)

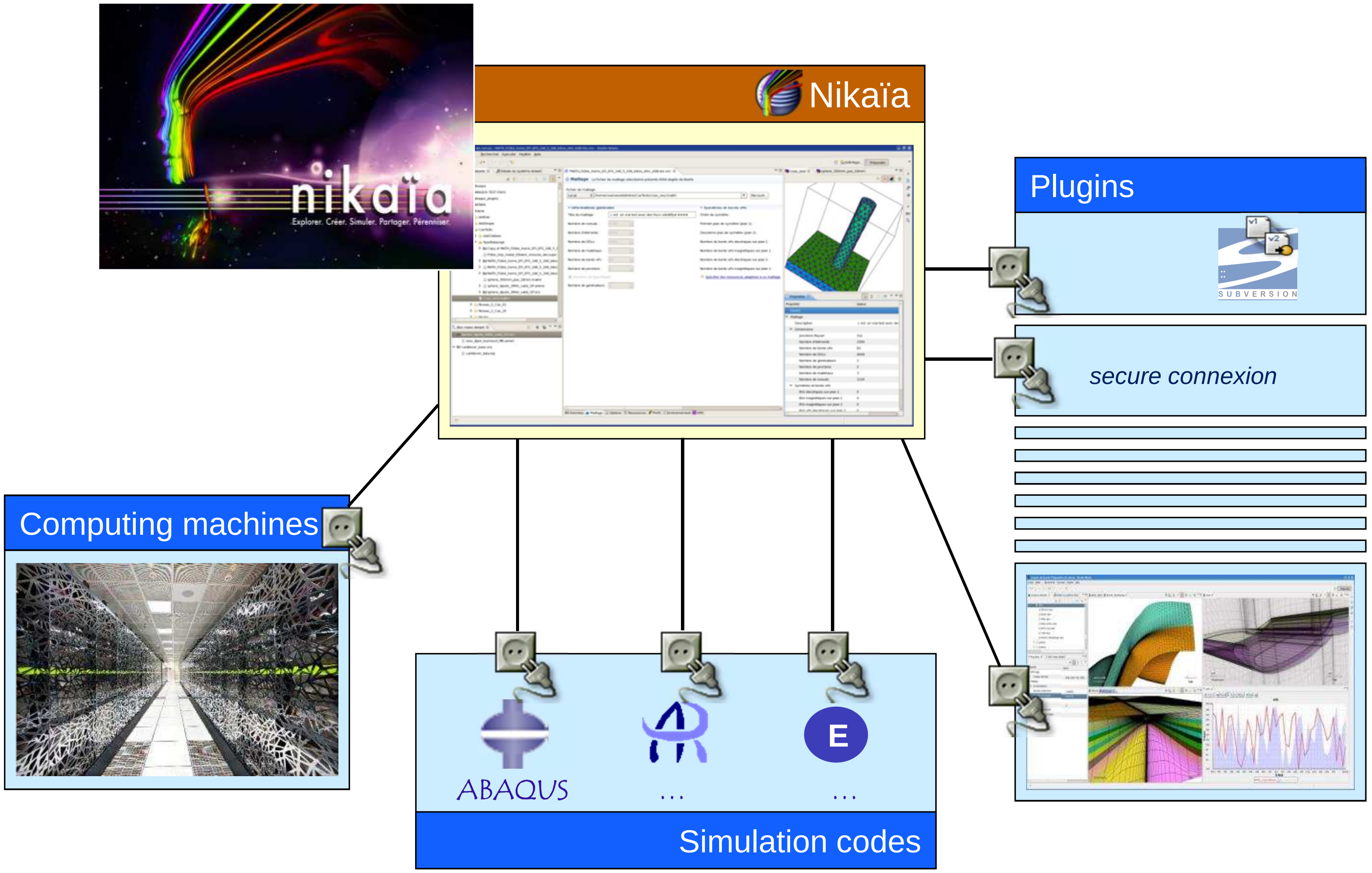
Analysis

- Simple and adapted analysis tools

Integrated simulation workbench : Requirements

- Security
  - Support network mechanism at CEA
    - SSH connexion
    - Strong security protocol
- Portability
  - Windows and Linux
- Design
  - Users of the simulation codes
    - Work on a local workstation
    - Exploit remote capabilities
    - Are not batch expert

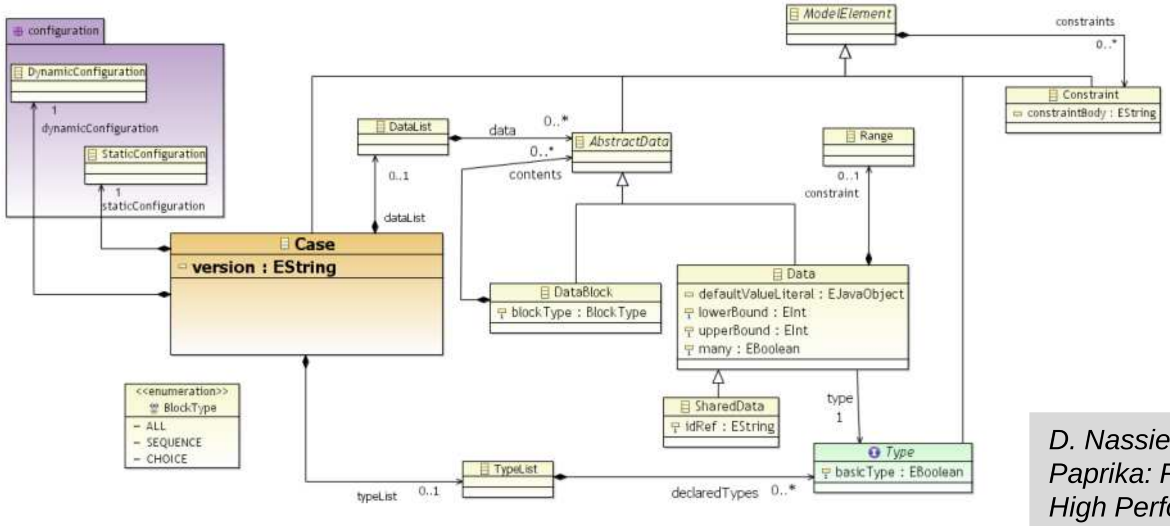
Integrated simulation workbench : Nikaïa



Eclipse RCP based applications

- Nikaïa is based on Eclipse
  - *Rich Client Platform* : devploment framework to produce applications
  - *Equinox* (OSGI) conception model
  - *Runtime* : execution
  - *SWT* : graphical components
  - *Jface* : advanced SWT compononents for GUI
  - *Eclipse* Workbench : mechanism to define views, perspectives, editors …

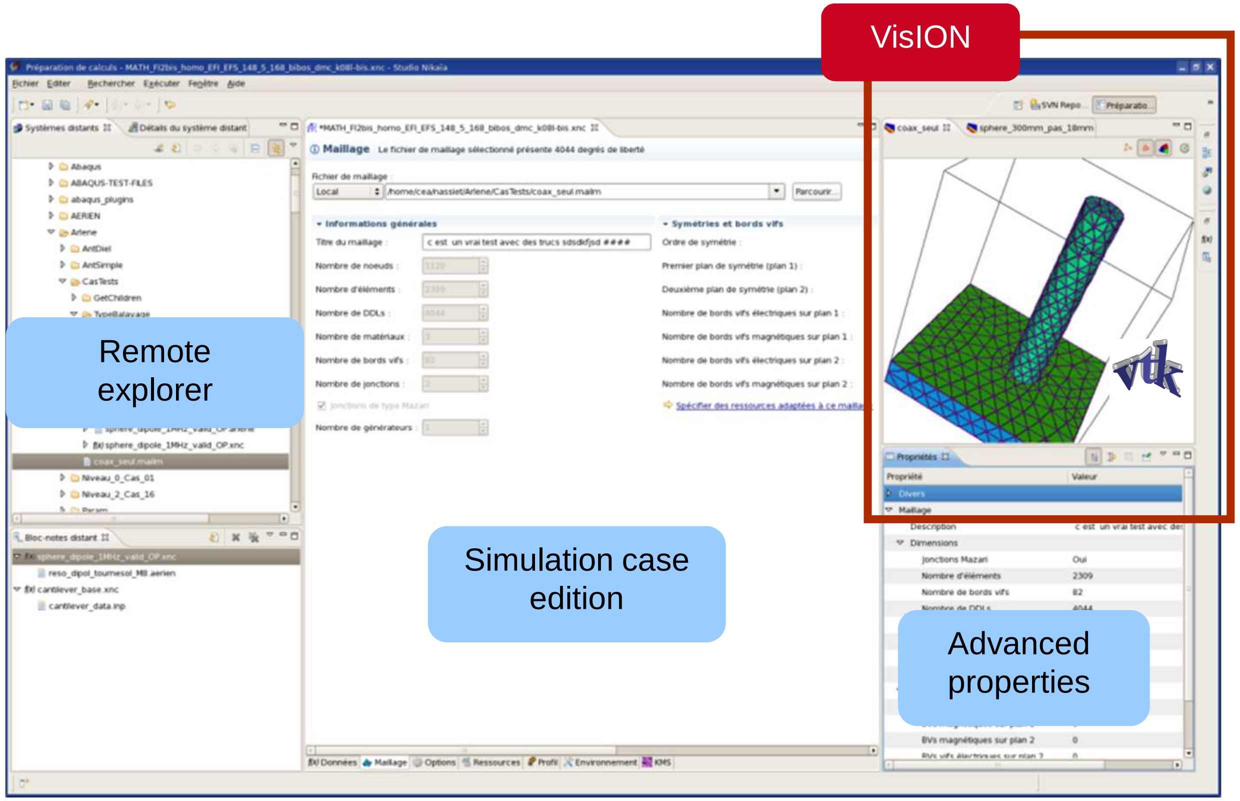
- Nikaïa uses Paprika (CEA/CESTA)
  - Modeling of a scientific dataset (Numerical metamodel)
  - Construction of a graphic editor based on dataset and GUI models (GUI metamodel)

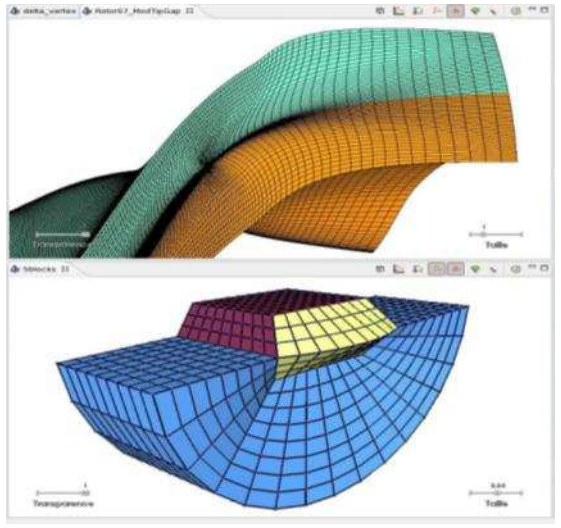


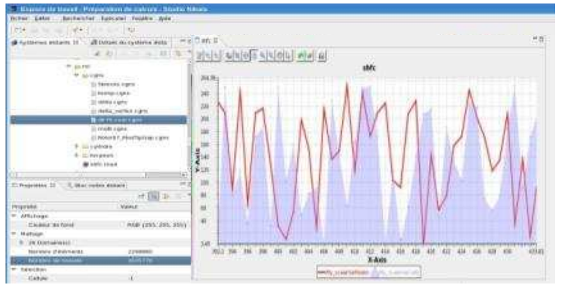
D. Nassiet, Y. Livet, M. Palyart, D. Lugato
Paprika: Rapid UI Development of Scientific Dataset Editors for High Performance Computing - 15th SDL Forum - 2011

Integrated visualization view : VisION

- Plugin integrated into the workbench Nikaïa
  - **Vis**ualisation **I**ntégrée à l'**O**util **N**ikaïa
  - Plugin useable from various software at CEA
  - Support specific data format used by the simulation codes
  - 3D Interaction for data input (properties, materials …)



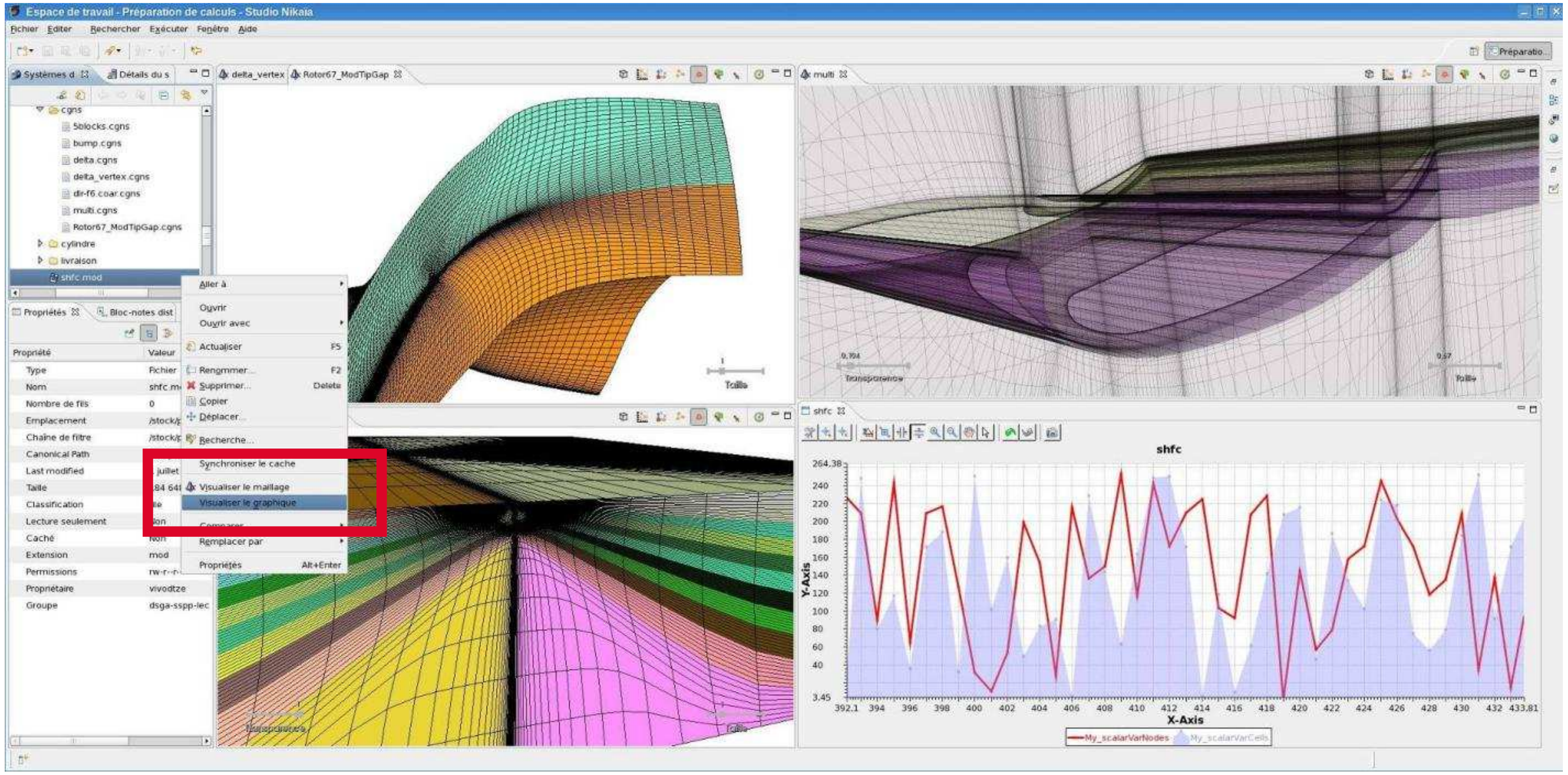




DE LA RECHERCHE À L'INDUSTRIE

cea

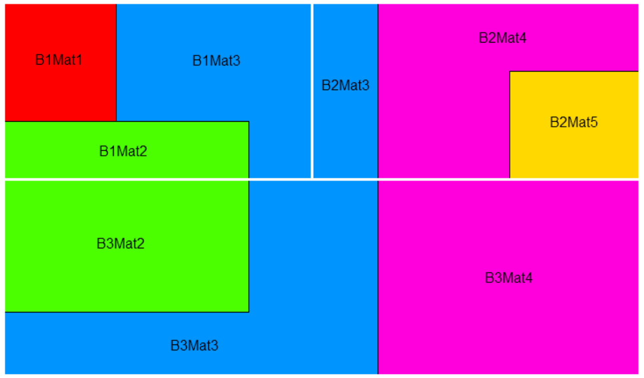
VisION : screenshot

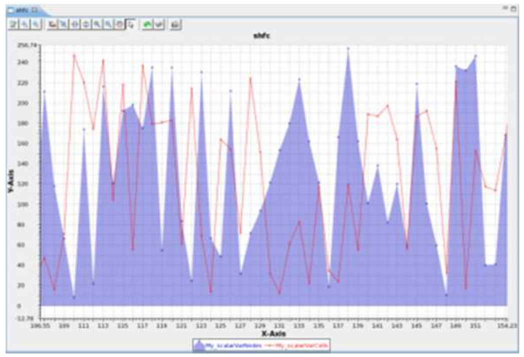


Propriété	Valeur
Type	Fichier
Nom	shfc.m
Nombre de fils	0
Emplacement	/stock/p
Chaîne de filtre	/stock/p
Canonical Path	
Last modified	juillet
Taille	84 641
Classification	le
Lecture seulement	lon
Caché	Non
Extension	mod
Permissions	rw-r--r-
Propriétaire	vivodtze
Groupe	dsga-sspp-lec

VisION : implementation details

- VTK based developments
  - Picking functionalities (blocks, boundaries …)
  - Materials and multi-blocks support
    - *vtkMultiBlockDataSet*
    - Block == Collection of *vtkActor*
  - Sub-settings
    - *vtkIdTypeArray* : scalar values to extract
    - *vtkSelectionNode* : entries to extract
    - *vtkSelection* : Selection nodes to perform
    - *vtkExtractSelection* : Extracted information
- Others
  - Curve plot with swt-xy-graph (SWT)
  - Eclipse Extension points for all the readers
  - CGNS reader with the JNI





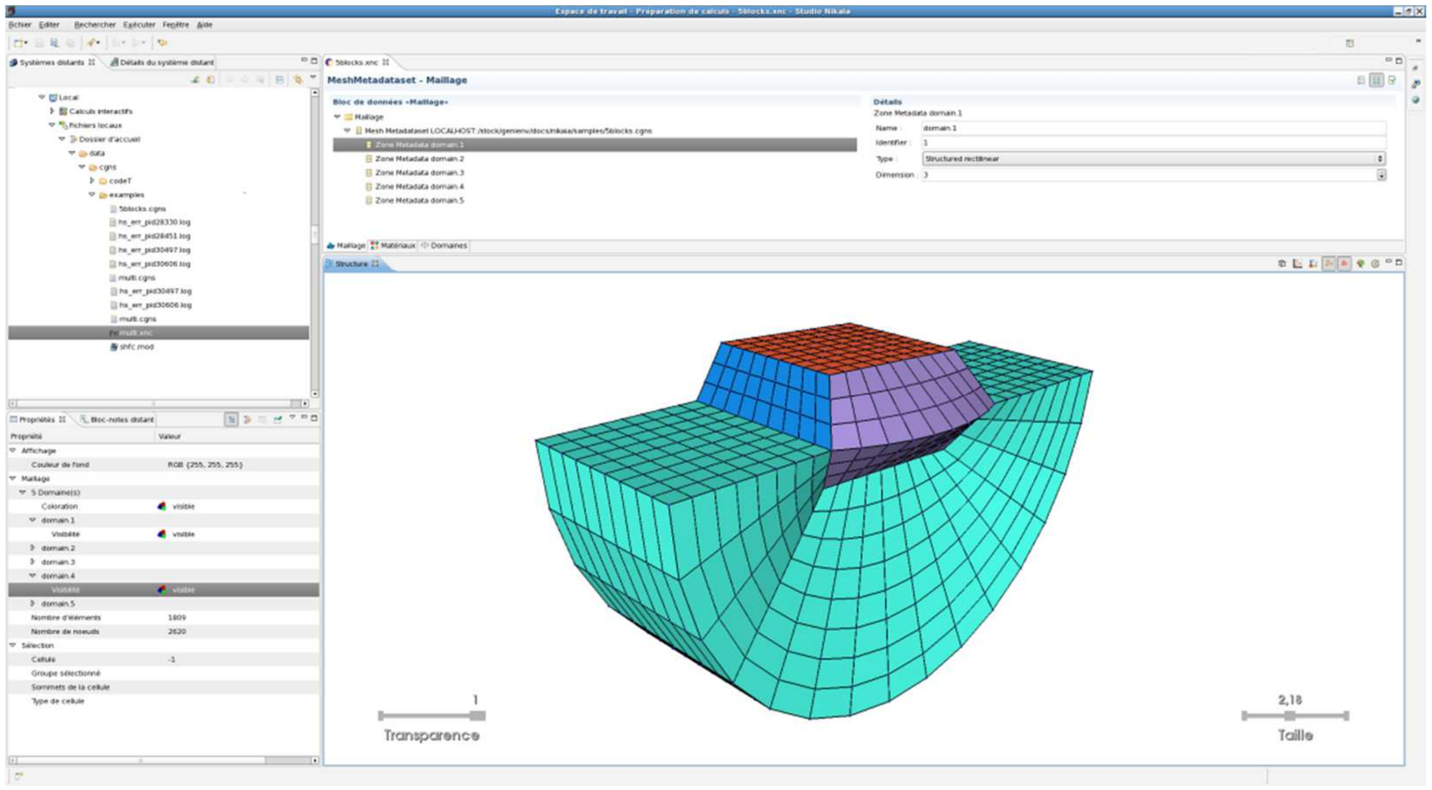
DE LA RECHERCHE À L'INDUSTRIE

cea

VisION : implementation details

Model based approach

- The 3D view becomes a structured view of the input model (IContentOutlinePage)
- Edition of the model modifies the input dataset
  - Boundary extraction, initial condition
- Picked information help the model edition



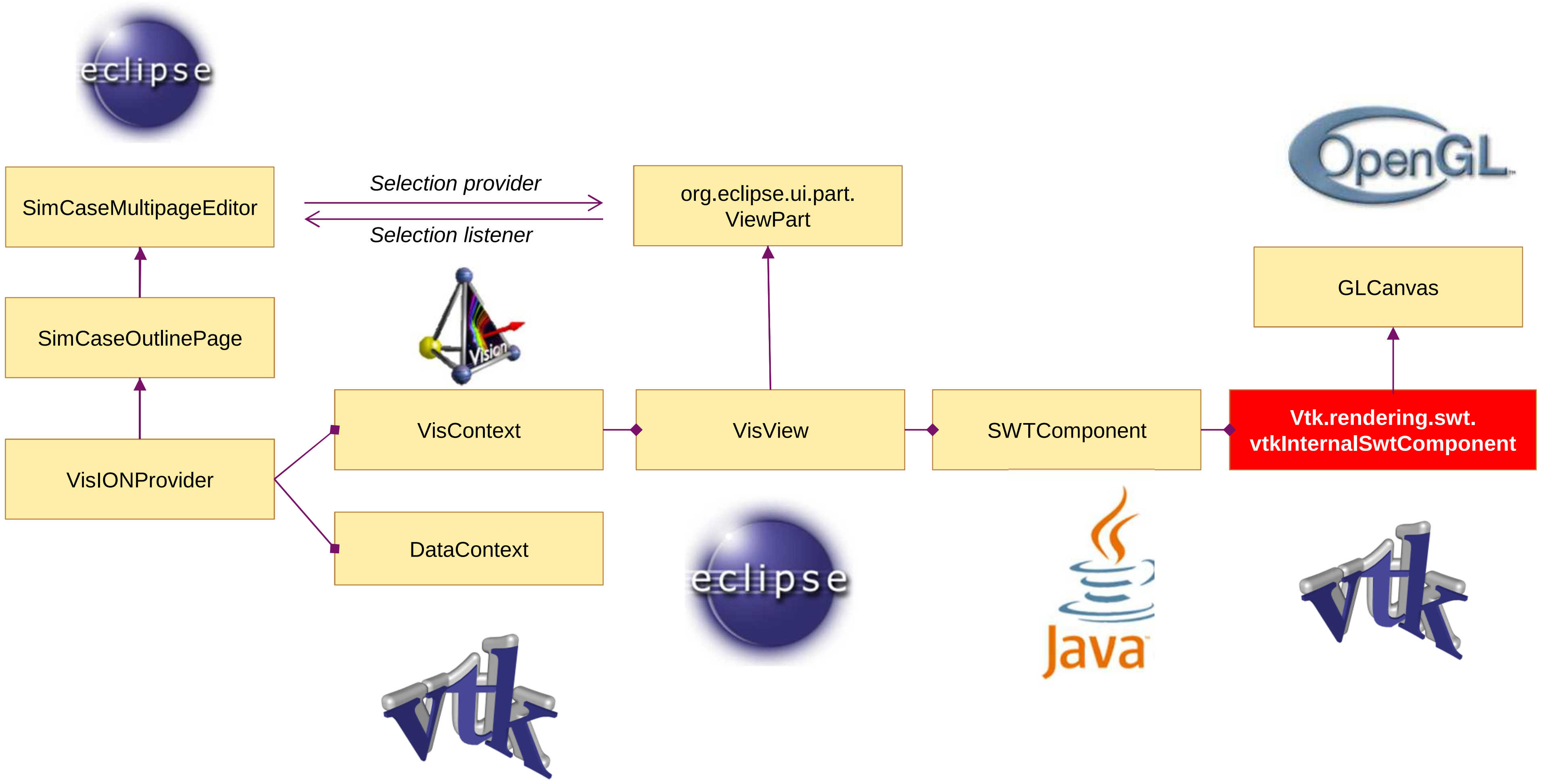
Détails
Zone Metadata domain 1

Name :	domain 1
Identifier :	1
Type :	Structured rectilinear
Dimension :	3

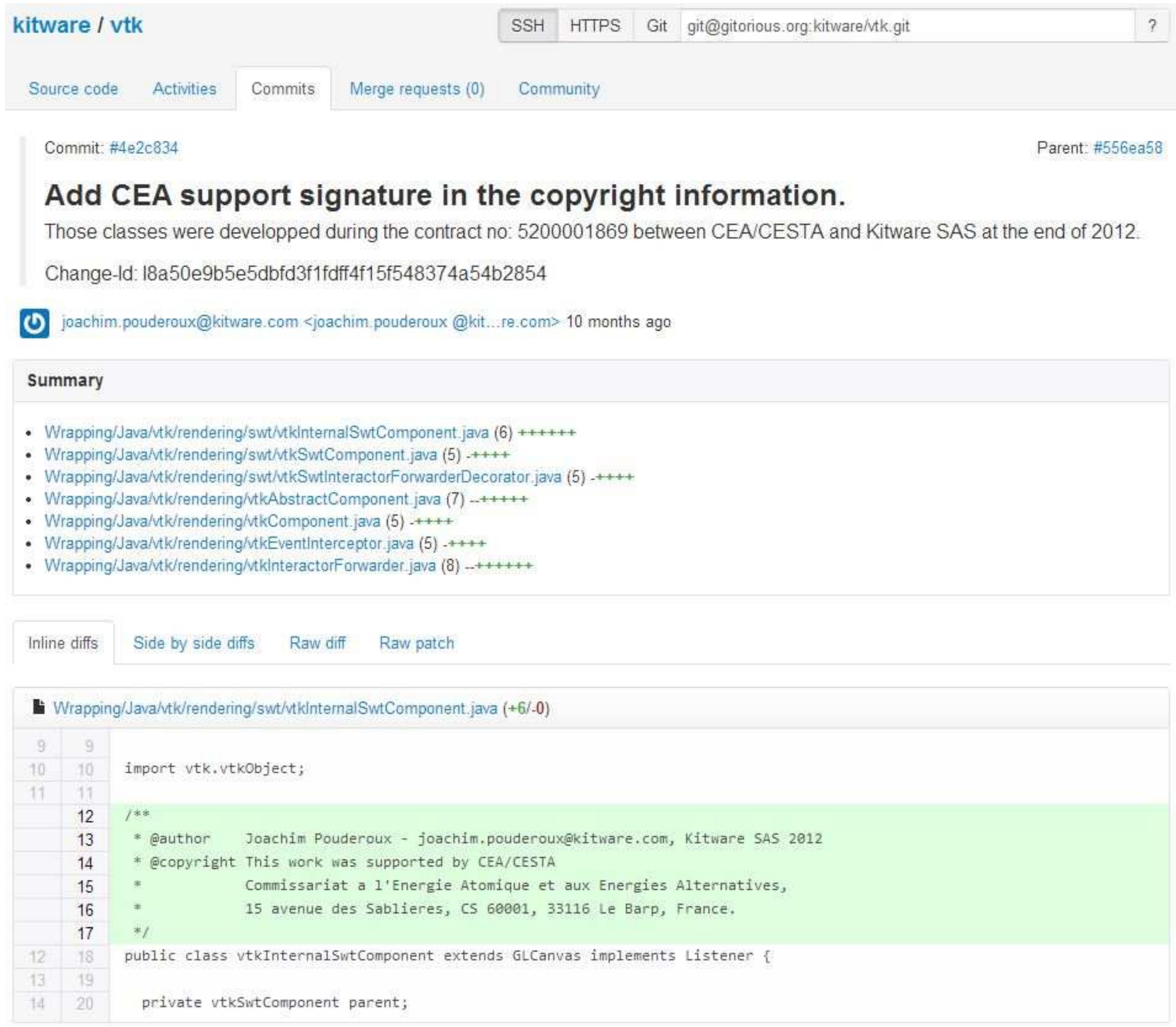
Propriété	Valeur
Affichage	
Couleur de fond	RGB {255, 255, 255}
Maillage	
5 Domaine(s)	
Coloration	visible
domain 1	
Visibilité	visible
domain 2	
domain 3	
domain 4	
Visibilité	visible
domain 5	
Nombre d'éléments	1809
Nombre de noeuds	2620
Sélection	
Cellule	-1
Groupe sélectionné	
Sommets de la cellule	
Type de cellule	

VisION architecture

- VTK contribution to integrate VTK Canvas into SWT components



Open source contribution to VTK



kitware / vtk

SSH HTTPS Git git@gitorious.org:kitware/vtk.git ?

Source code Activities Commits Merge requests (0) Community

Commit: #4e2c834 Parent: #556ea58

Add CEA support signature in the copyright information.

Those classes were developped during the contract no: 5200001869 between CEA/CESTA and Kitware SAS at the end of 2012.

Change-Id: I8a50e9b5e5dbfd3f1fdff4f15f548374a54b2854

joachim.pouderoux@kitware.com <joachim.pouderoux @kit...re.com> 10 months ago

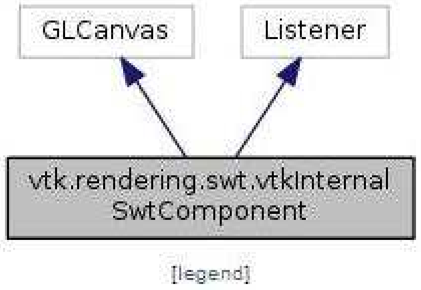
Summary

- Wrapping/Java/vtk/rendering/swt/vtkInternalSwtComponent.java (6) ++++++
- Wrapping/Java/vtk/rendering/swt/vtkSwtComponent.java (5) -++++
- Wrapping/Java/vtk/rendering/swt/vtkSwtInteractorForwarderDecorator.java (5) -++++
- Wrapping/Java/vtk/rendering/vtkAbstractComponent.java (7) --+++++
- Wrapping/Java/vtk/rendering/vtkComponent.java (5) -++++
- Wrapping/Java/vtk/rendering/vtkEventInterceptor.java (5) -++++
- Wrapping/Java/vtk/rendering/vtkInteractorForwarder.java (8) --++++++

Inline diffs Side by side diffs Raw diff Raw patch

Wrapping/Java/vtk/rendering/swt/vtkInternalSwtComponent.java (+6/-0)

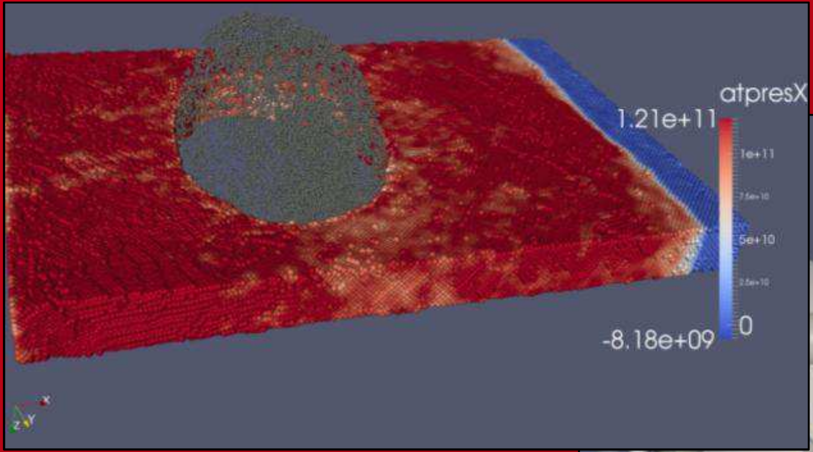
```
import vtk.vtkObject;

/**
 * @author    Joachim Pouderoux - joachim.pouderoux@kitware.com, Kitware SAS 2012
 * @copyright This work was supported by CEA/CESTA
 *            Commissariat a l'Energie Atomique et aux Energies Alternatives,
 *            15 avenue des Sablieres, CS 60001, 33116 Le Barp, France.
 */
public class vtkInternalSwtComponent extends GLCanvas implements Listener {

  private vtkSwtComponent parent;
```

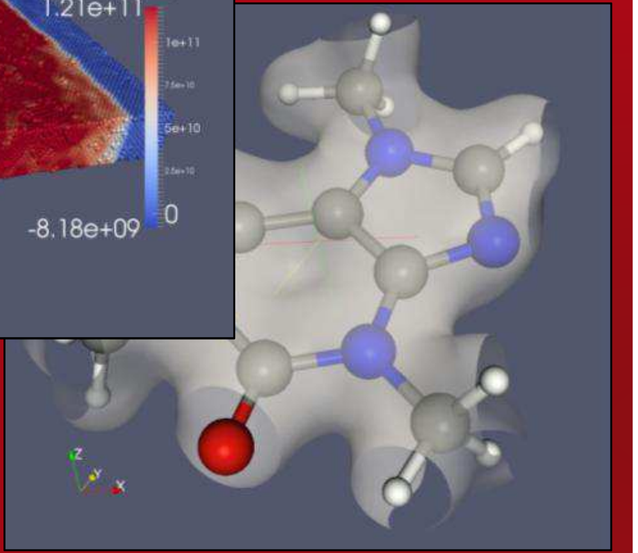


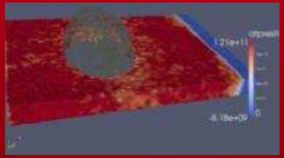
CONTRIBUTE TO OPEN SOURCE VISUALIZATION SOFTWARE

-

ANALYZING LARGE MOLECULAR DYNAMICS DATA SETS PRODUCED BY STAMP

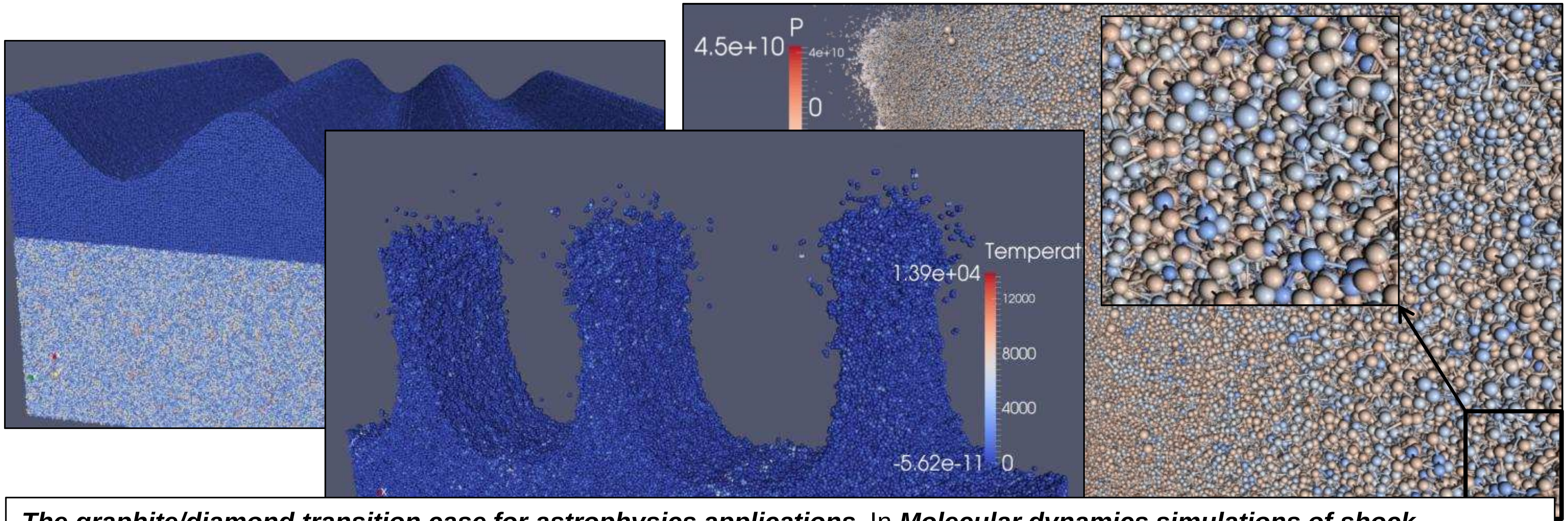






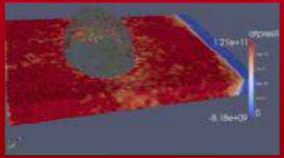
Stamp : molecular dynamics at large scale

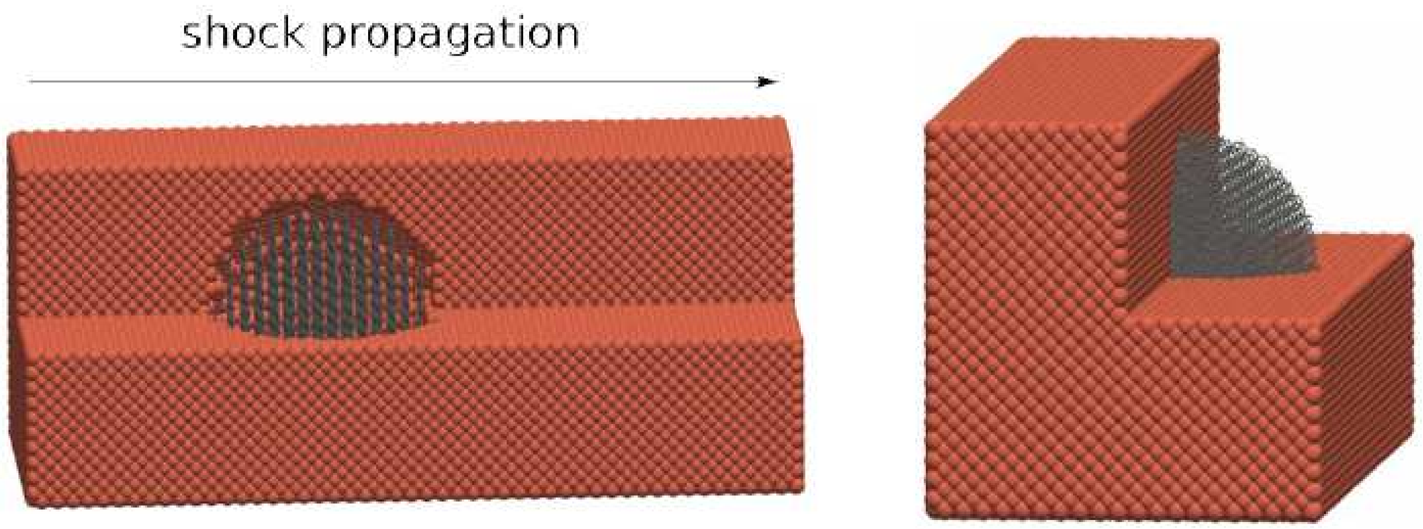
- Parallel molecular dynamics simulation code
  - Meshless, N-Body like simulation code
  - From a few millions up to a billion atoms
  - Enable accurate simulation of material properties at small scale
  - Complementary to usual hydrodynamics to estimate parameters of coarser grain physics models
  - Uses up to several dozen thousands of cores



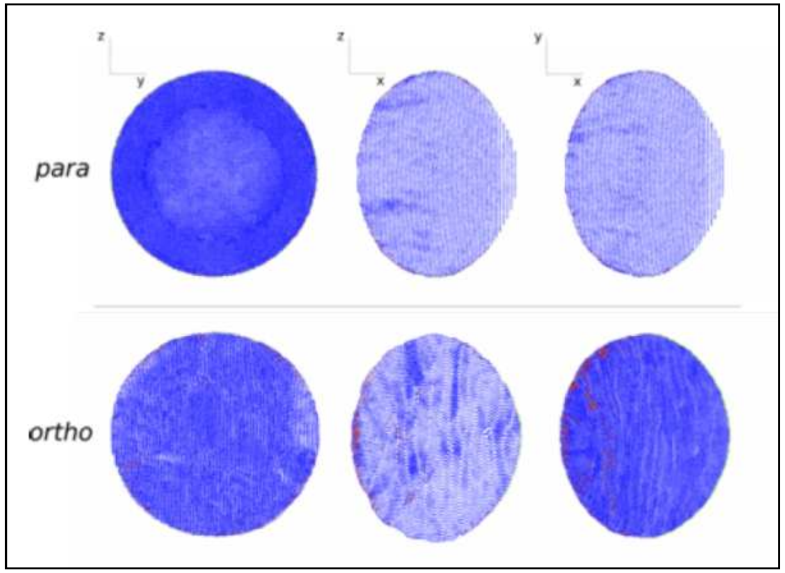
The graphite/diamond transition case for astrophysics applications. In ***Molecular dynamics simulations of shock compressed composite materials II.*** N. Pineau, L. Soulard, L. Colombet, T. Carrard, A. Pelle, Ph. Gillet, and J. Clerouin

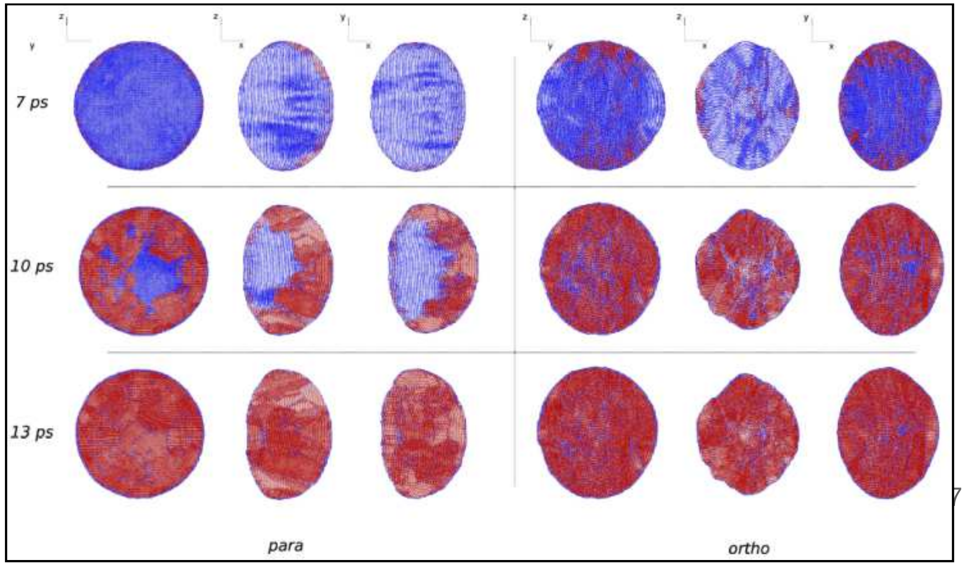
Shock compressed composite materials



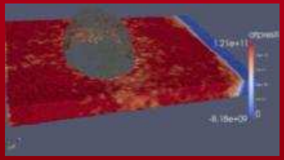


Graphite sheets orientation w.r.t. shock wave direction





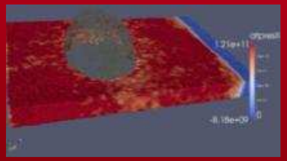
Exploring atoms & bonds at large scale



- What is available in the open source ?

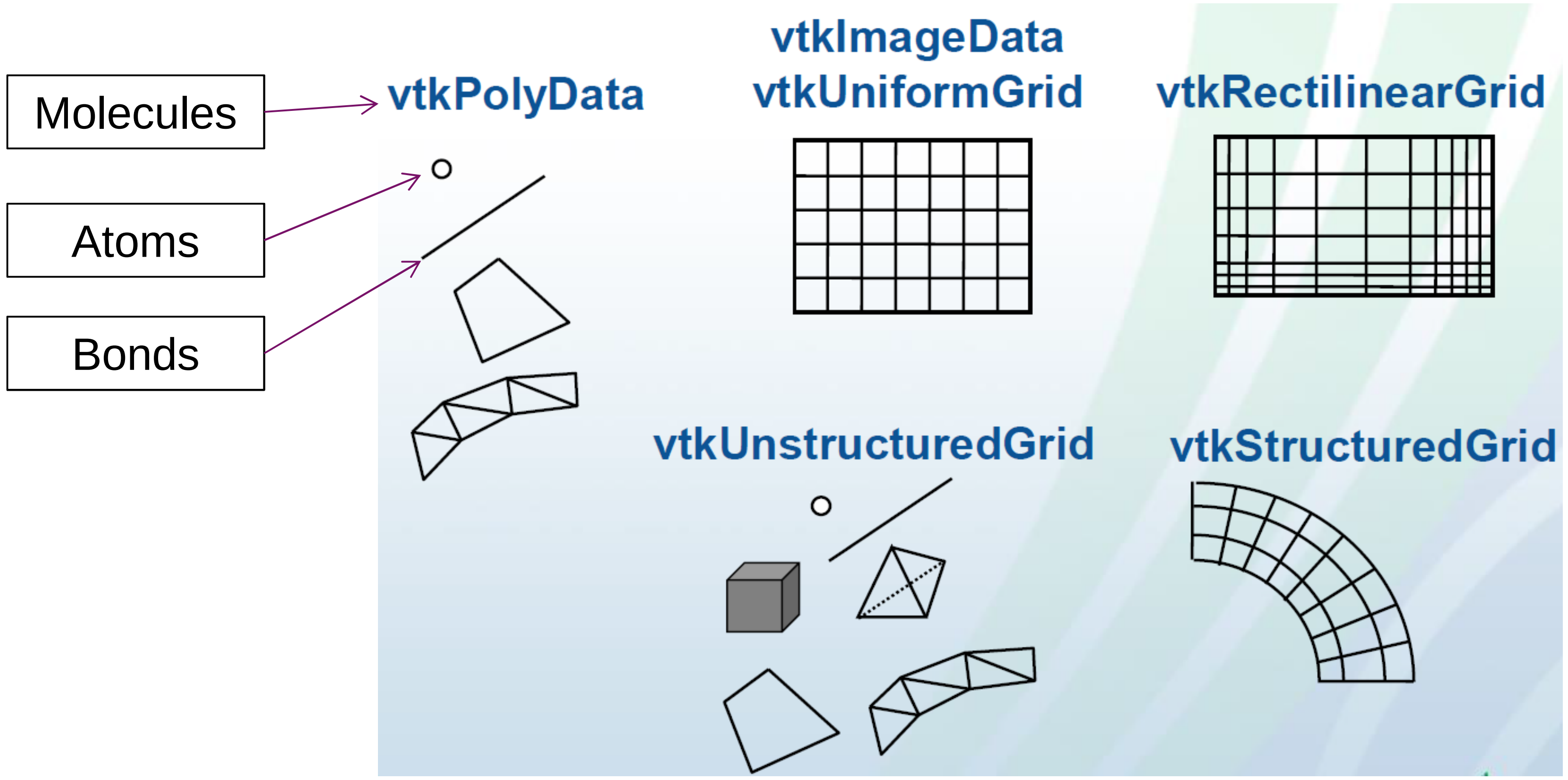
- Pre-requisites
  - → Fast atom and bonds rendering
  - → Parallel
  - → Atom bond computation
  - → basic crystallographic properties computation

- ParaView has
  - → Parallelism (processing and rendering)
  - → Fast rendering of sheres
  - → Many filters available to subset and explore large data sets

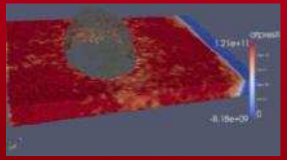


Data representation

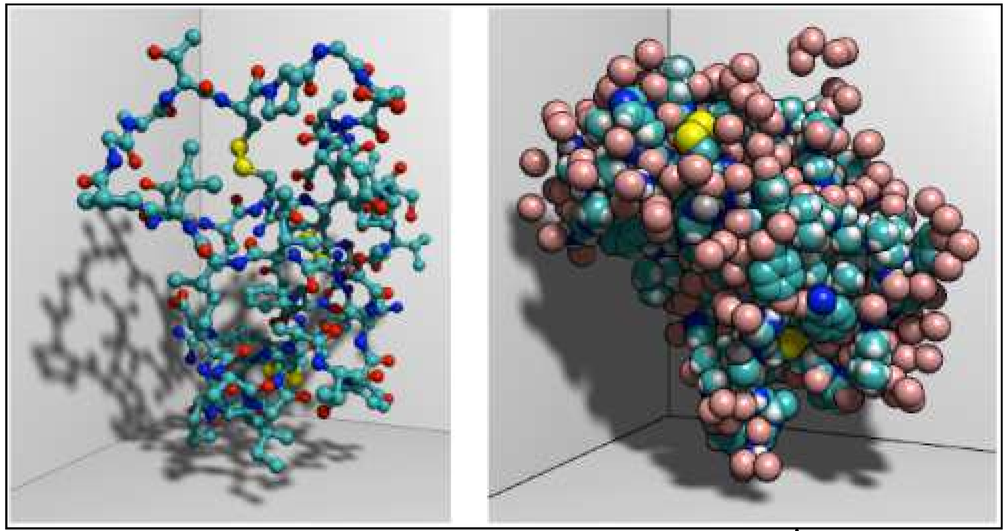
- Molecule data type available in VTK not suitable for high performance
- Very few existing algorithms may be applied
- Use a polygonal model where : 0D cells (vertex) are atoms and 1D cells (lines) are bonds



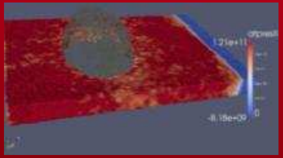
Rendering



- Geometry based glyphs
  - Average quality
  - Way too slow on large data sets



- Point Sprite plugin
  - Based on « GPU-Based Ray-Casting of Quadratic Surfaces” (EGPBG’06)
  - Contributed by EDF and CSCS, available in ParaView source tree.
  - perspective correctness issue.
  - Difficult to mix different types of quadrics (spheres / cylinders)
  - Sub-optimal OpenGL binding
  - Lack specific molecular visualization feature



Rendering

Implementation details

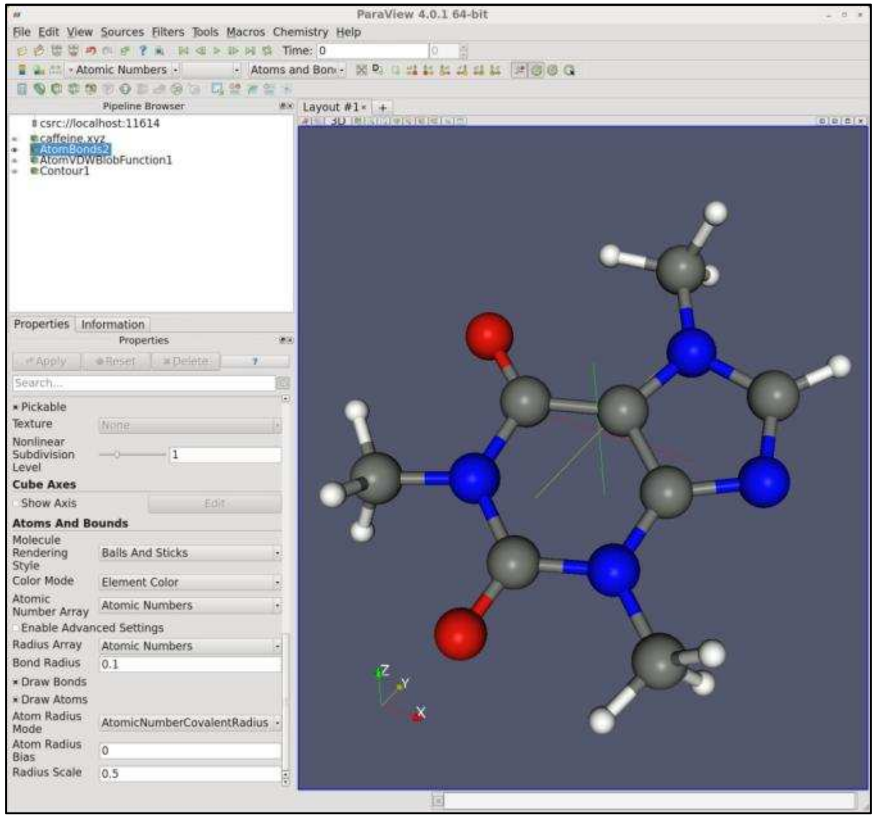
- Generalized Quadrics

$$\mathbf{x}^T\mathbf{Q}\mathbf{x} = 0 \qquad \mathbf{Q} = \begin{bmatrix} A & B & C & D \\ B & E & F & G \\ C & F & H & I \\ D & G & I & J \end{bmatrix}$$

- Reduced variability to reduce GPU bandwith: 3 parameters only for each quadrics $\underset{C}{\rightarrow}, \underset{N}{\rightarrow}, r$ converted to conic matrix$Q$.
- Recent GPU capabilities make development simpler : arbitrary matrix inverse, const arrays for inline lookup tables, etc.

Conclusion

- Benefit from existing code sniplets
- Integration in a feature rich, parallel platform
- Enable visualization oflarge scale molecular AND grid based data sets in the same tool

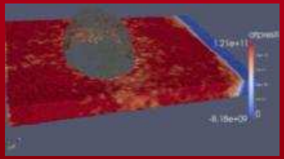


csrc://localhost:11614
caffeine.xyz
AtomBonds2
AtomVDWBlobFunction1
Contour1

Properties

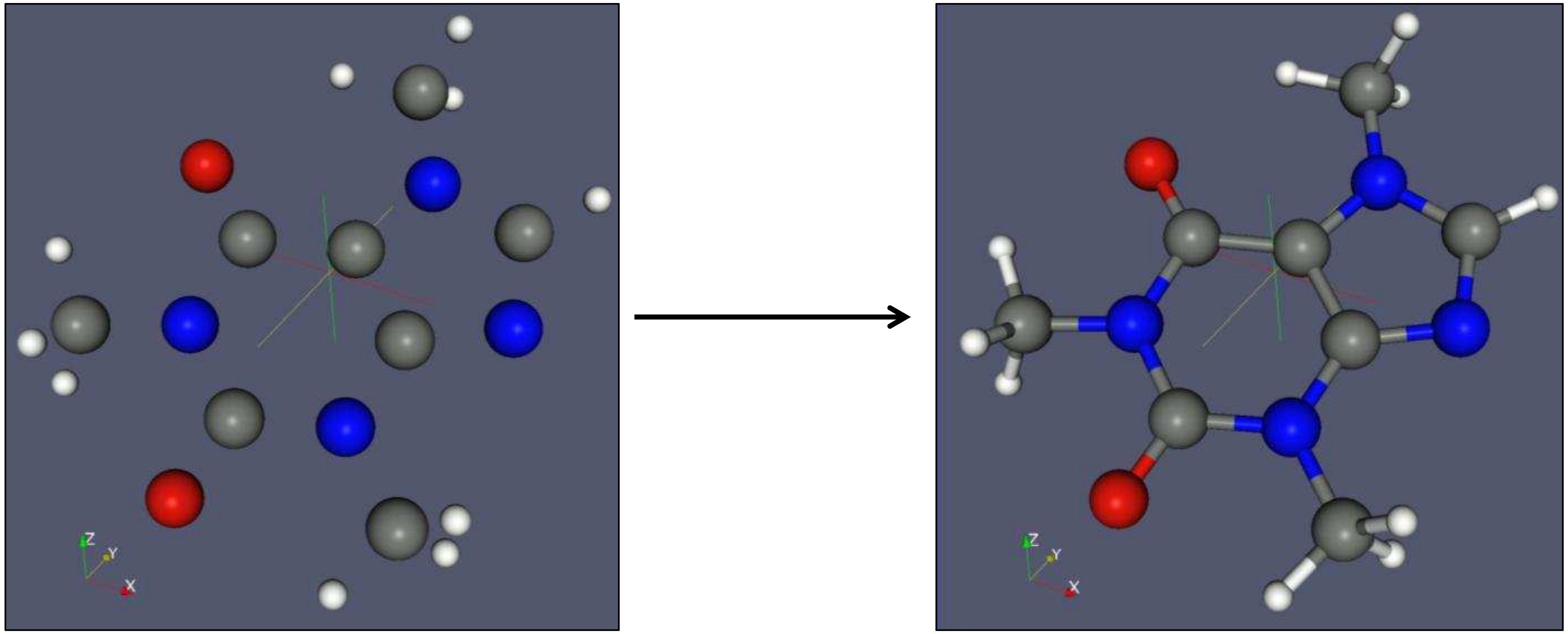
- Pickable
- Texture: None
- Nonlinear Subdivision Level: 1
- **Cube Axes**
- Show Axis
- **Atoms And Bounds**
- Molecule Rendering Style: Balls And Sticks
- Color Mode: Element Color
- Atomic Number Array: Atomic Numbers
- Enable Advanced Settings
- Radius Array: Atomic Numbers
- Bond Radius: 0.1
- Draw Bonds
- Draw Atoms
- Atom Radius Mode: AtomicNumberCovalentRadius
- Atom Radius Bias: 0
- Radius Scale: 0.5

Algorithms



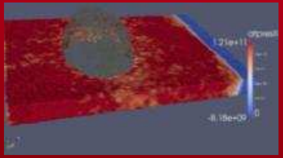
Atom bonds generation filter

- Based on proximity with respect to covelent radius

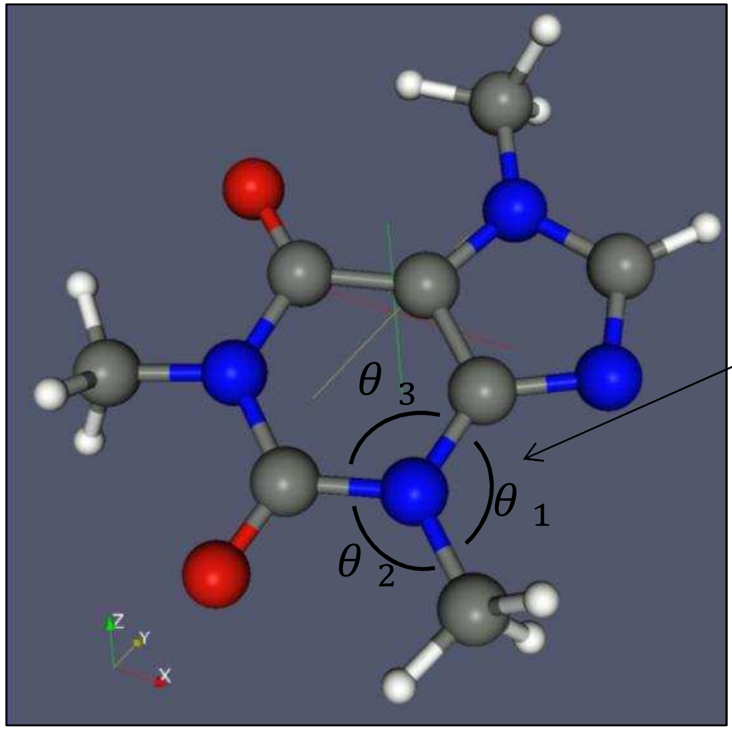


DE LA RECHERCHE À L'INDUSTRIE

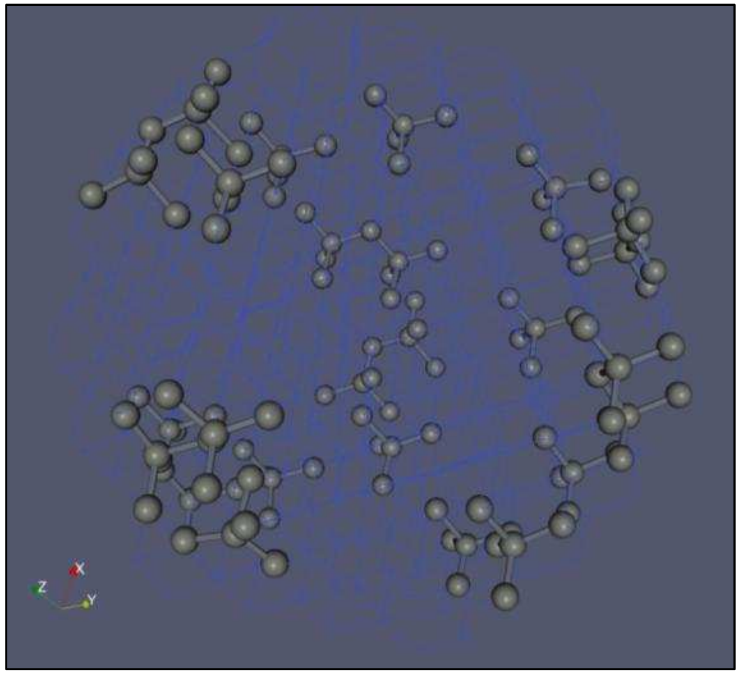
Algorithms



- Dihedral angle statistics to discriminate crystallographic structures

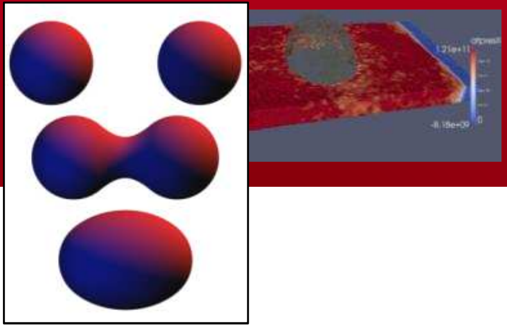


Based on statistics about dihedral angle between bonds of a single atom.
i.e. minimum or maximum dihedral angle

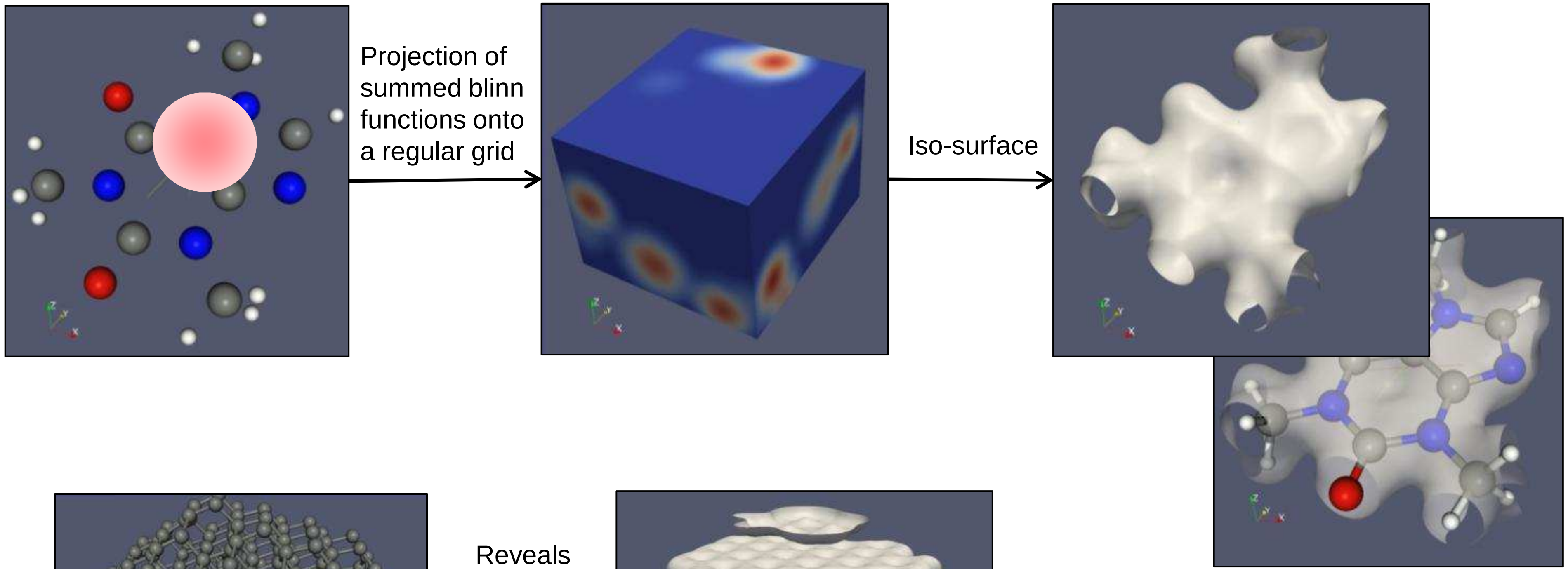


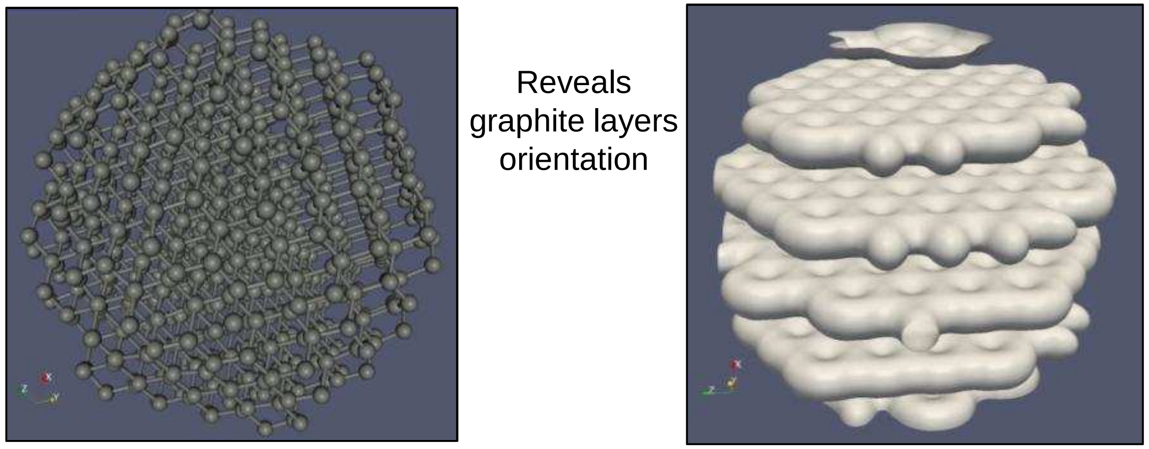
Discriminate a subset of carbon structure.
i.e. where diamonds replace graphite

Algorithms

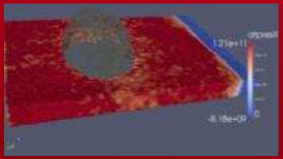


- Blinn’s blob function splatting on regular grid
- Isosurface produce pseudo surface surrounding atoms





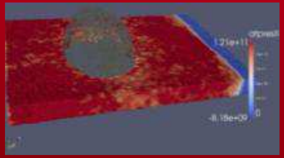
Paraview plugin

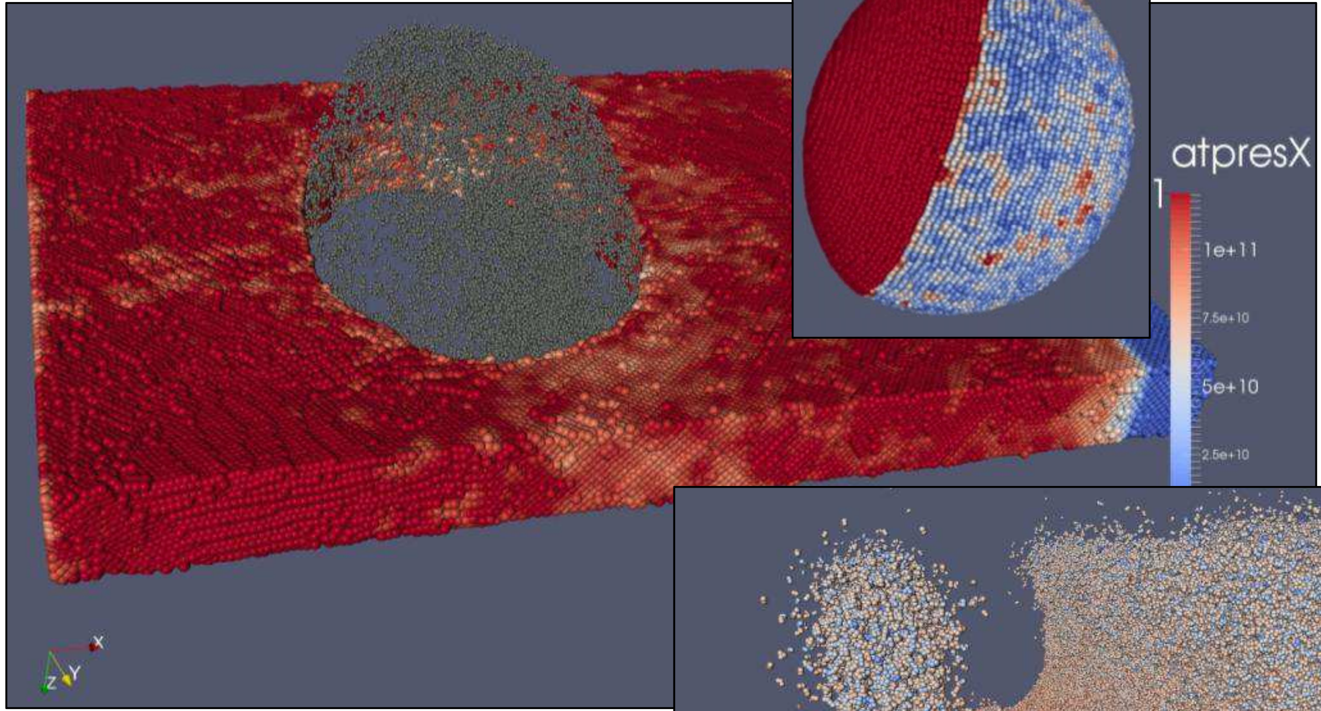


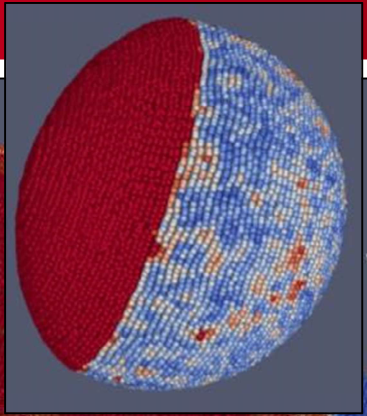
- ChemiSpriteFilters
  - Atom bonds computation
  - Blob splatting
  - Bond dihedral angle statistics

- ChemiSprite
  - VTK painter’s extension
    - ✓ turn points and lines to raytraced spheres and tubes
  - Representation
    - ✓ Map high level properties to low-level painter configuration
  - ParaView plugin
    - ✓ Makes representation availbale in ParaView’s representation dropdown menu

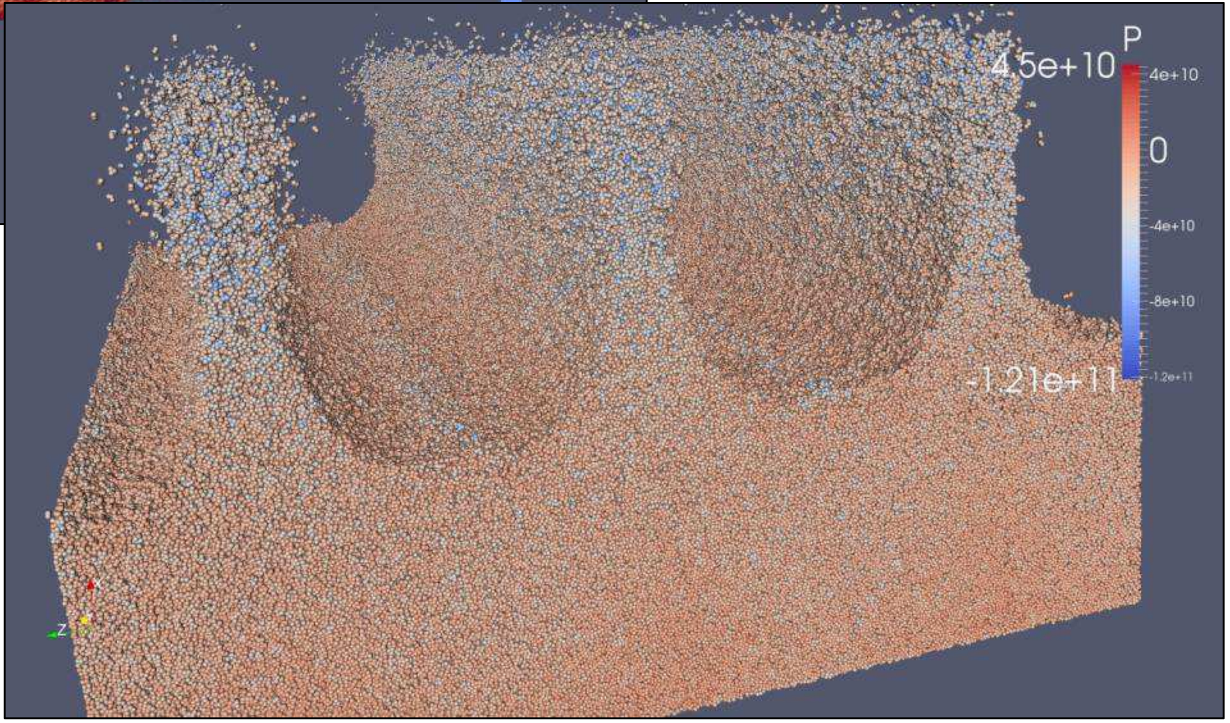
Results







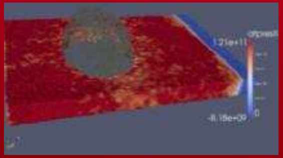
Diamond found in meteorite pieces explained through simulation



Understanding instabilities at atom level

DE LA RECHERCHE À L'INDUSTRIE

Results



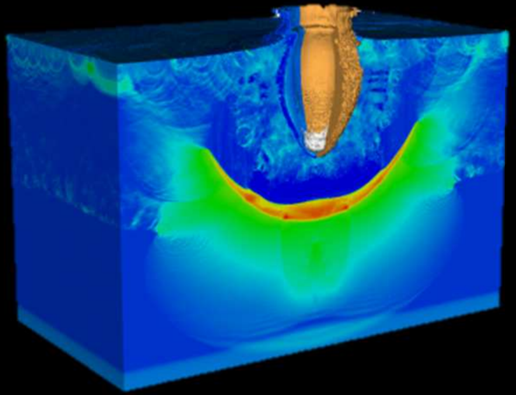
- A good case of successful interoperation between open source platform and domain specific developments

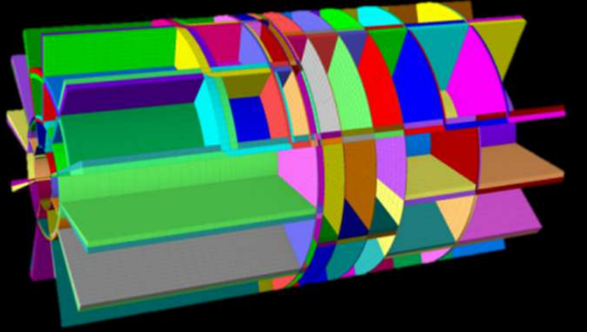
- Achieved good performance and scalability, thanks to ParaView's parallel architecture
  - → atom bond parallel scalability has to be improved

- Tested with over 1 billion atoms (about 500 CPUs and GPUs)

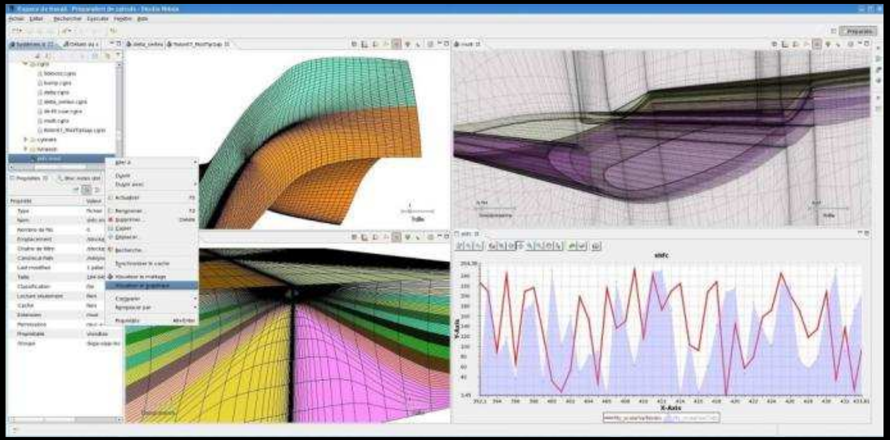
- Additional work needed to make this available in source tree

DE LA RECHERCHE À L'INDUSTRIE





Thank you !

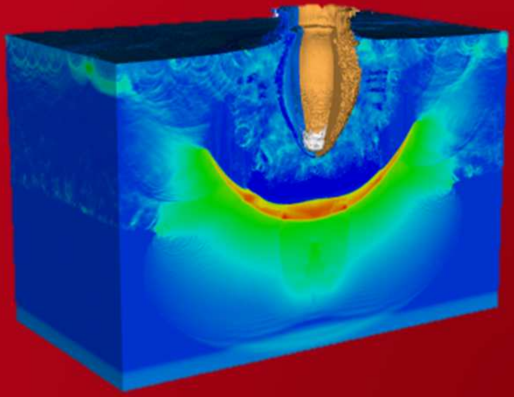


www.cea.fr

Fabien VIVODTZEV ***Thierry CARRARD***

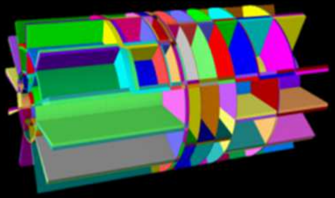
DE LA RECHERCHE À L'INDUSTRIE

cea



www.cea.fr

Open Source Software to Visualize Complex Data on Remote CEA's Supercomputing Facilities



Fabien Vivodtzev, Thierry Carrard
CEA (French Alternative Energies and Atomic Energy Commission)

In order to guaranty performances of complex systems using numerical simulation, CEA is performing advanced data analysis and scientific visualization with open source software using High Performance Computing (HPC) capability. The diversity of the physics to study produces results of growing complexity in terms of large-scale, high dimensional and multivariate data. Moreover, the HPC approach introduces another layer of complexity by allowing computation amongst thousands of remote cores accessed from sites located hundreds of kilometers away from the computing facility.

This paper presents how CEA deploys and contributes to open source software to enable production class visualization tools in a high performance computing context. Among several open source projects used at CEA, this presentation will focus on Visit, VTK and Paraview.

In the first part we will address specific issues encountered when deploying VisIt and Paraview in a multi-site supercomputing facility for end-users. Several examples will be given on how such tools can be adapted to take advantage of a parallel setting to explore large multi-block dataset or perform remote visualization on material interface reconstructions of billions of cells. Then, the specific challenges faced to deliver Paraview's Catalyst capabilities to end-users will be discussed.

In the second part, we will describe how CEA contributes to open source visualization software and associated software development strategy by emphasizing on two recent development projects. The first is an integrated simulation workbench providing plugins for every step required to achieve numerical simulation independently on a local or a remote computer. Embedded in an Eclipse RCP environment, VTK views allow the users to perform data input using interaction or mesh preview before running the simulation code. Contributions to VTK have been made in order to smoothly integrate these technologies. The second details how recent developments at CEA have helped to visualize and to analyze results from ExaStamp, a parallel molecular dynamics simulation code dealing with molecular systems ranging from a few millions up to a billion atoms. These developments include a GPU intensive rendering method specialized for atoms and specific parallel algorithms to process molecular data sets.