JARA|HPC

In-Situ Visualization of Large-Scale Data

on the example of 'VisIt' for turbulence research

Guest Talk – SimLab Seminar 23.06.2015

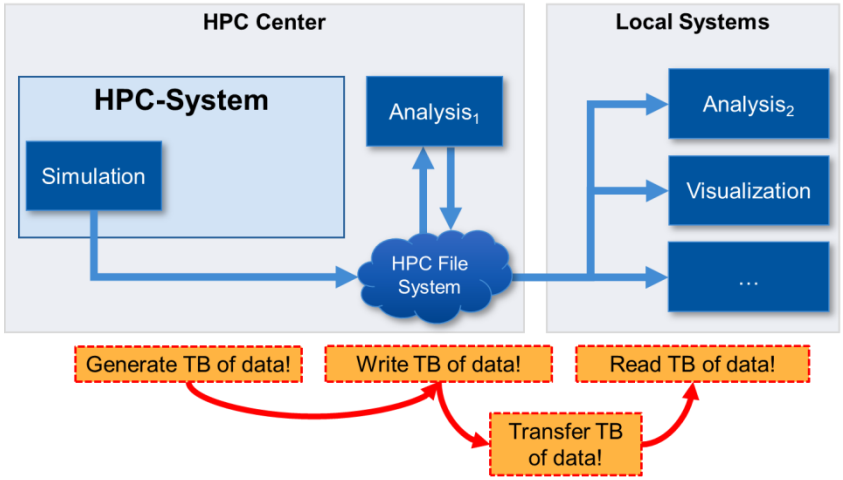
Jens Henrik Göbbert

AN INITIATIVE OF
RWTH AACHEN UNIVERSITY
JÜLICH FORSCHUNGSZENTRUM

Motivation

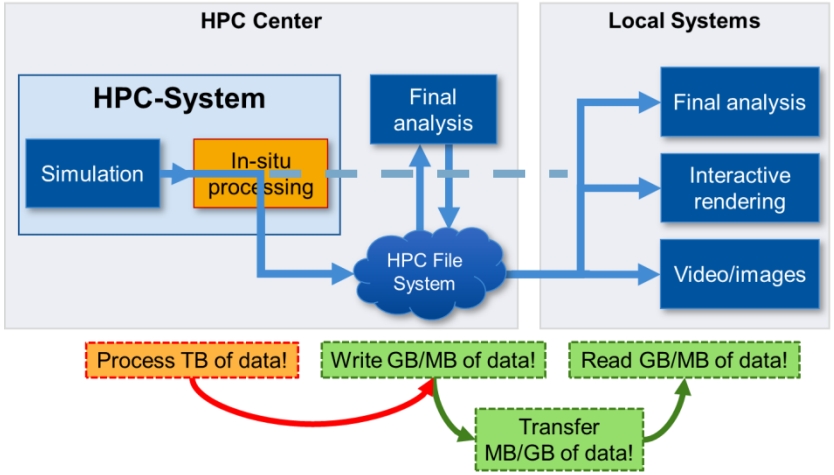
In-Situ Visualization

- simulation data is rapidly growing
- time for post-processing becomes a major bottleneck



In-Situ strategies

- provide a mid- to long-term solution for in-situ processing of large-scale data
- simplify coupling of simulation and in-situ processing



Codes

- psOpen (ITV, RWTH Aachen University)
- CIAO (ITV, RWTH Aachen University)
- ZFS (AIA, RWTH Aachen University)
- ...

Outline

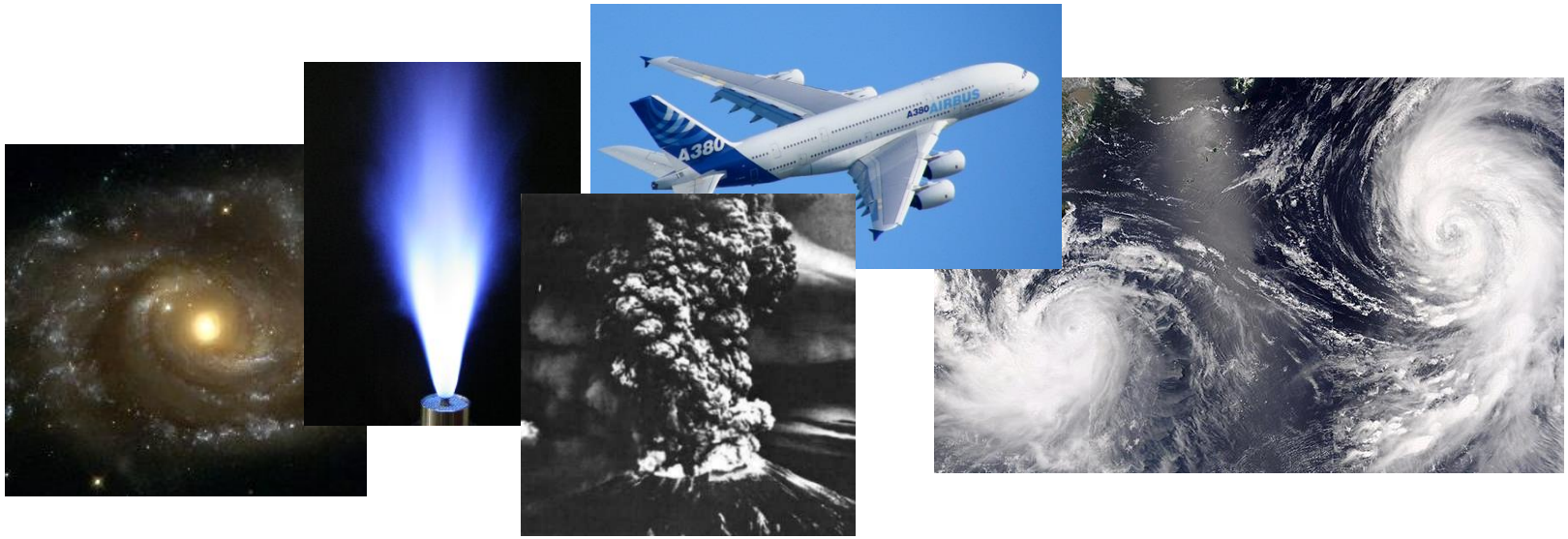
- Turbulence – a source for large-scale simulation data
  - motivation , method, simulations, analysis
- In-Situ Visualization
  - approaches, tools
- VisIt integration in scientific applications
  - details, source, outlook

Turbulence Research …
a source for large-scale simulation data

Scientific Embedding

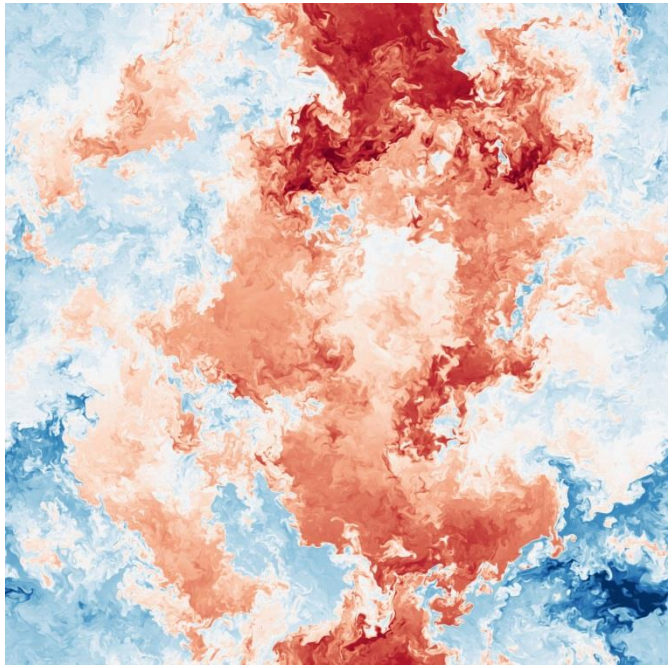
Turbulence

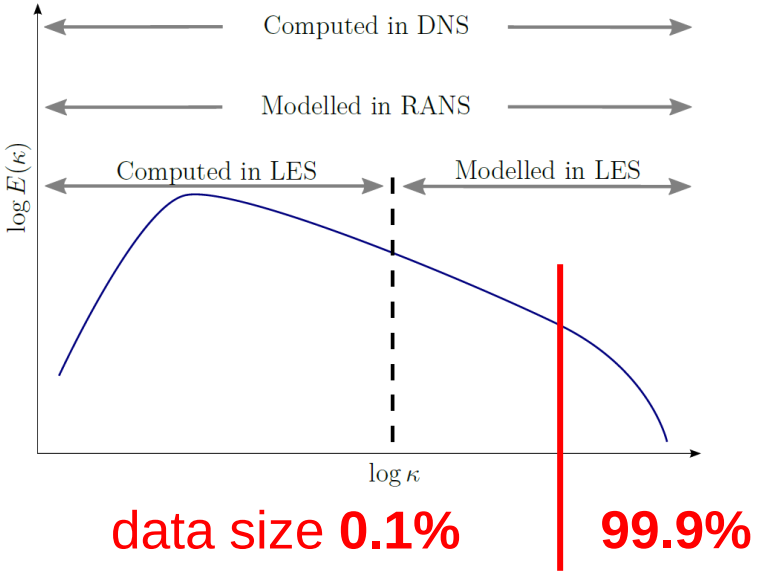
phenomenologically a fluid regime characterized by chaotic and stochastic property changes



Large-Scale Data

Why „large-scale data“ ?

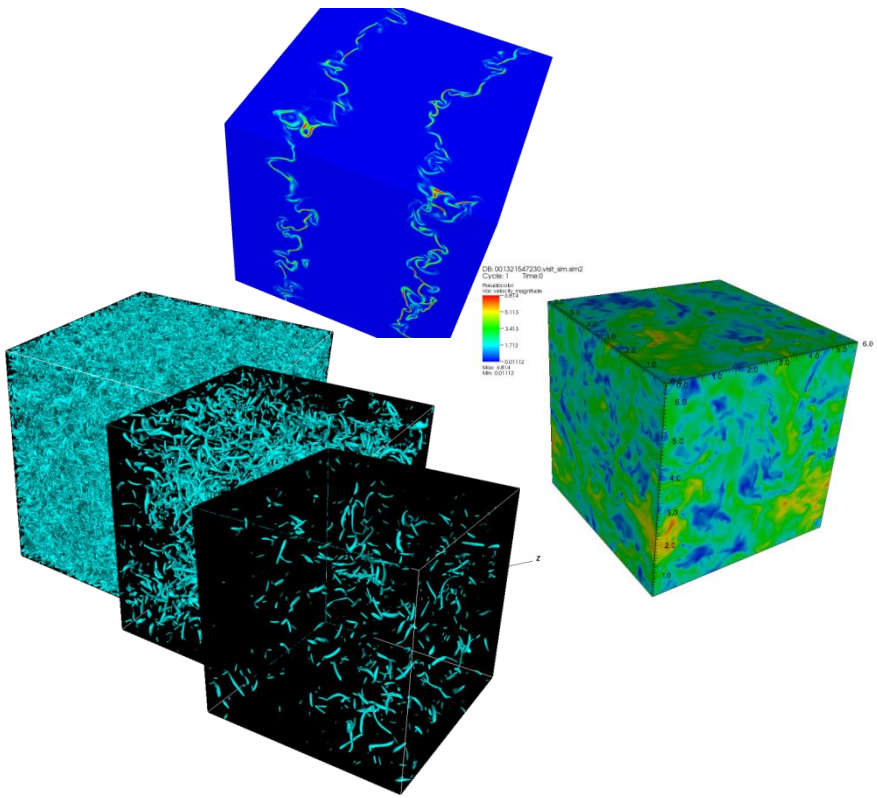


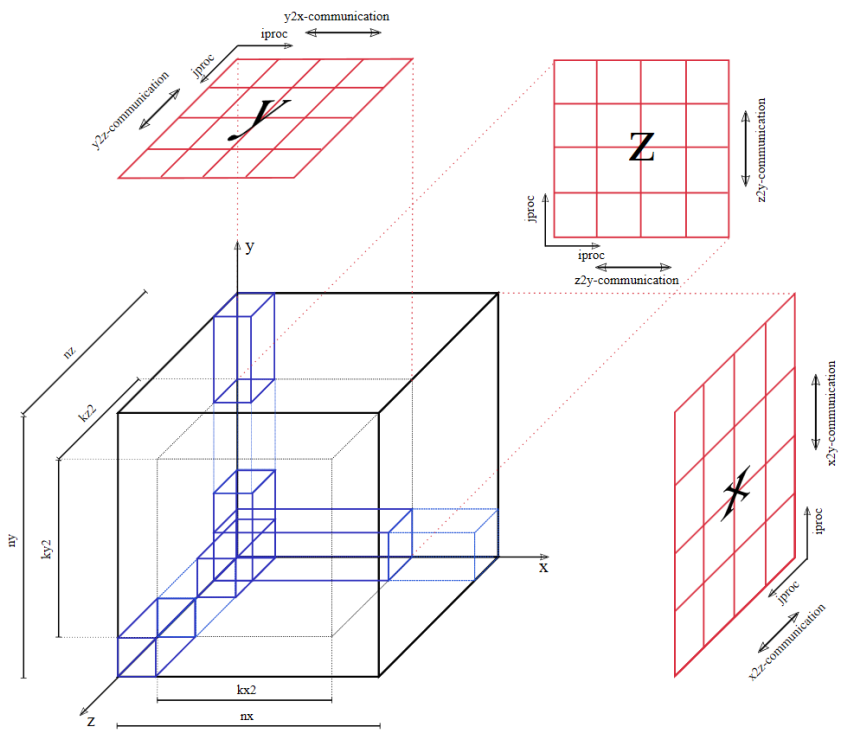


Simulation Method

psOpen – DNS by a pseudo-spectral approach

- hybrid OpenMP/MPI
- Fortran90



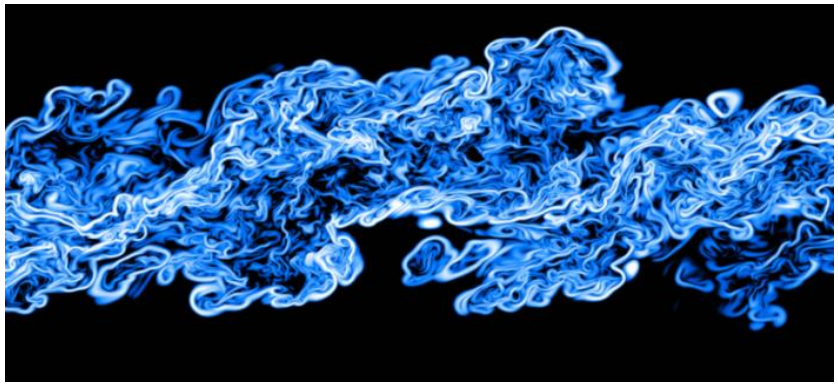


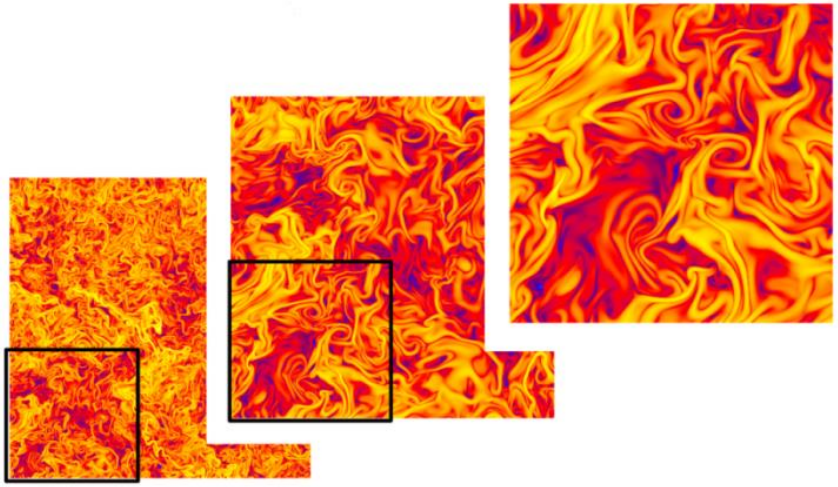
http://www.fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/psOpen/_node.html

Large-Scale Simulations

Direct Numerical Simulation (DNS)

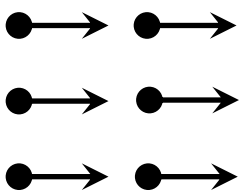
- turbulent flow and mixing
- for analysis of fine scale structures
- simulations on JUQUEEN with
  - up to 240 billion grid points
  - up to full machine possible (High-Q club)

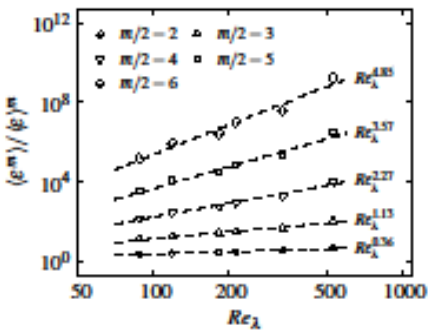




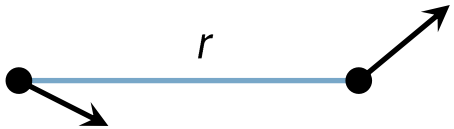
Data Analysis (1)

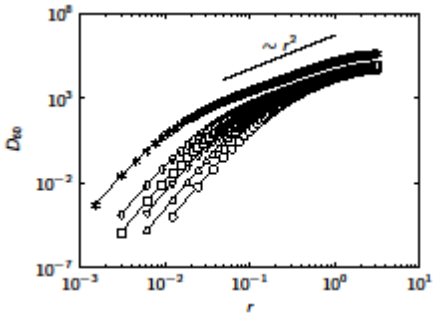
1-point statistics



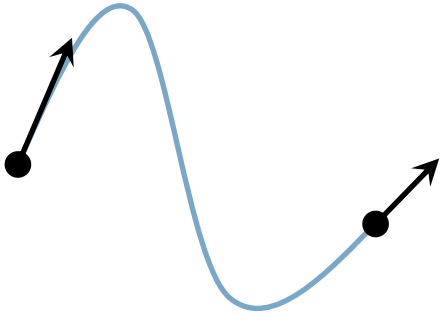


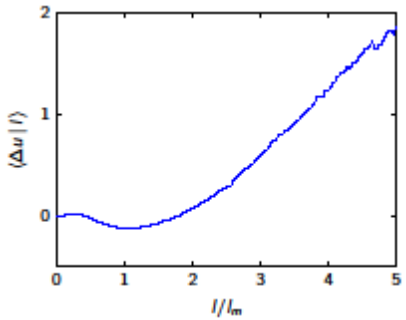
2-point statistics





statistics along path lines

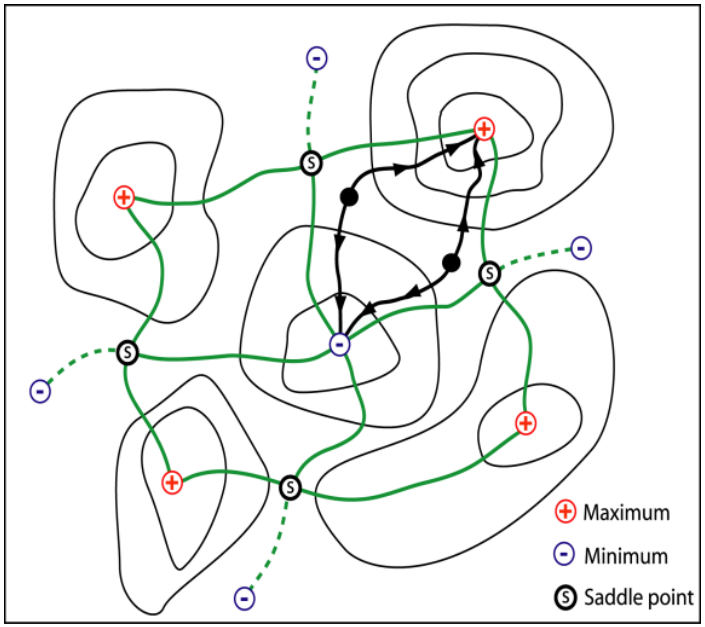


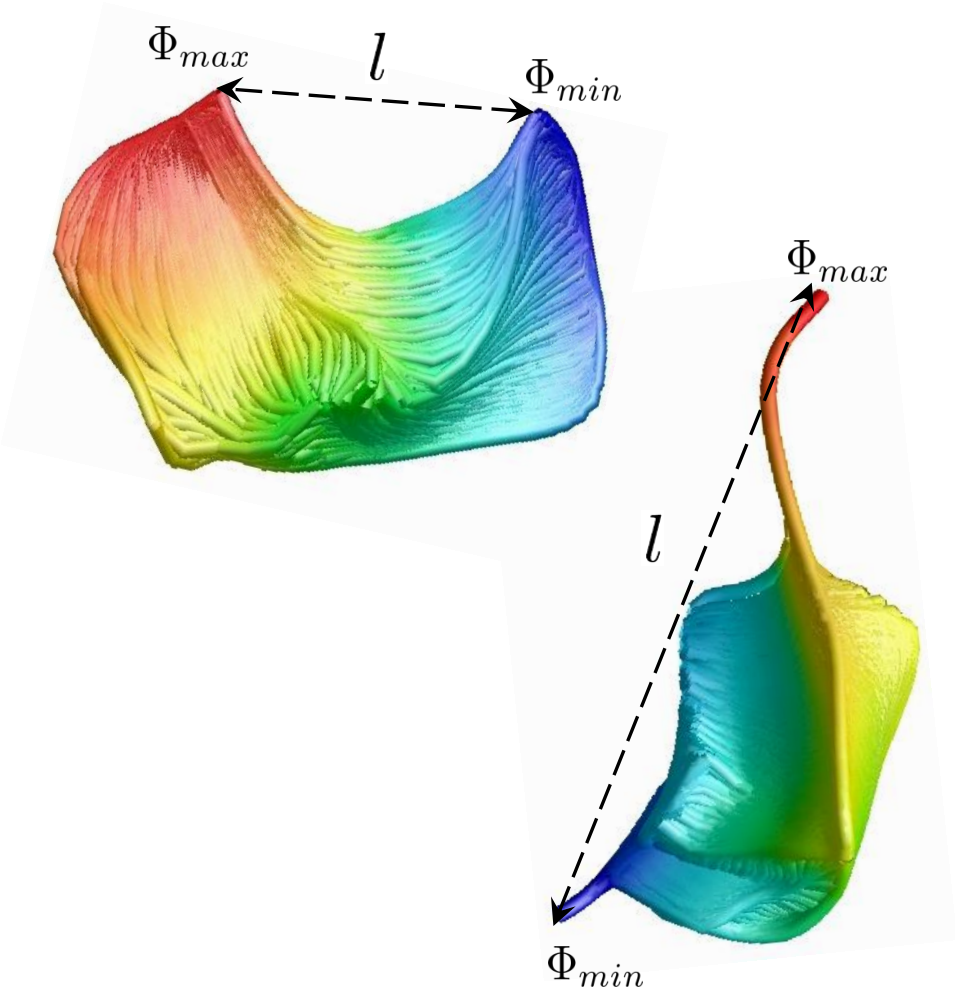


Data Analysis(2)

Dissipation elements

- ensemble of grid points from which the same extreme points are reached
- non-arbitrary and space-filling
- statistical description by two parameters: scalar difference and linear length

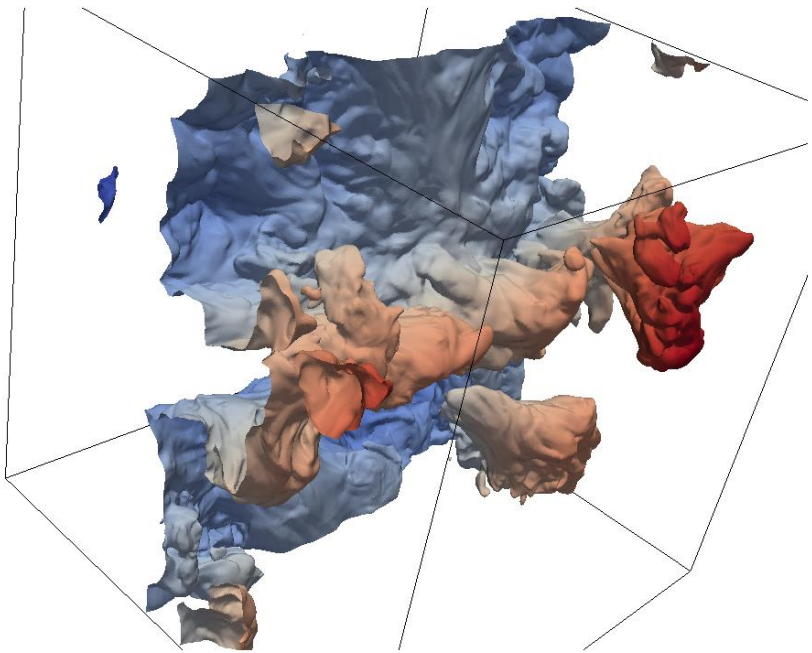


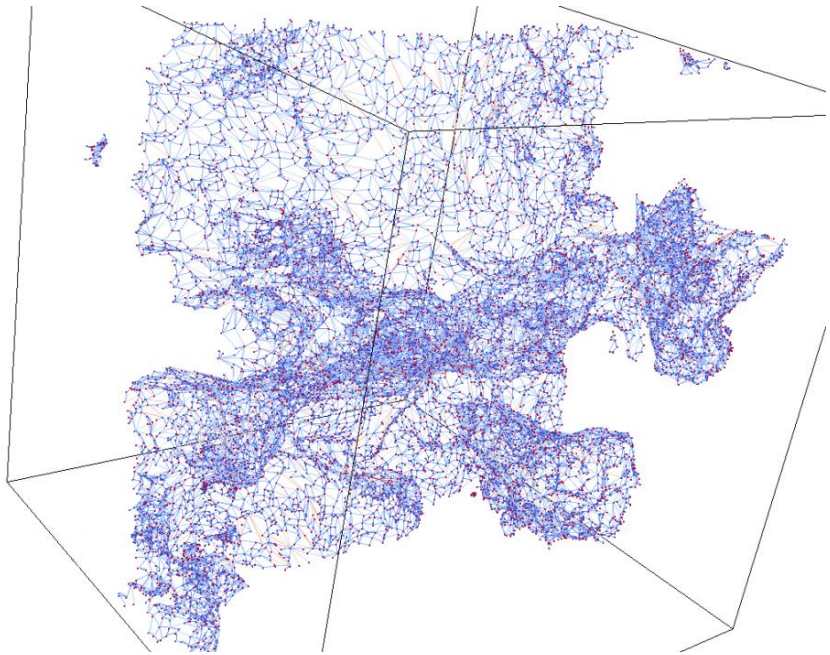


Data Analysis (3)

Premixed Flames

- segmentation of flame surface
- non-arbitrary and surface-filling





Turbulence Data Base

	A0	A1	A2	A3	A4	A5	A6
N^3	512^3	1024^3	1024^3	2048^3	2048^3	4096^3	4096^3
Re_λ	88	119	184	215	331	529	754
file size (GB)	8	64	64	512	512	4096	4096
M	180	60	60	10	10	10	10
data size (TB)	1.44	3.81	3.81	5	5	22	22

	B0	B1	B2	B3	B4	B5	B6
N^3	720^3	1440^3	1440^3	2816^3	2816^3	5632^3	6144^3
Re_λ	84	115	173	207	297	529	770
file size (GB)	22	177	177	1331	1331	5324	6912
file size compressed (GB)	6.6	52.6	52.6	393.2	393.2	1572.8	2041.9
M	40	20	20	10	10	5	5
data size (TB)	0.88	3.54	3.54	13.3	13.3	22.4	34.5
data size compressed (TB)	0.26	1.05	1.05	3.9	3.9	6.6	10.3

In-Situ Visualization …

Motivation

In-situ Visualization

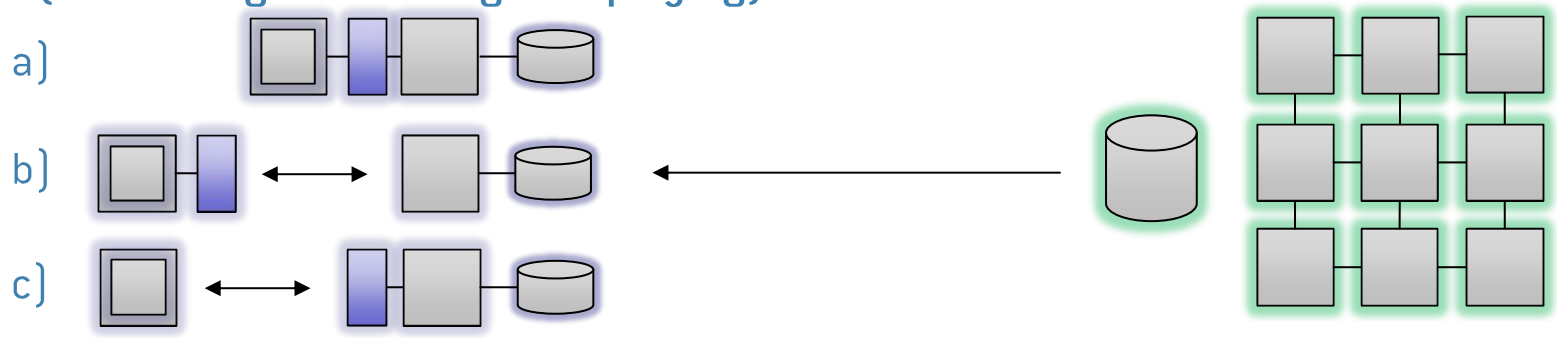
- easily check any running simulation
- …

- eliminate I/O to and from storage
- eliminate data transfer from compute cluster to visualization cluster
- direct access to all time steps, all variables … while simulation is running

Classical Visualization

Visualization
(Processing + Rendering + Displaying)

Simulation

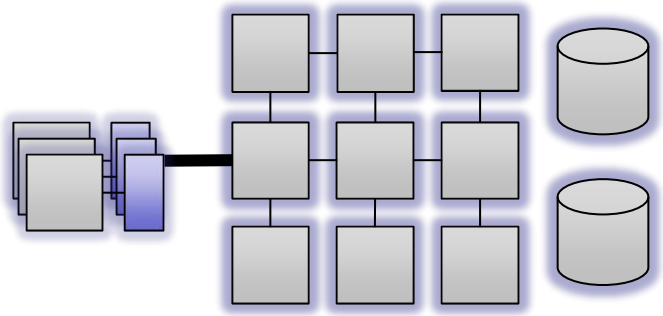


1. compute
2. dump data to disk
3. copy data through network
4. load data from disk
5. visualize

In-Situ Visualization

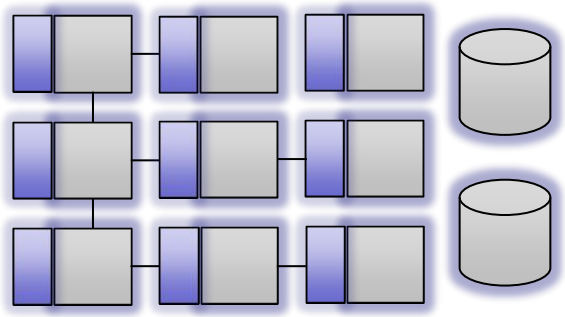
Staged In-Situ Visualization

- simulation and visualization on **different** nodes
  - + sim./vis. parallelization strategies can differ
  - + sim./vis. required resources can differ
  - - move data between compute and staging nodes
  - - compute nodes might wait while visualization



On-Node In-Situ Visualization

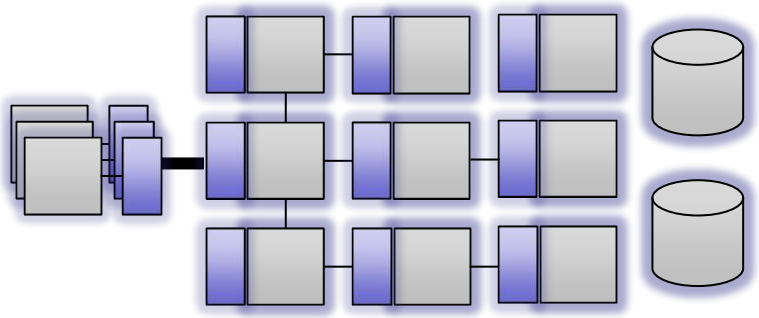
- simulation and visualization on **same** nodes
  - + no/less data movement required
  - + no/less compute resources wasted
  - - sim./vis. should have same parallelization strategy
  - - sim./vis. share same compute resources



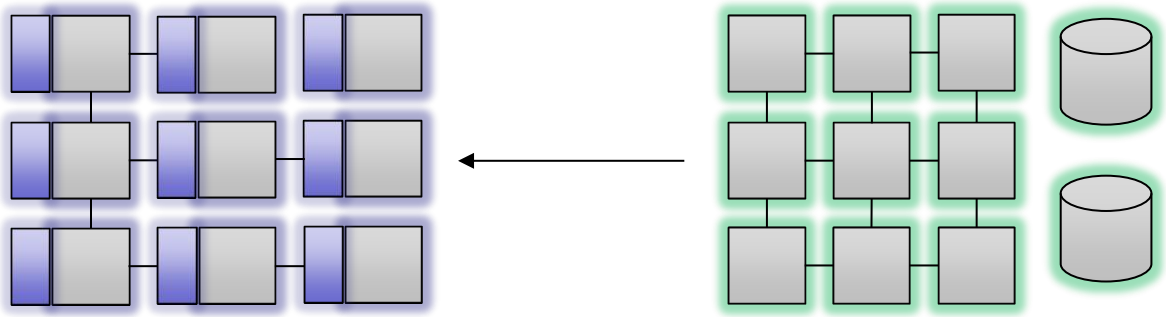
In-Situ Visualization

Hybrid (On-Node + Staged) In-Situ Visualization

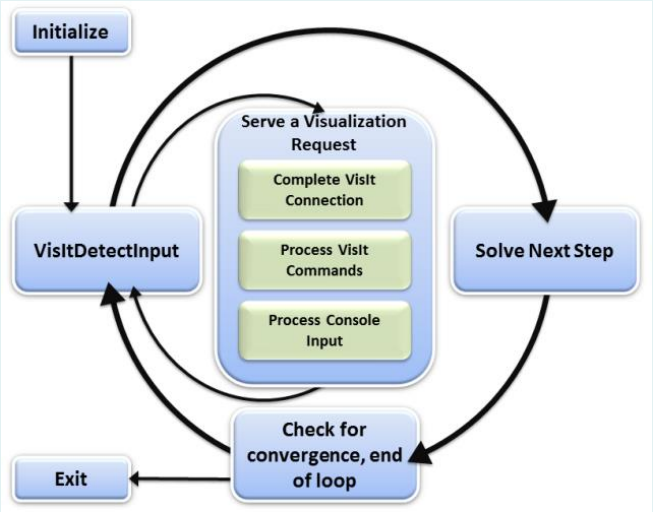
- simulation and on compute nodes
  and visualization on compute + staging nodes

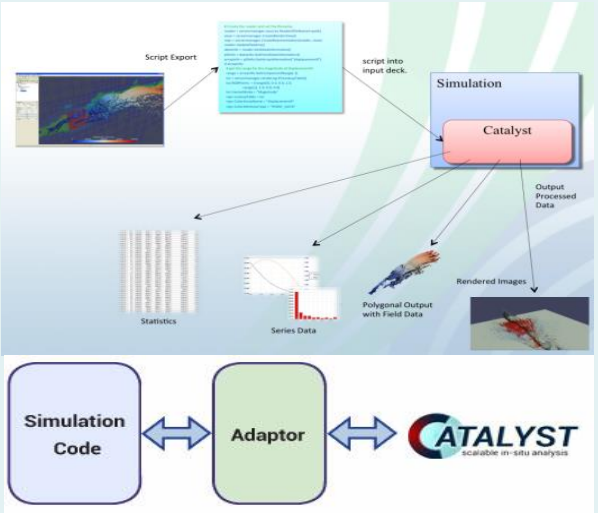


In-Situ Visualization coupling two clusters



Visualization Software





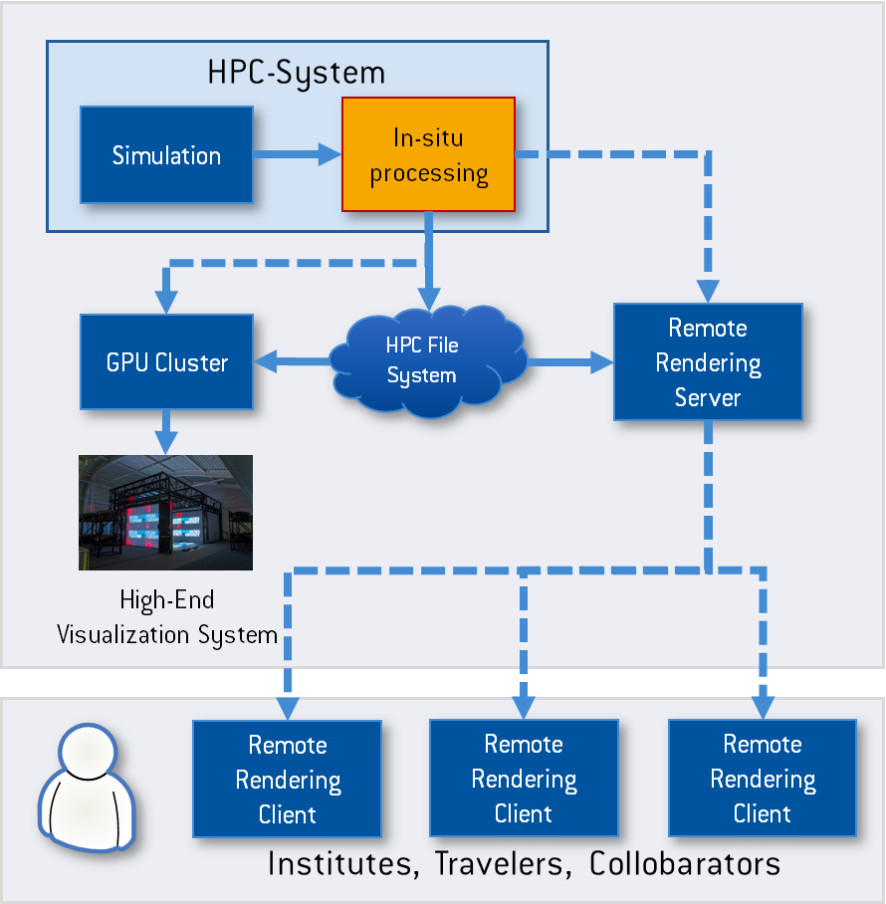
VisIt

developed by multiple national US labs
funded by Department of Energy (DOE)
initial release 2002
hosted at LLNL
open source (BSD)

ParaView

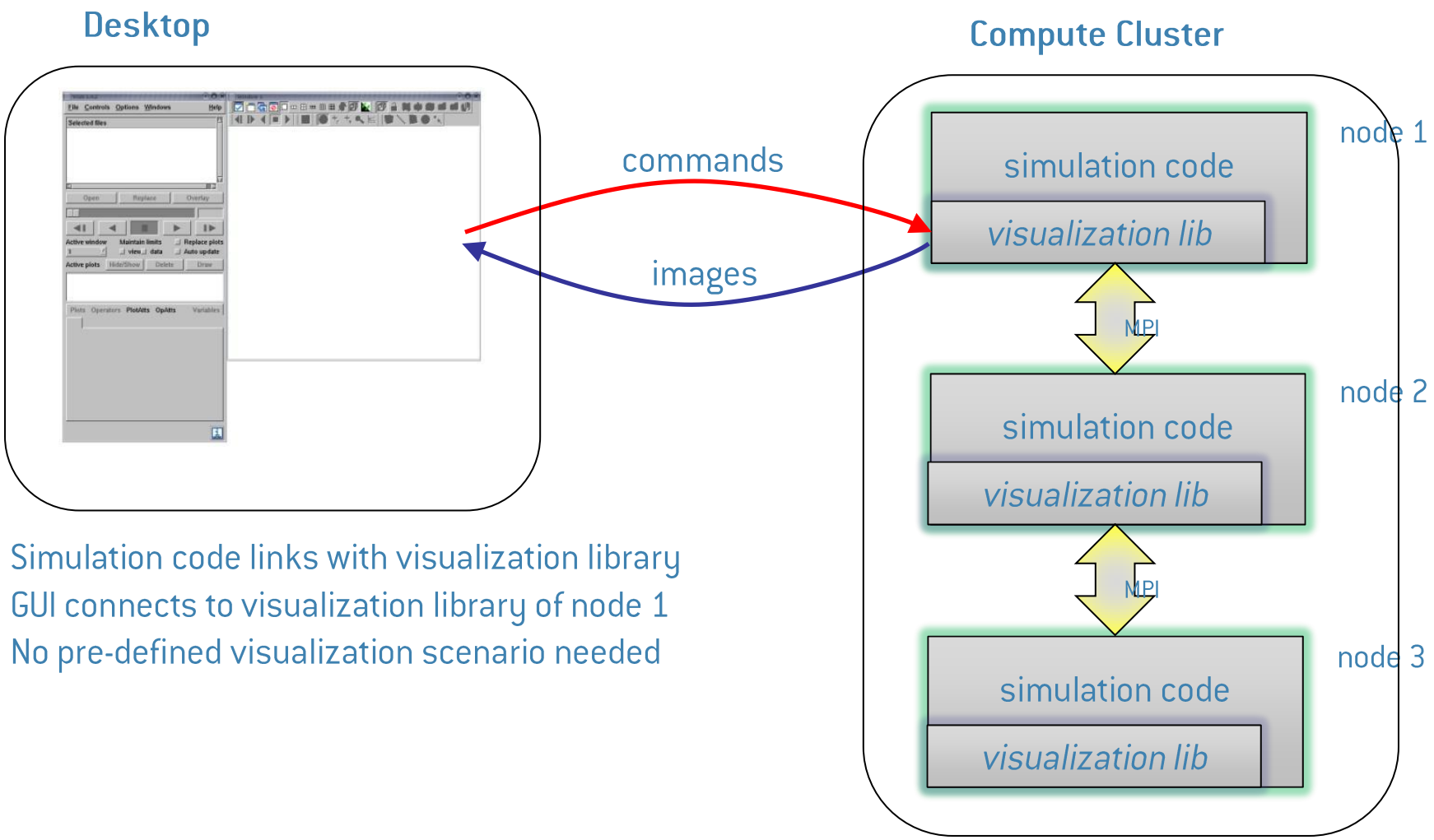
developed by Kitware + Los Alamos National Lab
funded by Department of Energy (DOE)
initial release 2002
hosted by Kitware
open source (BSD)

In-Situ Visualization and Remote Rendering



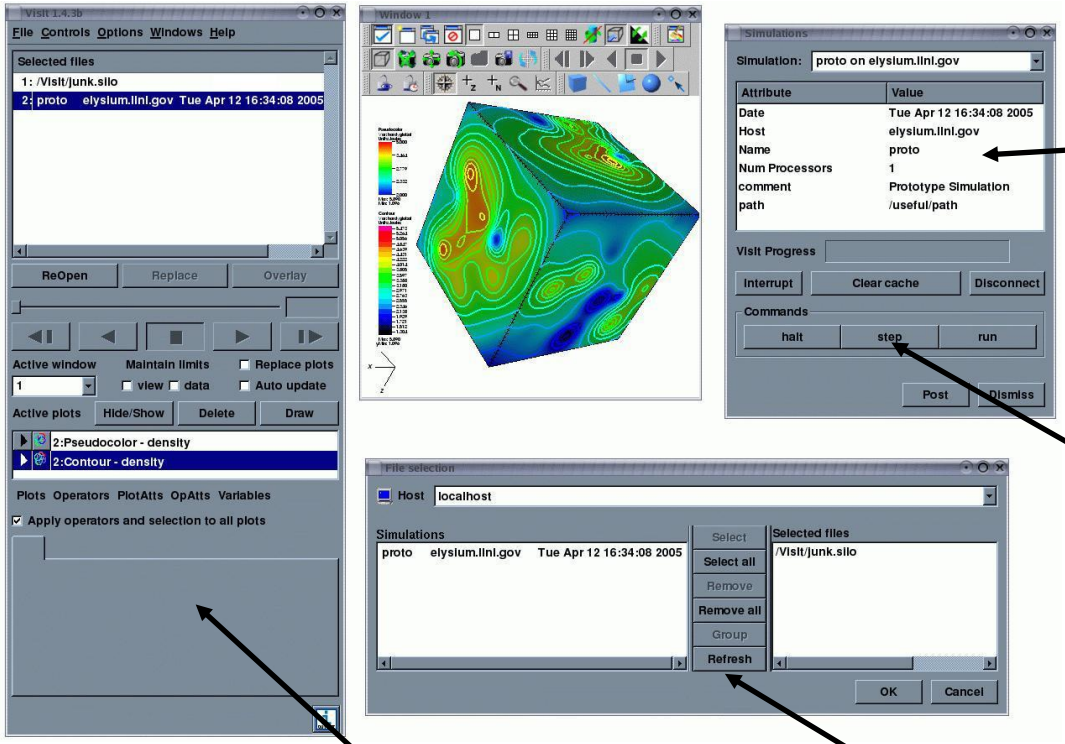
VisIt integration in scientific applications

In-Situ Visualization with VisIt



- Simulation code links with visualization library
- GUI connects to visualization library of node 1
- No pre-defined visualization scenario needed

In-Situ Visualization with VisIt



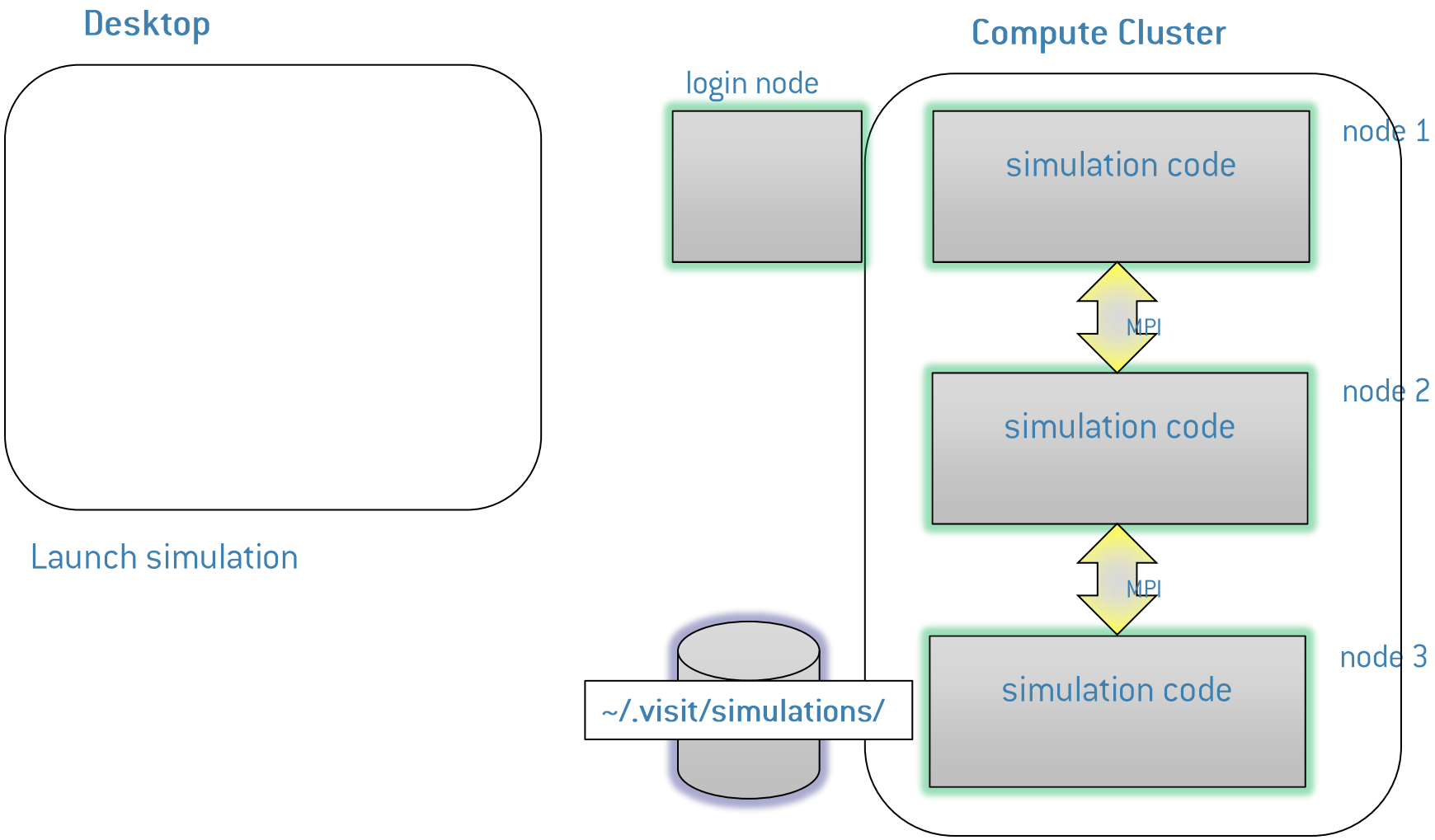
The simulation windows shows the meta-data about the running code

Control commands defined by the simulation code accessible here.

All VisIt existing functionality is accessible.

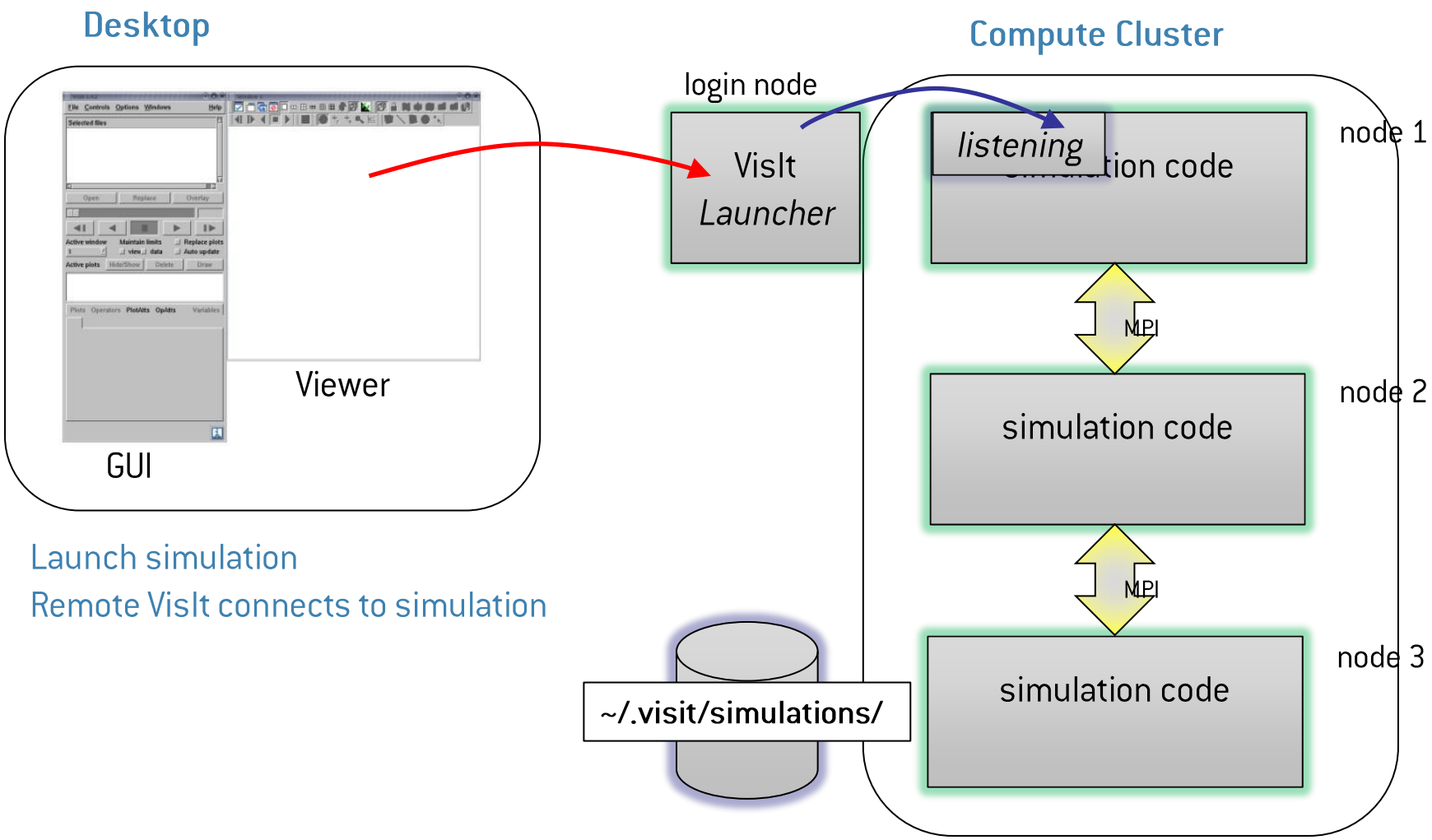
User selects running simulations to connect to as if they were files.

In-Situ Visualization with VisIt



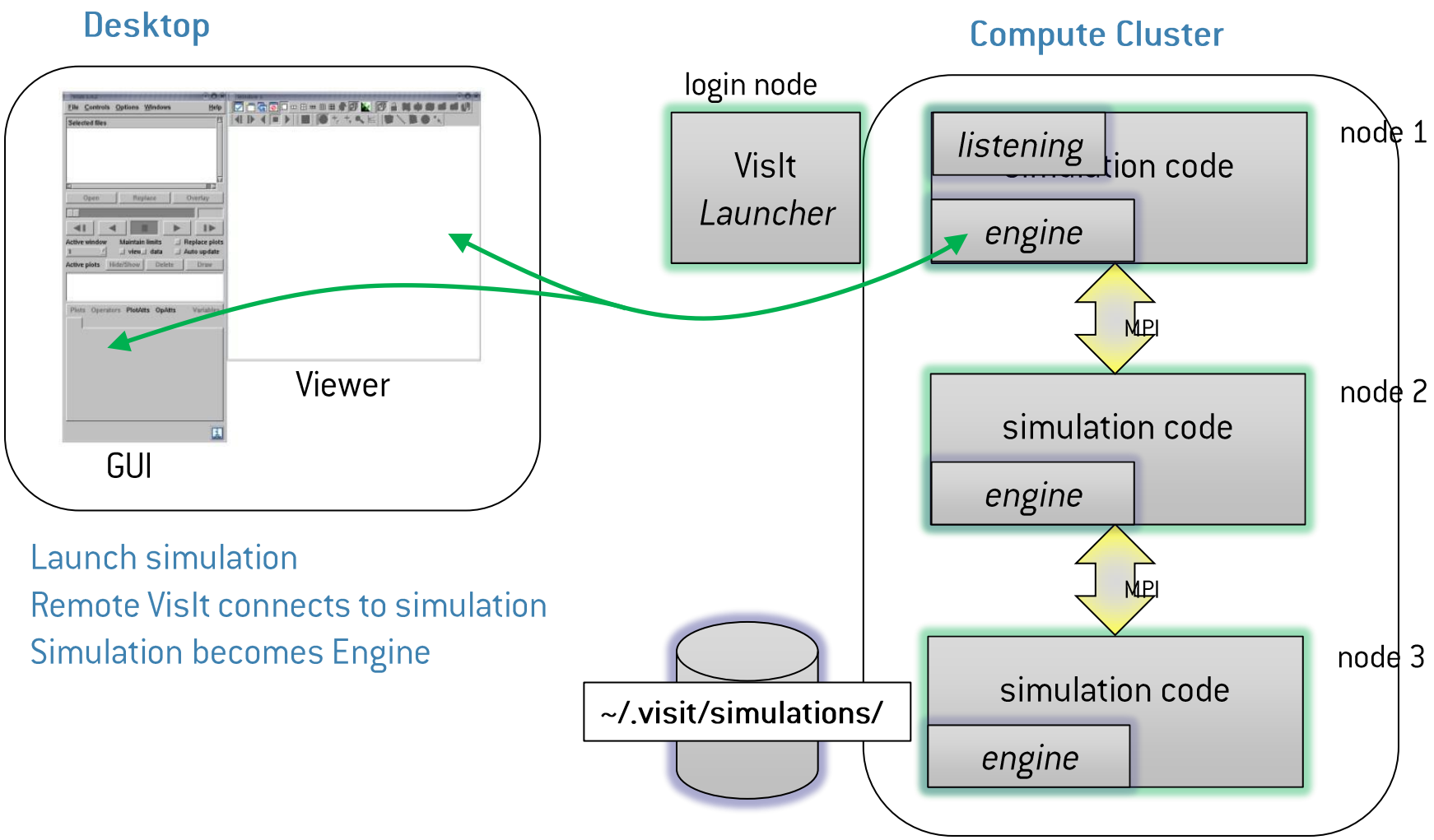
1. Launch simulation

In-Situ Visualization with VisIt



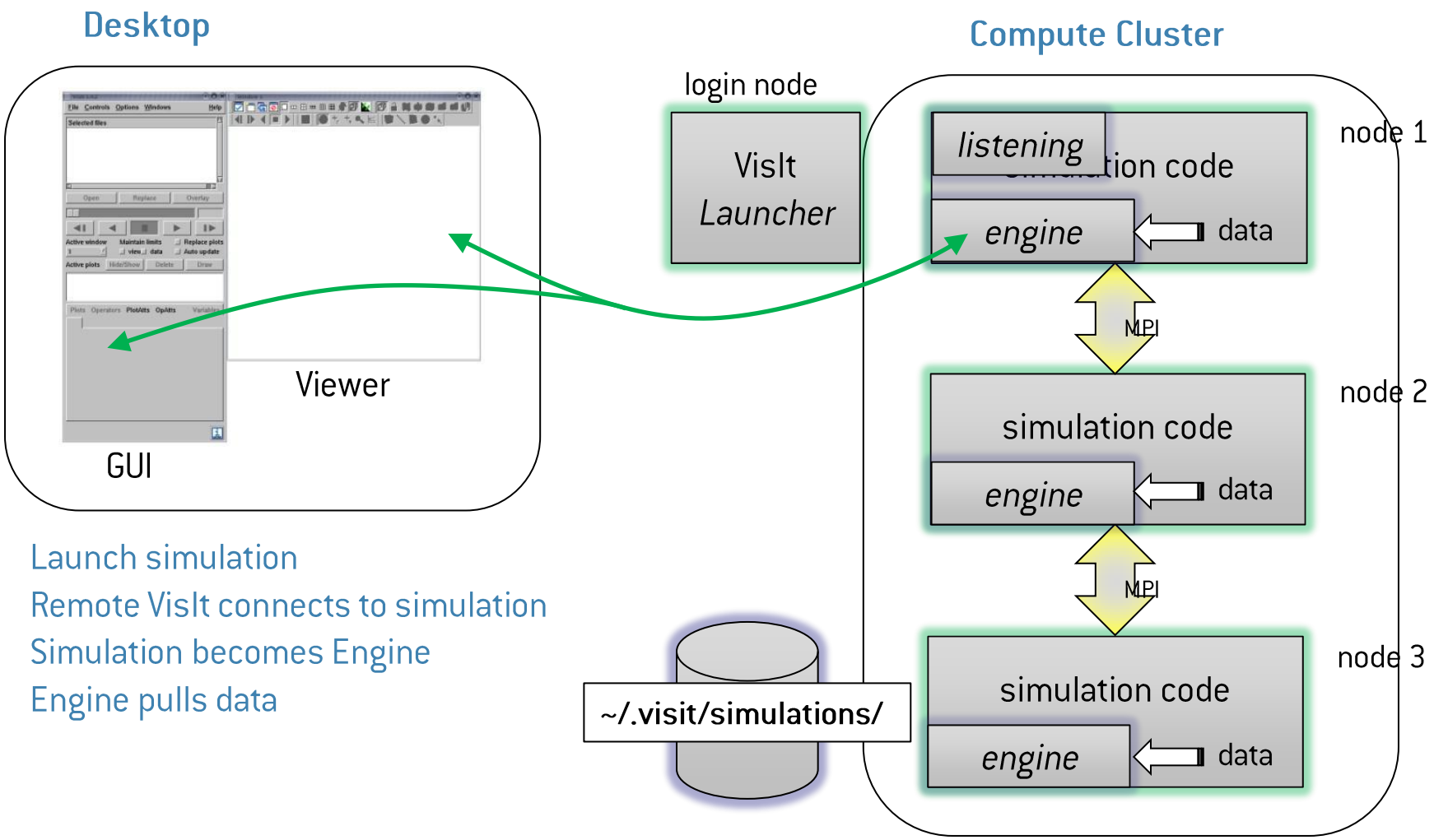
1. Launch simulation
2. Remote VisIt connects to simulation

In-Situ Visualization with VisIt



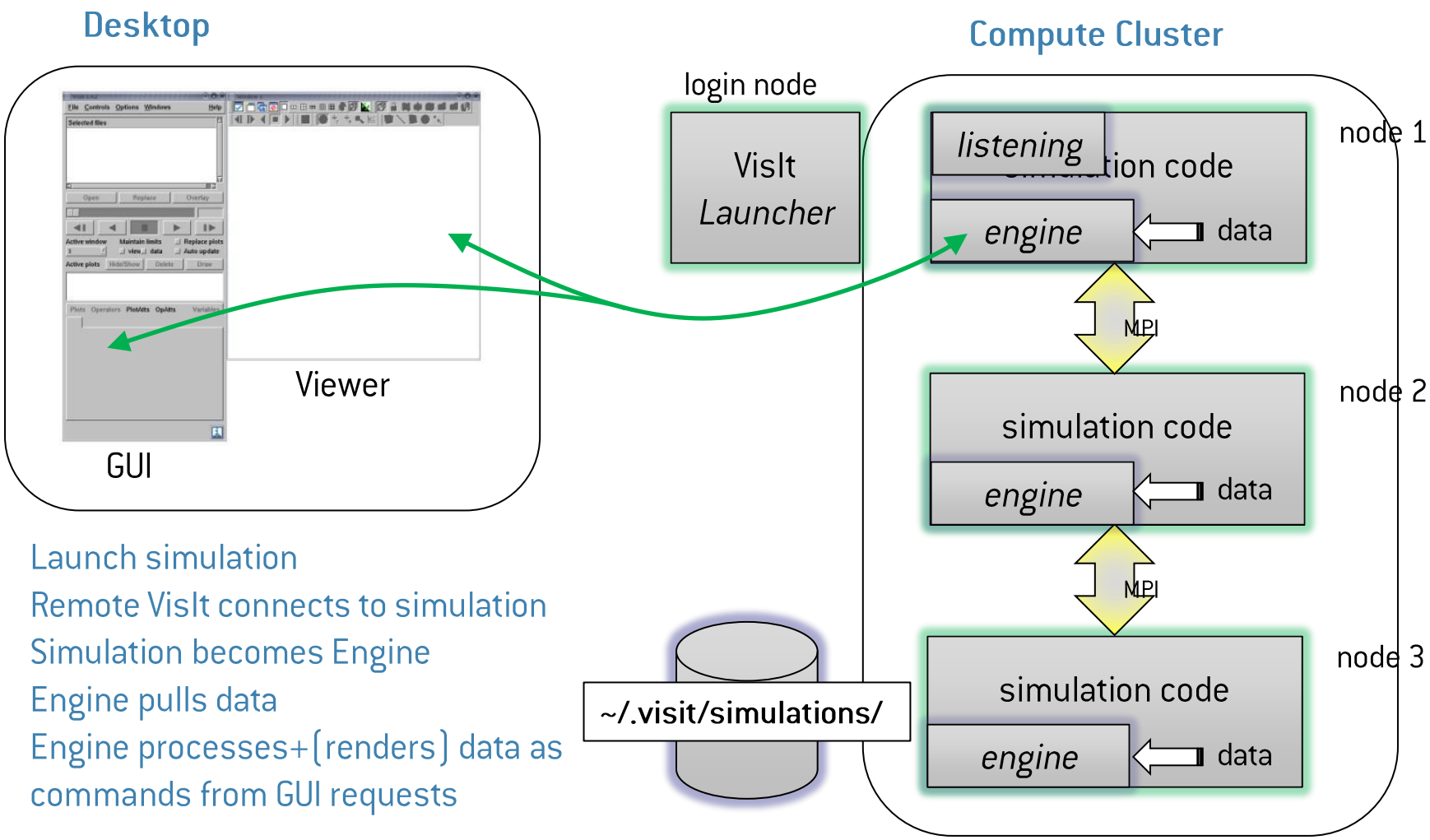
1. Launch simulation
2. Remote VisIt connects to simulation
3. Simulation becomes Engine

In-Situ Visualization with VisIt



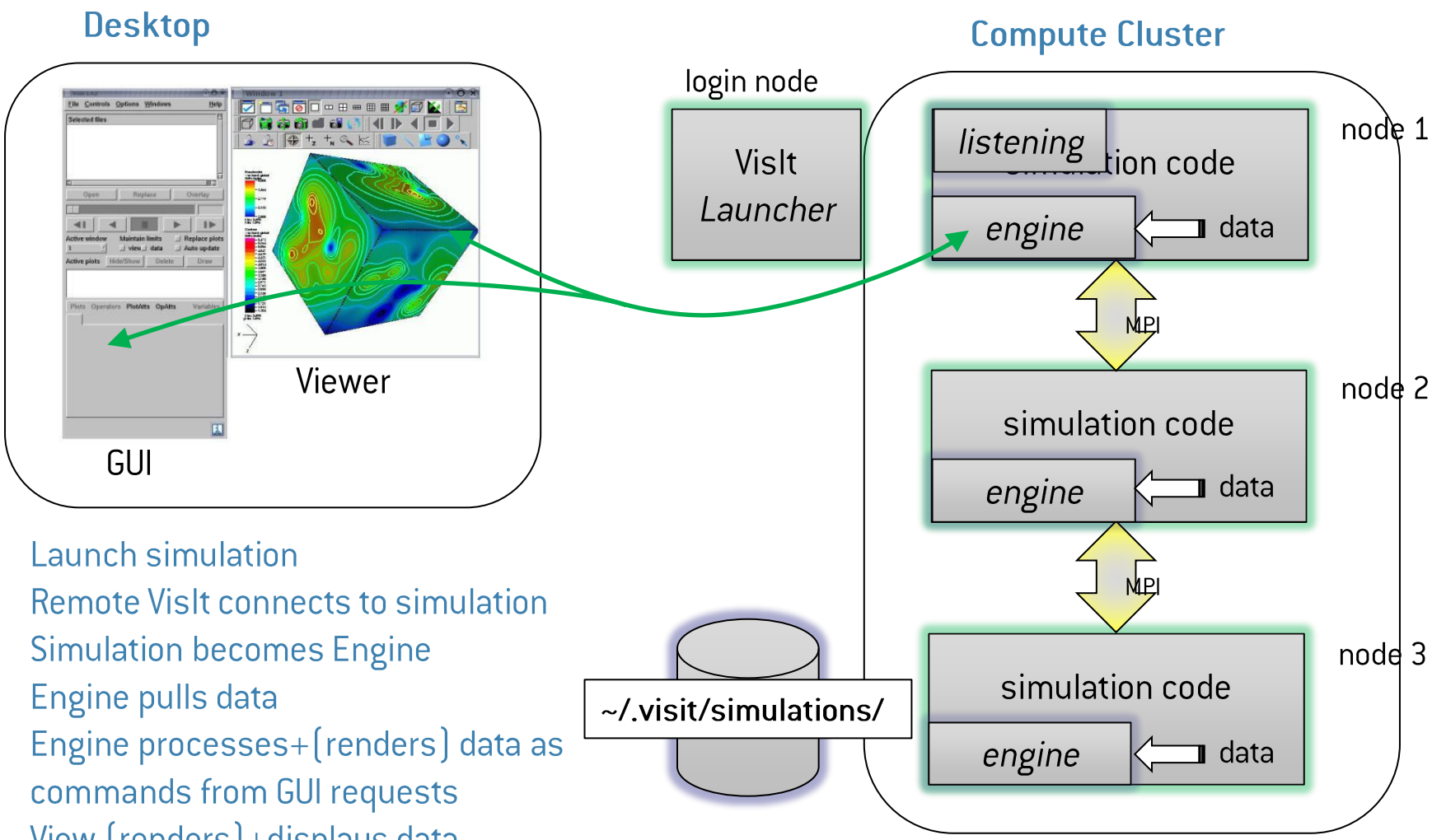
1. Launch simulation
2. Remote VisIt connects to simulation
3. Simulation becomes Engine
4. Engine pulls data

In-Situ Visualization with VisIt



1. Launch simulation
2. Remote VisIt connects to simulation
3. Simulation becomes Engine
4. Engine pulls data
5. Engine processes+(renders) data as commands from GUI requests

In-Situ Visualization with VisIt



1. Launch simulation
2. Remote VisIt connects to simulation
3. Simulation becomes Engine
4. Engine pulls data
5. Engine processes+(renders) data as commands from GUI requests
6. View (renders)+displays data

VisIt – Integration (source)

```
! main simulation loop
do

    ! check VisIt
    if( do_visit > 0 ) &
      visit_working = visit_checkstatus(ierr); CHKERRQ0(ierr)

    if(visit_working == 0) then

                    ! calc next iteration
                    call sim_timestep (ierr); CHKERRQ0(ierr)

                    [...]

                    ! if single step than switch to pause
                    if (simloop_mode == SINGLE_TSTEP) &
                      simloop_mode = PAUSE_TSTEP
    end if
end do
```

VisIt – Integration (source)

```
integer function visit_checkstatus (ierr)
  [...]

 ! detect input from proc 0 and broadcast that input to all others
 if (mpi_proc_id == 0) then
    visitState = VisItDetectInputWithTimeout(blocking_call, 0, -1)
 end if
 call MPI_BCAST(visitState, 1, MPI_INTEGER, 0, mpi_mycomm, ierr)

 [...]

! VisItDetectInput() returns with "VisIt wants to tell the engine something"
 else if (visitState == 2) then

    visit_cmd_serie = visit_cmd_serie +1
    visit_checkstatus = 1 ! recall visit_checkstatus(..)

    ! disconnect on an error or closed connection
    proccmdResult = visit_processCommand ()
    if (proccmdResult /= VISIT_OKAY) then
                        [...]
    end if
 end if
```

VisIt – Integration (source)

```
integer function visitgetmetadata ()
  [...]

  ! Add meshes to meta data
  if (visitmdmeshalloc(m1) == VISIT_OKAY) then
    ierr = visitmdMeshSetName(m1, "mesh3d", 6)
    ierr = visitmdMeshSetMeshType(m1,VISIT_MESHTYPE_RECTILINEAR)
    [...]

  ! Add mesh variables to meta data
  meshname = 'mesh3d'
  vecname = 'velocity'
  if (visitmdvaralloc(vmd) == VISIT_OKAY) then
    ierr = visitmdVarSetName(vmd, trim(vecname), len_trim(vecname))
    ierr = visitmdVarSetMeshName(vmd, trim(meshname), len_trim(meshname))
    [...]

  ! Add simulation commands to meta data
  if (visitmdcmdalloc(cmd) == VISIT_OKAY) then
    ierr = visitmdcmdsetname(cmd, "pause", 5)
    ierr = visitmdsimaddgenericcommand(md, cmd)
    [...]
```

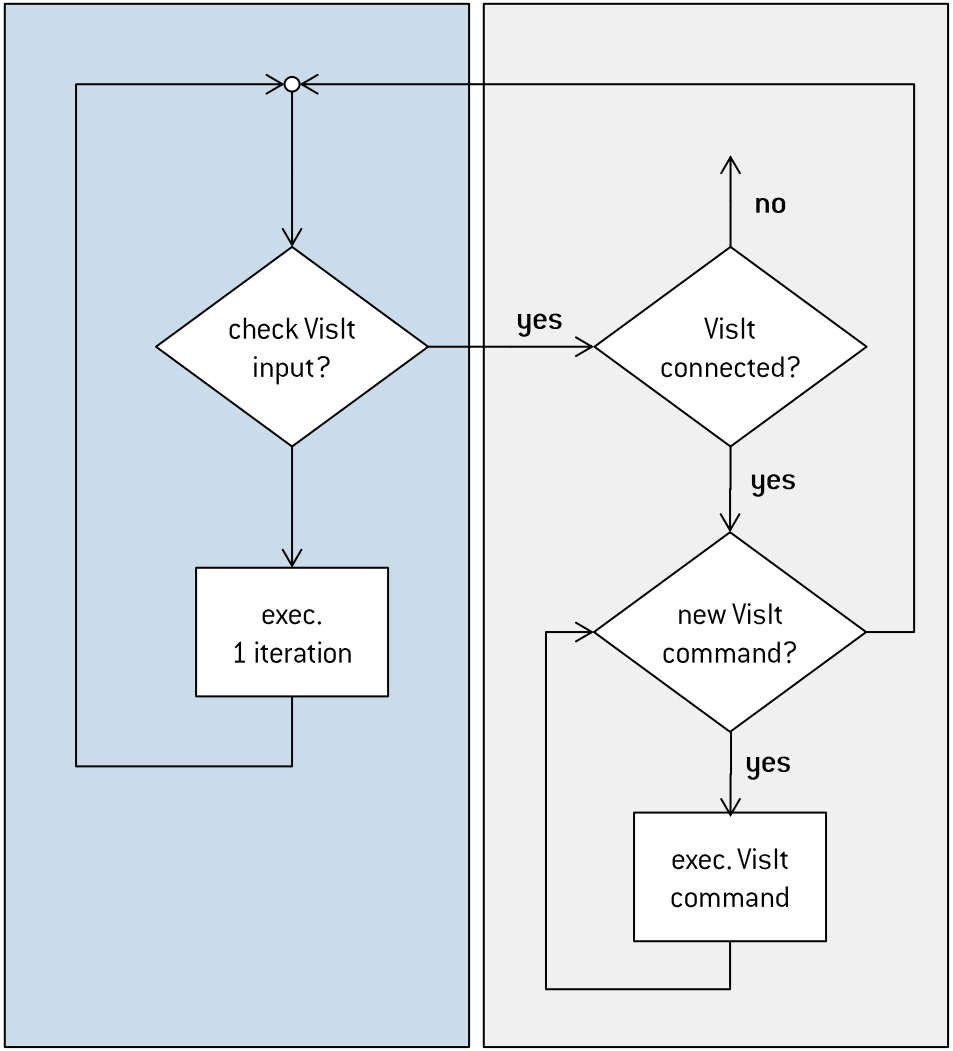
VisIt – Integration

- simulation main loop is augmented
- callbacks are added to advertise the data

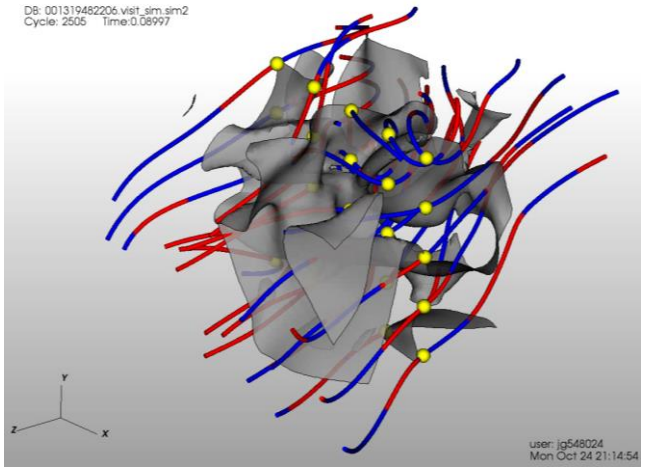
=> complete integration in psOpen
only 1447 lines of code

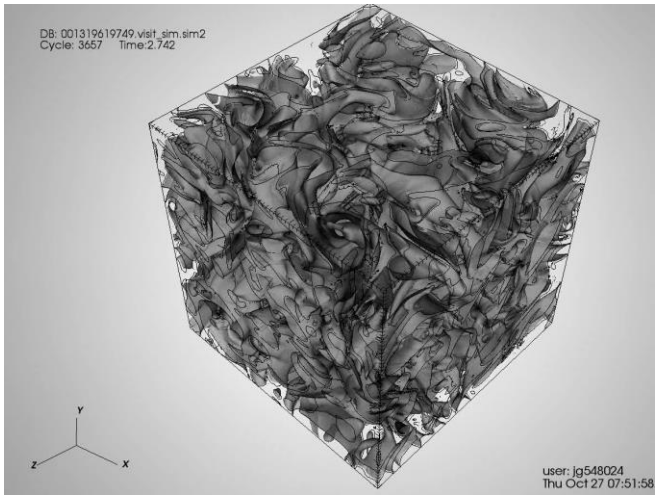
VisIt Simulation API
- set up the environment
- open a socket and start listening
- process a VisIt command
- set the control callback routines

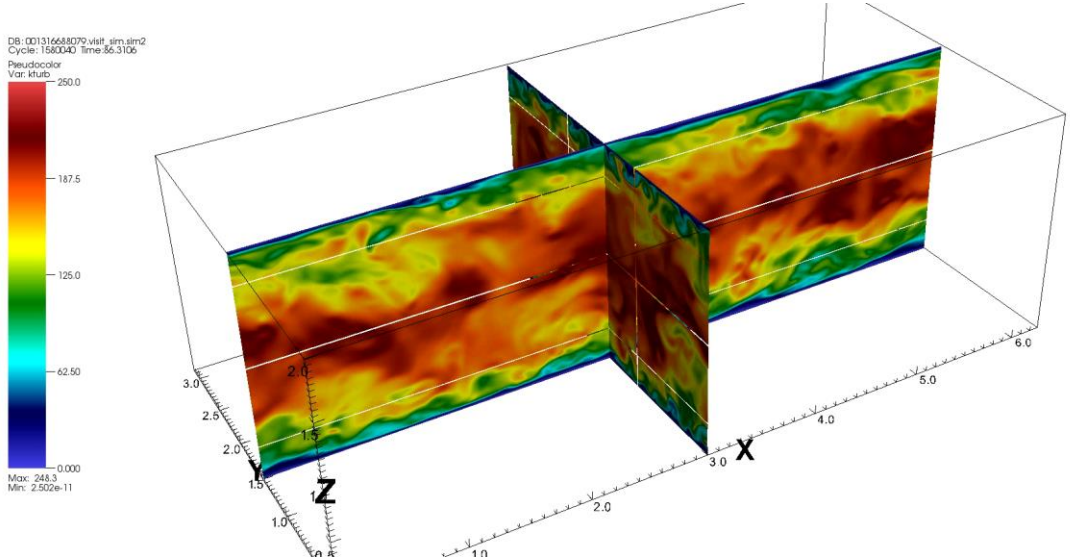
VisIt Data API
- GetMetaData()
- GetMesh()
- GetScalar(), etc



Results

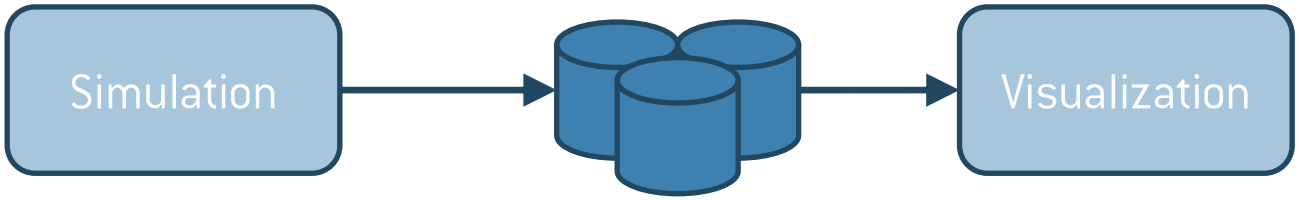

DB: 001319482206.visit_sim.sim2
Cycle: 2505 Time:0.08997
user: jg548024
Mon Oct 24 21:14:54


DB: 001319619749.visit_sim.sim2
Cycle: 3657 Time:2.742
user: jg548024
Thu Oct 27 07:51:58


DB: 001316688079.visit_sim.sim2
Cycle: 1580040 Time:86.3106
Pseudocolor
Var: ktotb
250.0
187.5
125.0
62.50
0.000
Max: 248.3
Min: 2.502e-11
X
Z

Coupling Strategies

„no coupling“



„tight coupling“

„loose coupling“

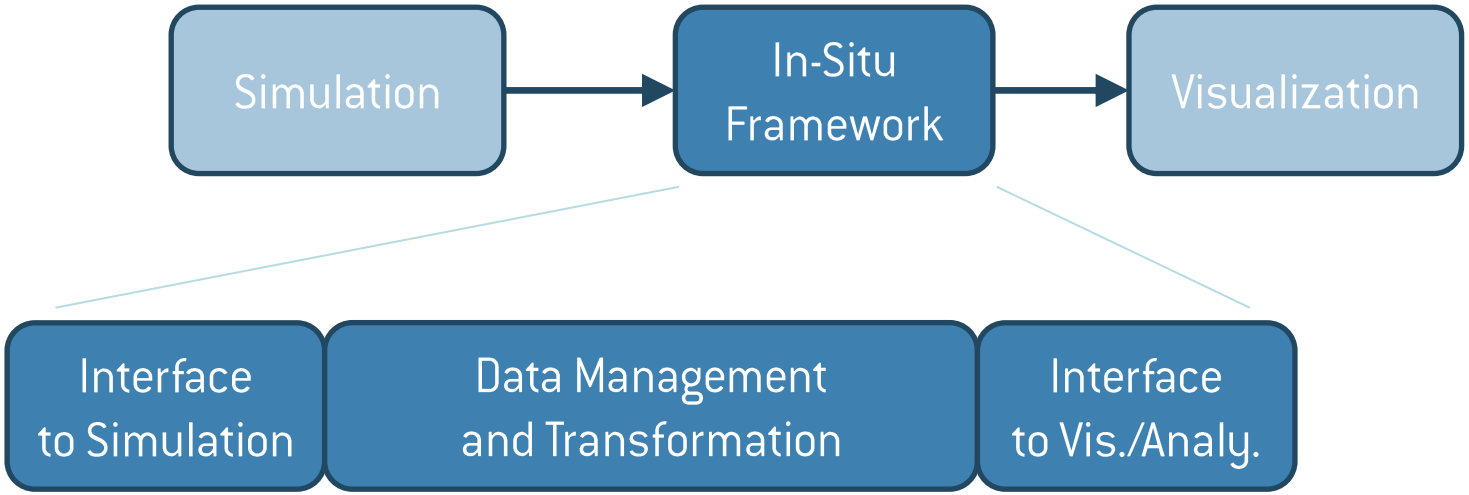
Outlook – In-Situ Framework

Researchers seldomly accept tight coupled in-situ visualization

- Usability
  - low impact on simulation code
    - no compile/link difficulty
    - no impact of runtime errors
  - adaptability (to different simulations and visualization scenarios)

- Performance
  - low impact on simulation run time
  - good resource utilization

Outlook – In-Situ Framework



- ADIOS
- HDF5

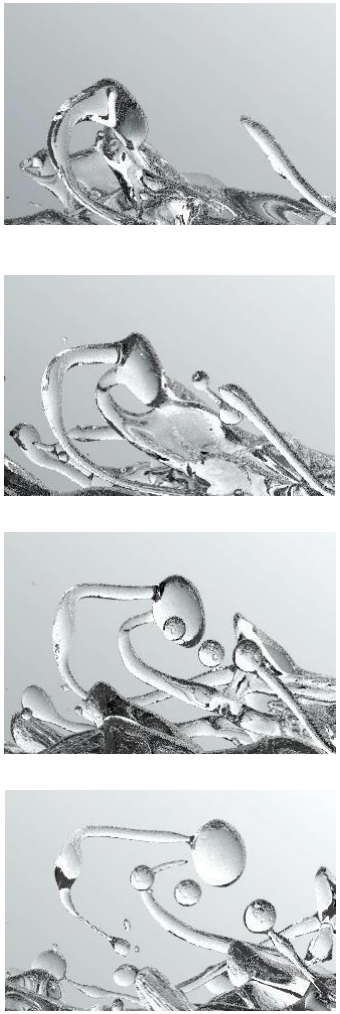
- data sharing using a staging area
- direct point-to-point transfers bypass any staging area

Current Project

Detailed Investigation of Liquid Sheet Breakup Using Direct Numerical Simulation and In-situ Visualization

- understanding of the mechanism underlying the breakup of ligaments into multiple droplet
- resolving the breakup of ligaments temporally and spatially
- in-situ visualization → saving all required data to disk is impossible

DNS of a scaled-up Diesel injector [1]



Breakup of a ligament into multiple droplets

[1] Le Chenadec, V.H.M.; Pitsch, H.: A Monotonicity Preserving Sharp Interface Flow Solver for High Density Ratio Two-Phase Flows. J. Comp. Phys., in press, 2013.

Conclusion

Integration of In-Situ Visualization to HPC codes

… is easy.

details will be gathered by the Cross-Sectional Team „Visualization“ on

https://trac.version.fz-juelich.de/vis