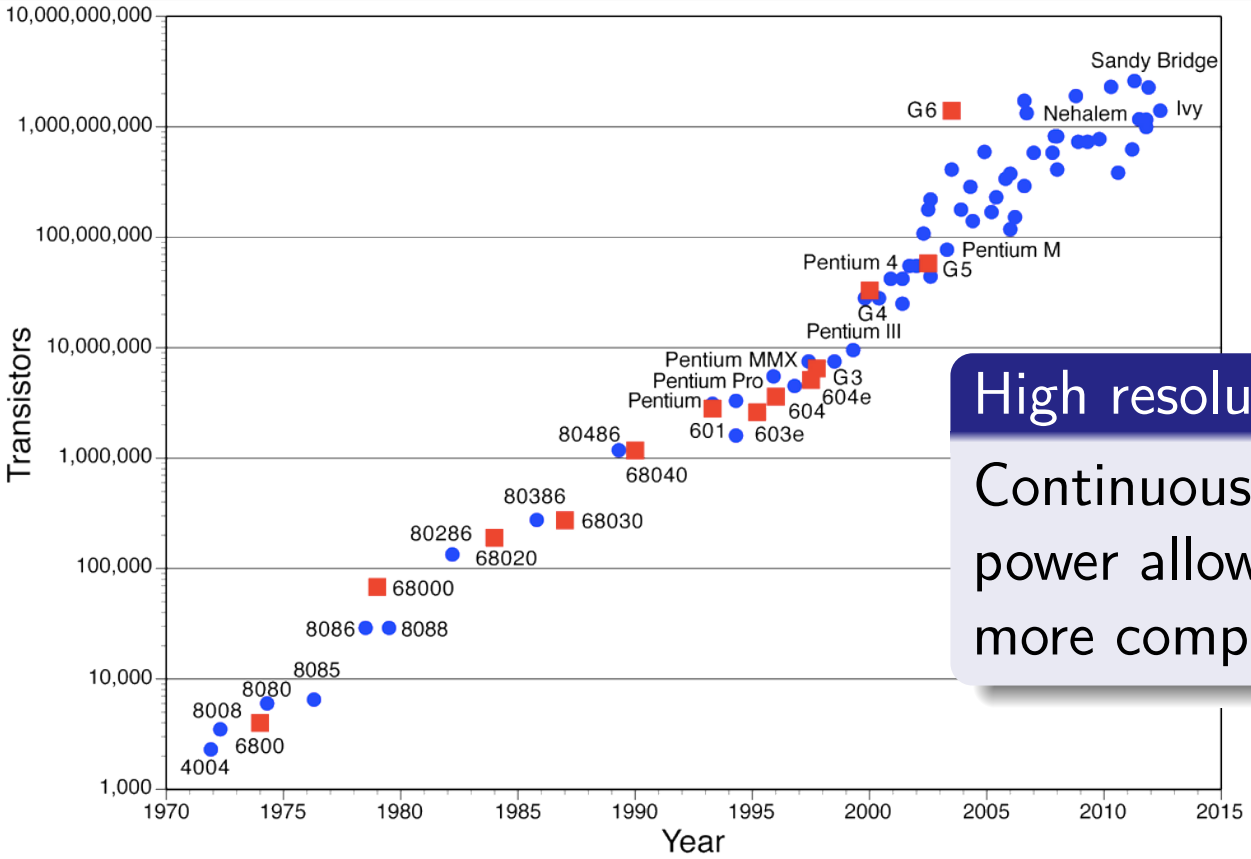
Visualisations tridimensionnelles de données issues de simulations numériques à très haute résolution en mécanique des fluides

Patrick Bégou (LEGI/MoST)
Patrick.Begou@legi.grenoble-inp.fr

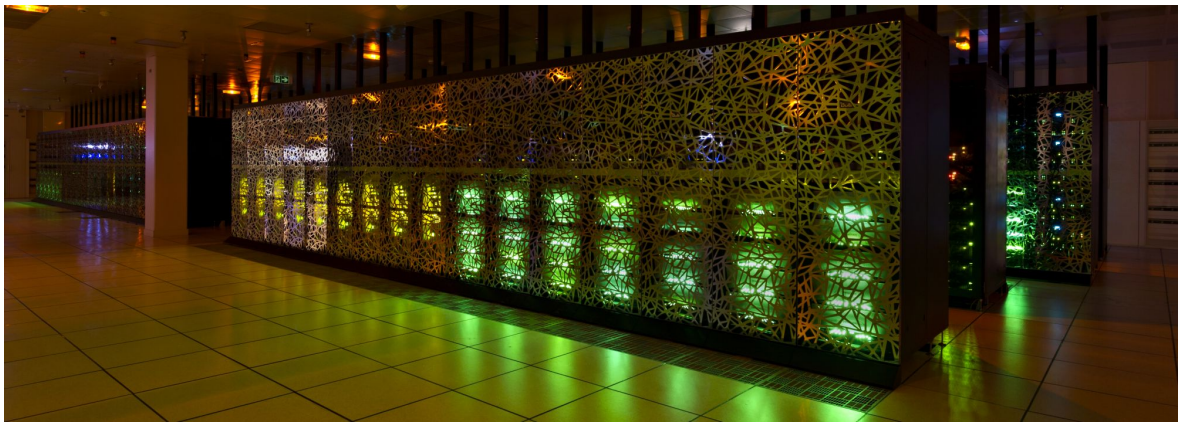
30 novembre 2017

Ever more powerful computational tools



High resolution simulations

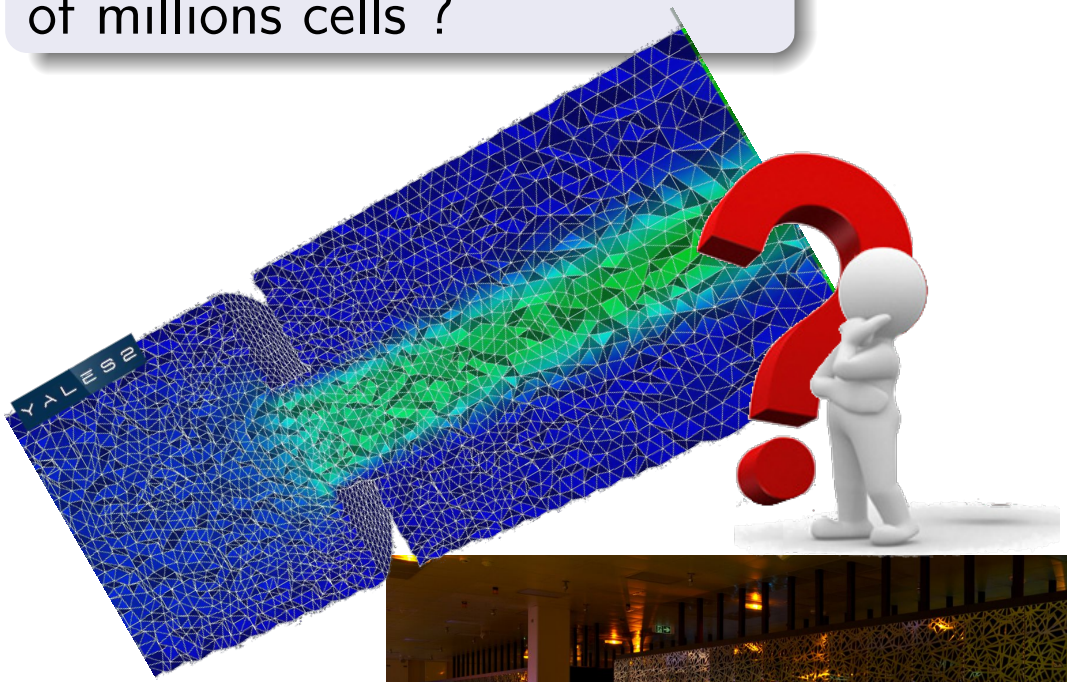
Continuous growth of computational power allows to investigate more and more complex problems.

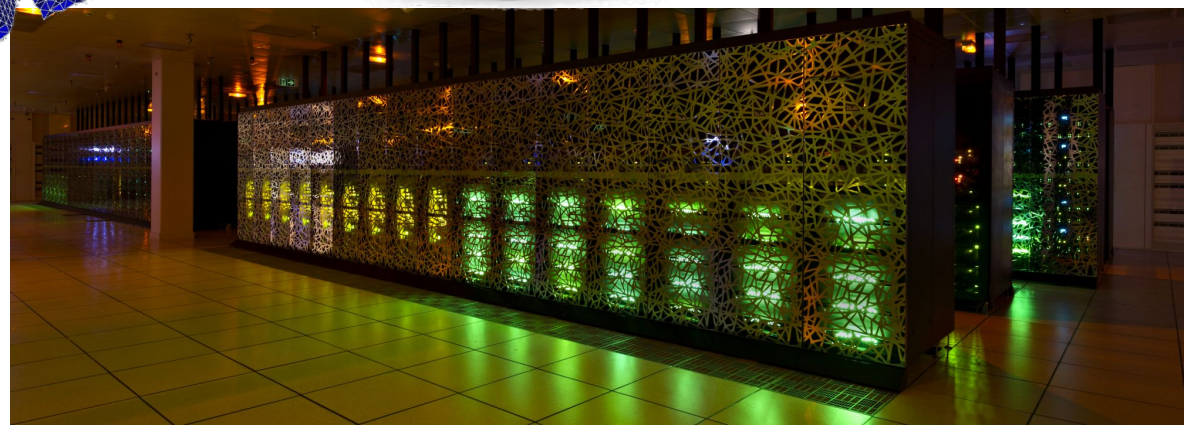


Ever more powerful computational tools

Upstream

How to create [complex] meshes with several hundreds of millions cells ?





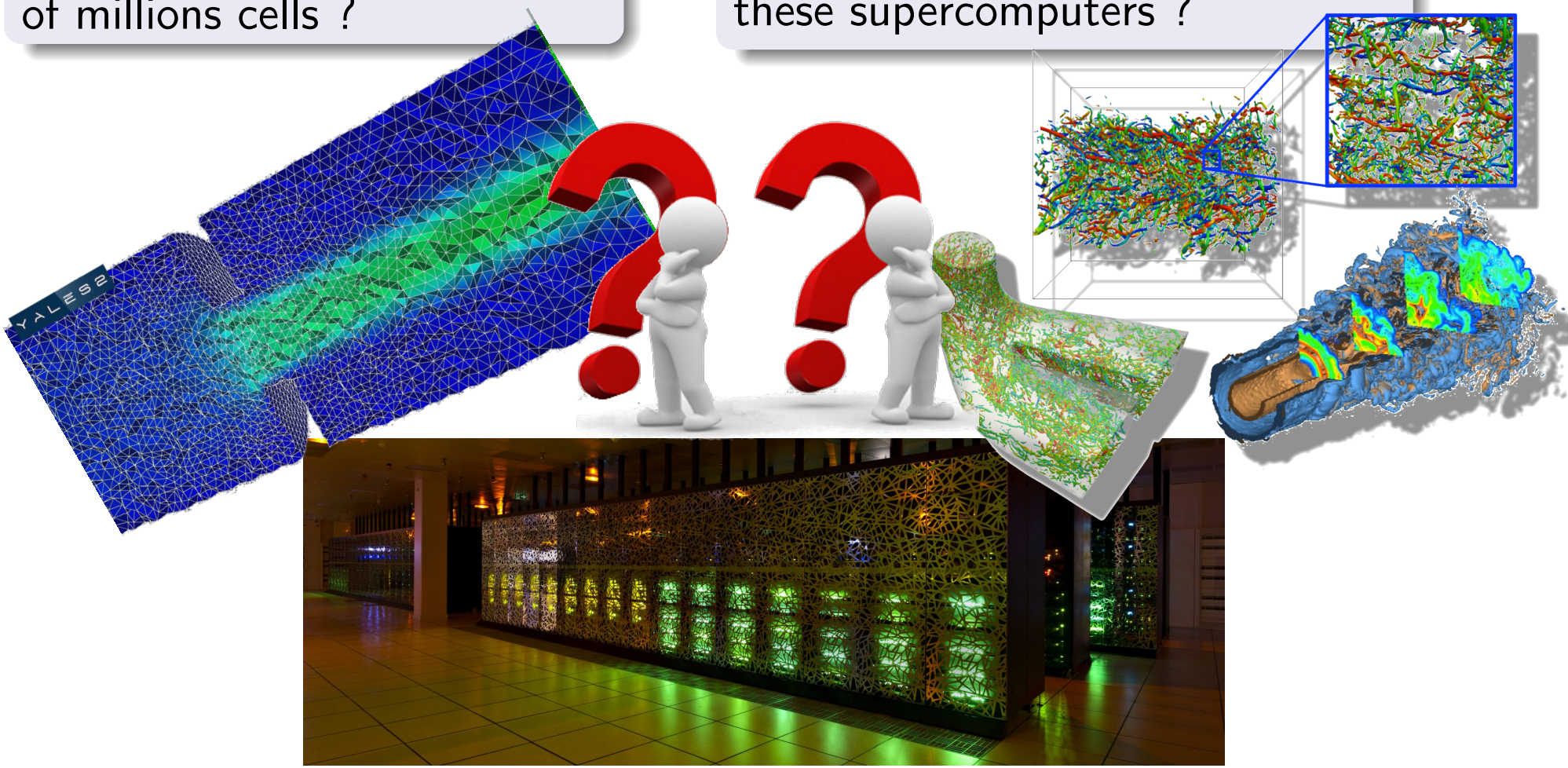
Ever more powerful computational tools

Upstream

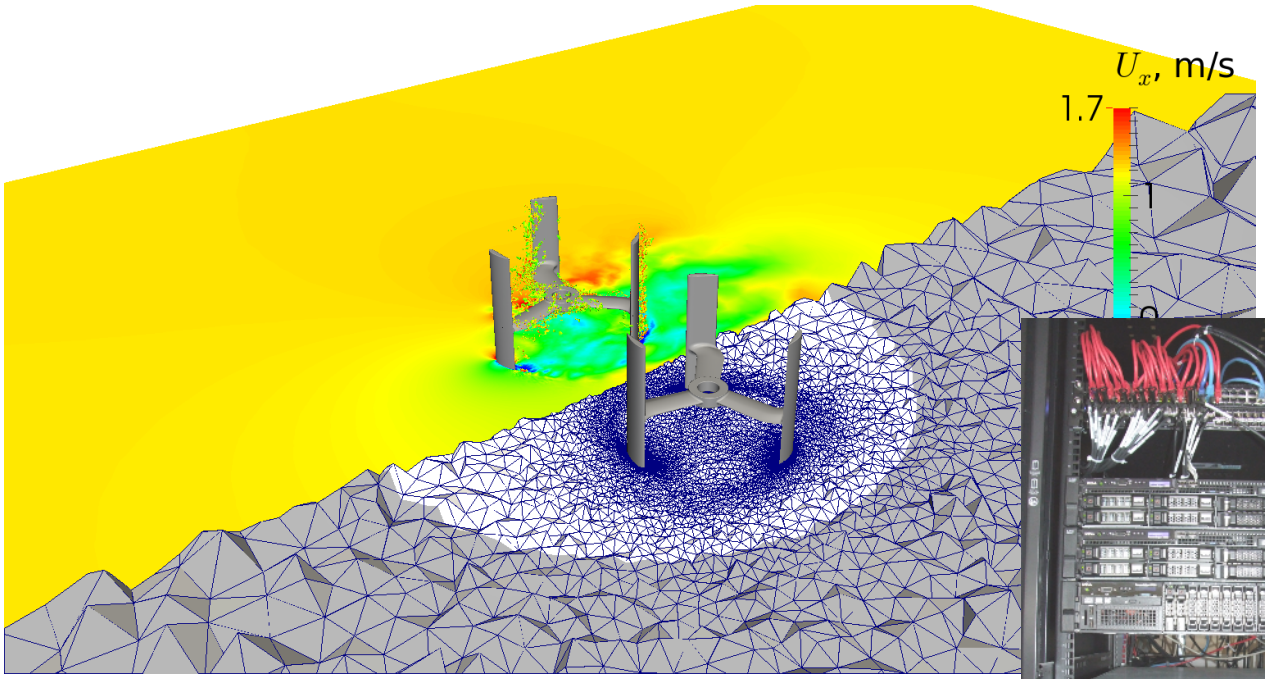
How to create [complex] meshes with several hundreds of millions cells ?

Downstream

How to post-process and visualize these large datasets generated on these supercomputers ?



Ever more powerful computational tools

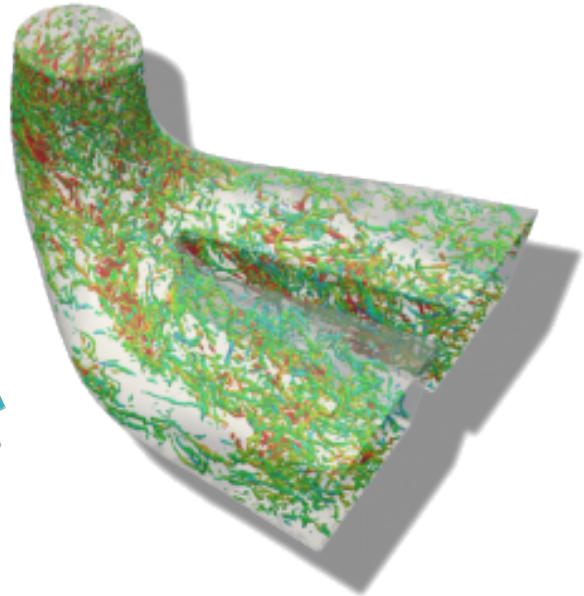


Picture: Andrey Pushkarev
LEGI/MoST

Local computational resources

- Parallel simulations
- 19 millions cells
- 200 cores on the local cluster

Ever more powerful computational tools

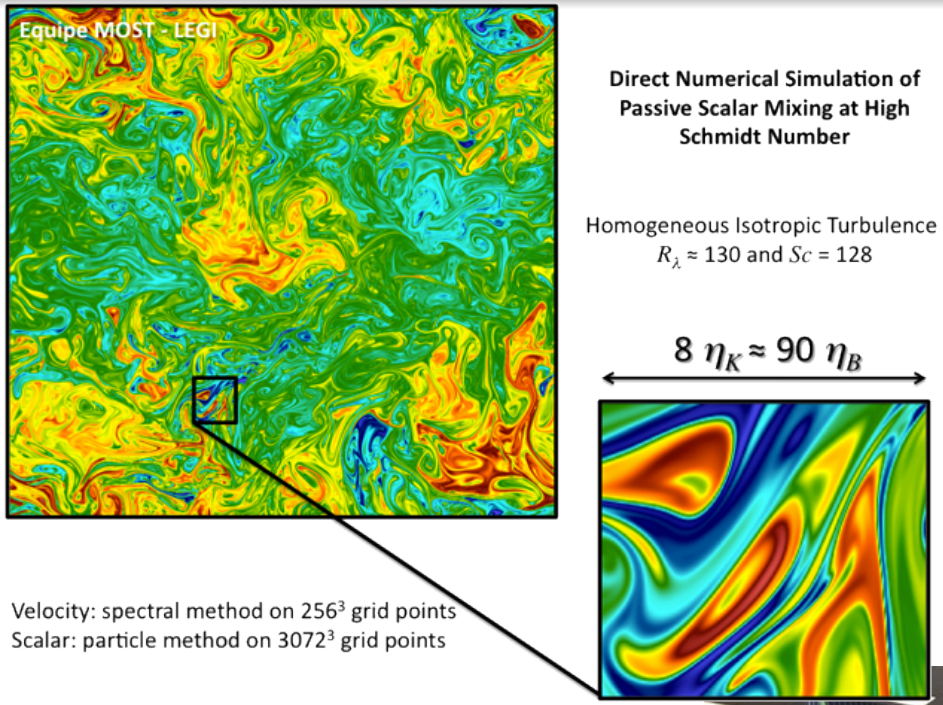


Picture: Guillaume Balarac
LEGI/MoST

Mesocenters computational resources

- Highly parallel simulations
- 148 millions cells
- Until 2048 cores

Ever more powerful computational tools

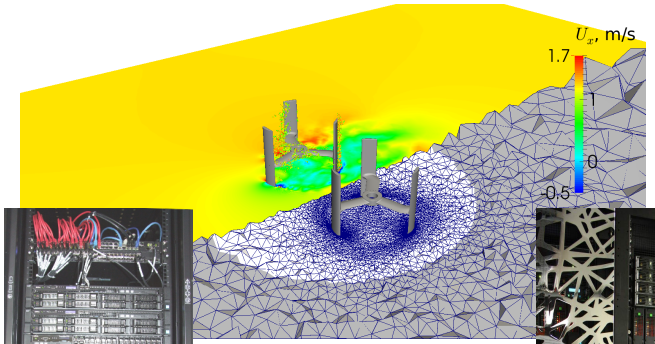


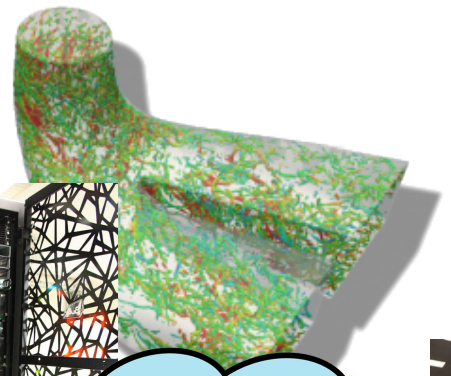
Picture: Jean Baptiste Lagaert LEGI/MoST

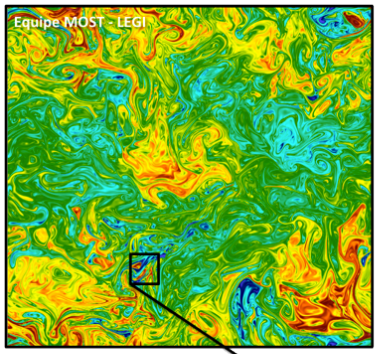
National computional resources

- Massively parallel simulations
- 29 billions cells
- until 16384 cores on the IBM Blue-Gene *Turing*

Size of the data for visualization



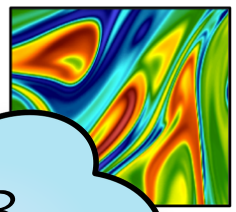




Direct Numerical Simulation of Passive Scalar Mixing at High Schmidt Number

Homogeneous Isotropic Turbulence
$R_\lambda \approx 130$ and $Sc = 128$

$8\,\eta_K \approx 90\,\eta_B$



684*MB* velocity + coord.

5.3*GB* Velocity + coord.

232*GB* Scalar

Local computational resources

- 19M cells
- ⇒200 cores

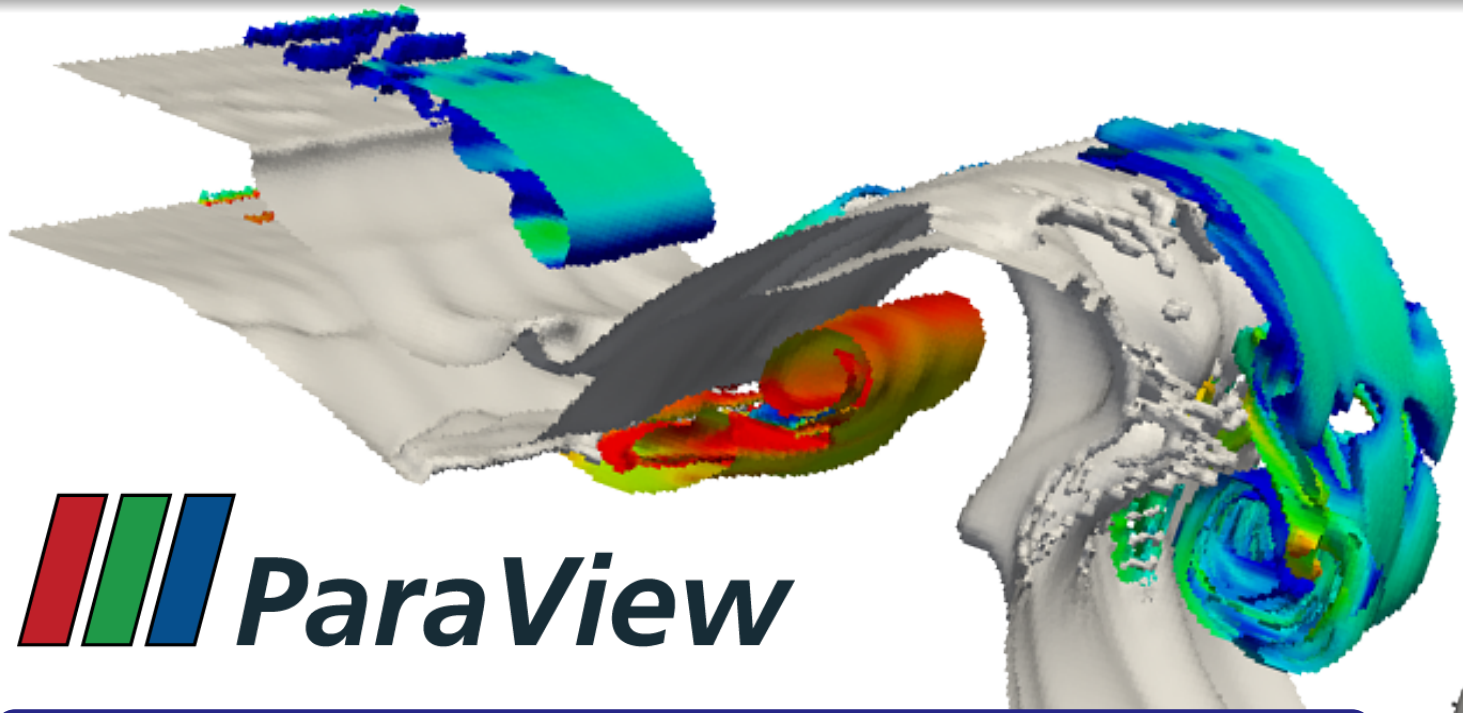
Mesocenters computational resources

- 148M cells
- ⇒2048 cores

National computational resources

- 29 billions cells
- ⇒16384 cores

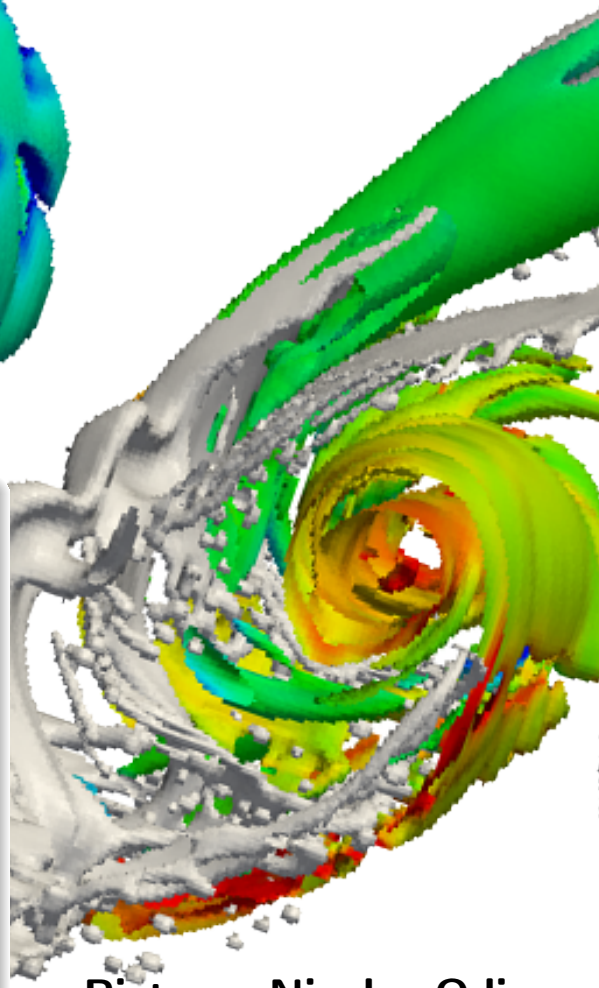
Paraview



ParaView

Paraview

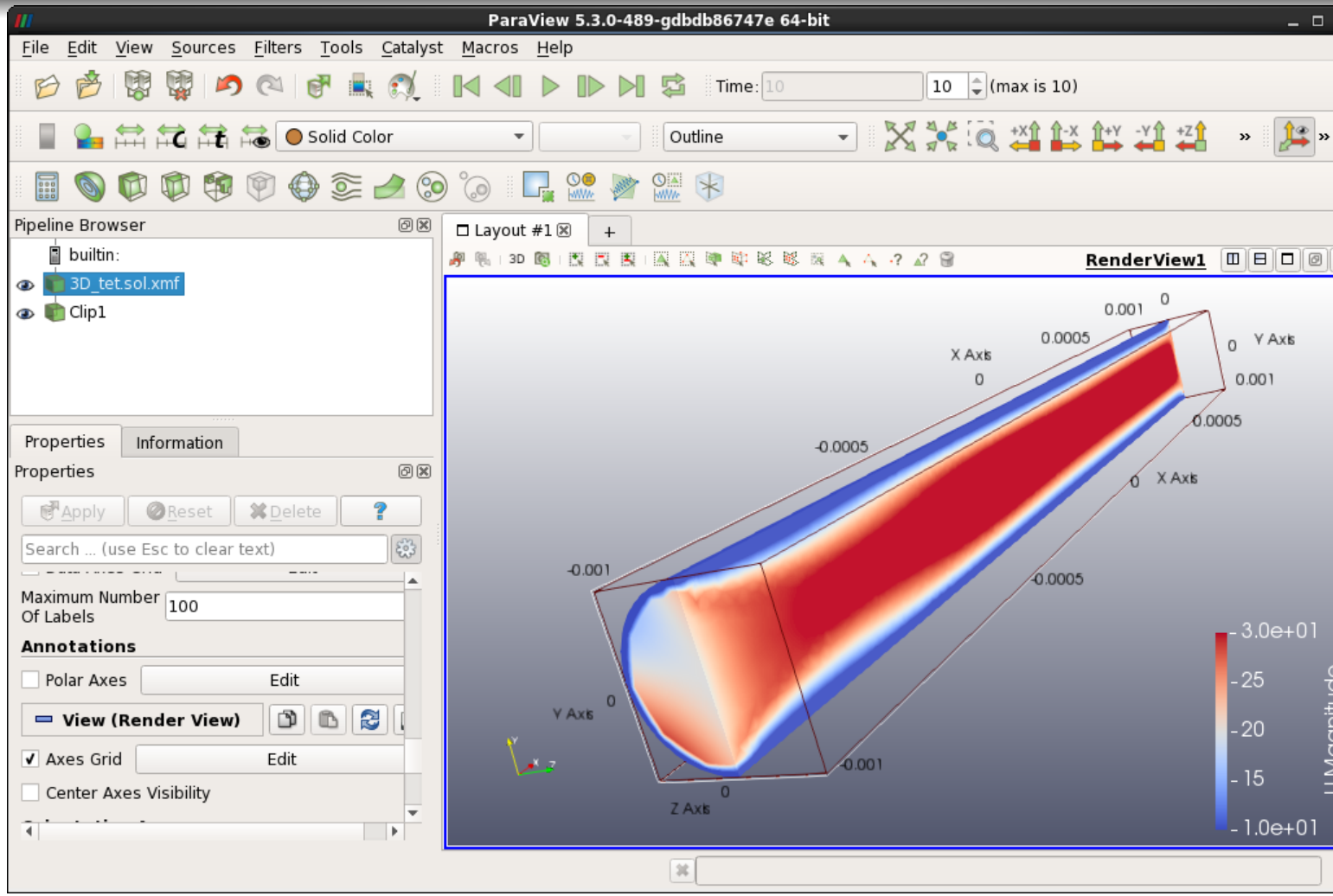
- Open source software for scientific visualizations
- From laptops to supercomputers
- Used for many years at LEGI
- First release 0.6 in 2002, version 5.4 in june 2017!
- Large user community



Picture: Nicolas Odier LEGI/MoST

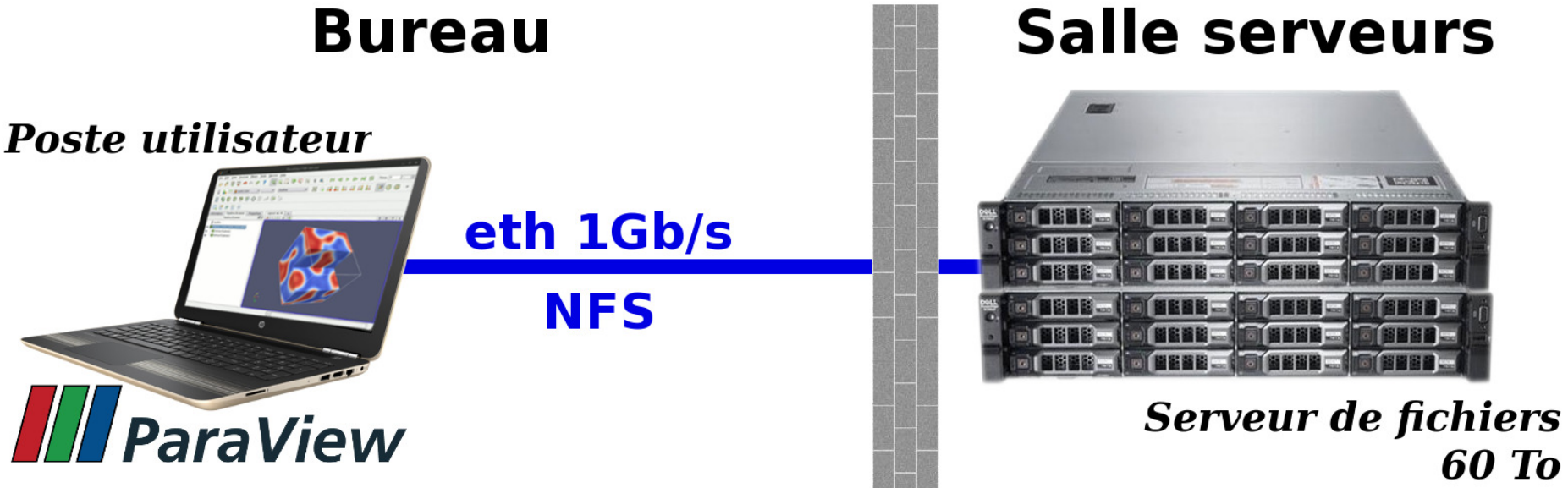
Paraview

Paraview Graphic User Interface



- **pvpython** for paraview scripting (python)
- **pvbatch** for paraview scripting (non interactive mode)
- **pvserver** client/server mode

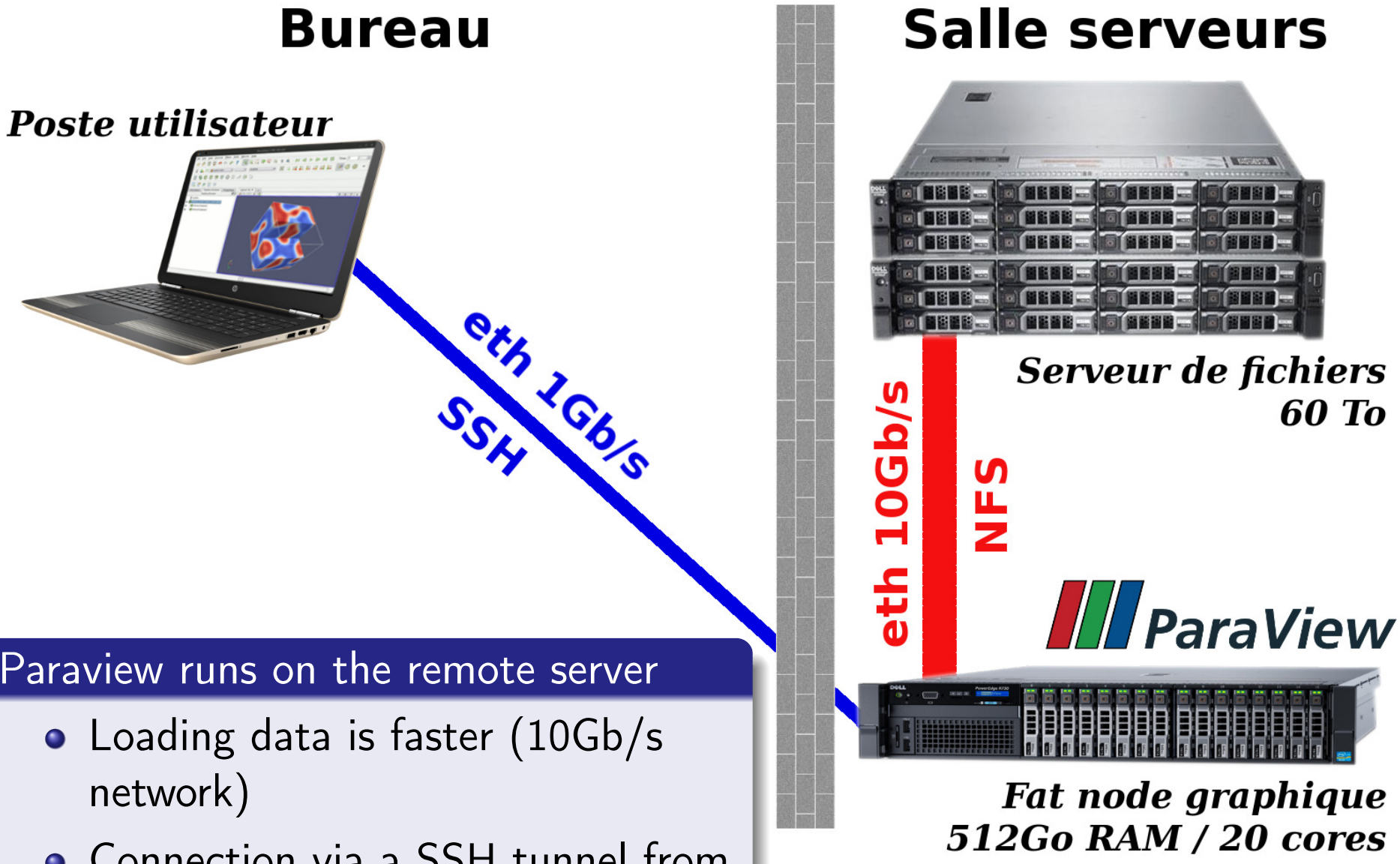
Load data from a remote server



Paraview runs on the laptop

- Access to remote data via NFS
- Loading data from file is slow (1Gb/s network)
- Laptop computing and memory capacities are low

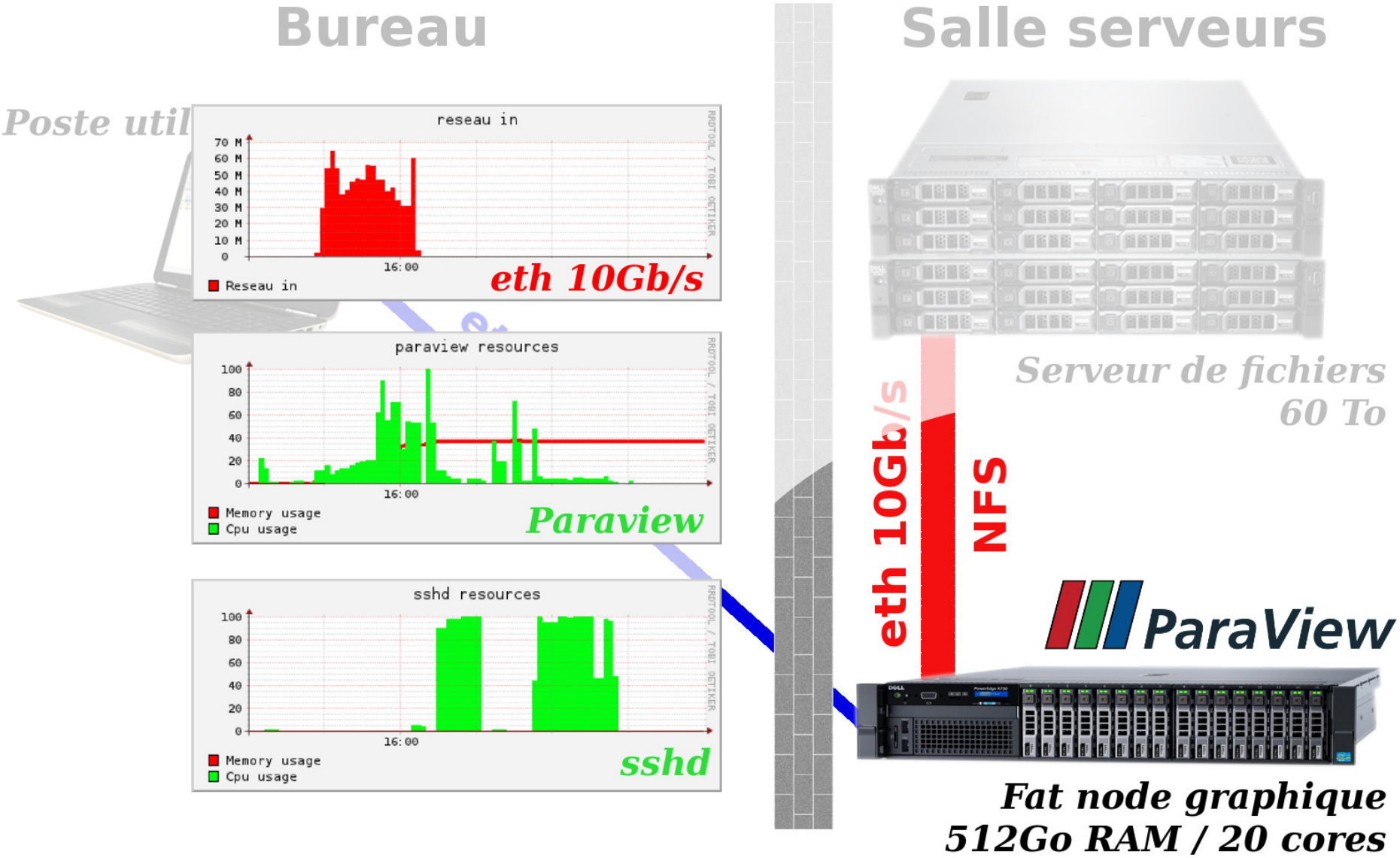
Remote fat node with GPU



Paraview runs on the remote server

- Loading data is faster (10Gb/s network)
- Connection via a SSH tunnel from the laptop

Remote fat node with GPU



Graphic session through the ssh tunnel uses a lot of cpu resources and is causing slowdowns (verbose X11 protocol)

Virtual Network Computing

Bureau

Poste utilisateur

TigerVNC Client

eth 1Gb/s
VNC dans SSH

Salle serveurs

Serveur de fichiers 60 To

eth 10Gb/s

NFS

TigerVNC Serveur

ParaView

Fat node graphique 512Go RAM / 20 cores

Remote graphic session using VNC

- Simple connection via a SSH tunnel (-*via* option)
- Data compression
- Optimized network bandwidth

But. . .

All is sequential

- Isosurfaces computations become slow as data size increase!
- How to add parallelism and speed up visualization ?
- How to take advantage of the 20 cores of this "*fat node*" ?

Let's talk parallel

build-charlie-manual : mpirun

File Edit View Scrollback Bookmarks Settings Help

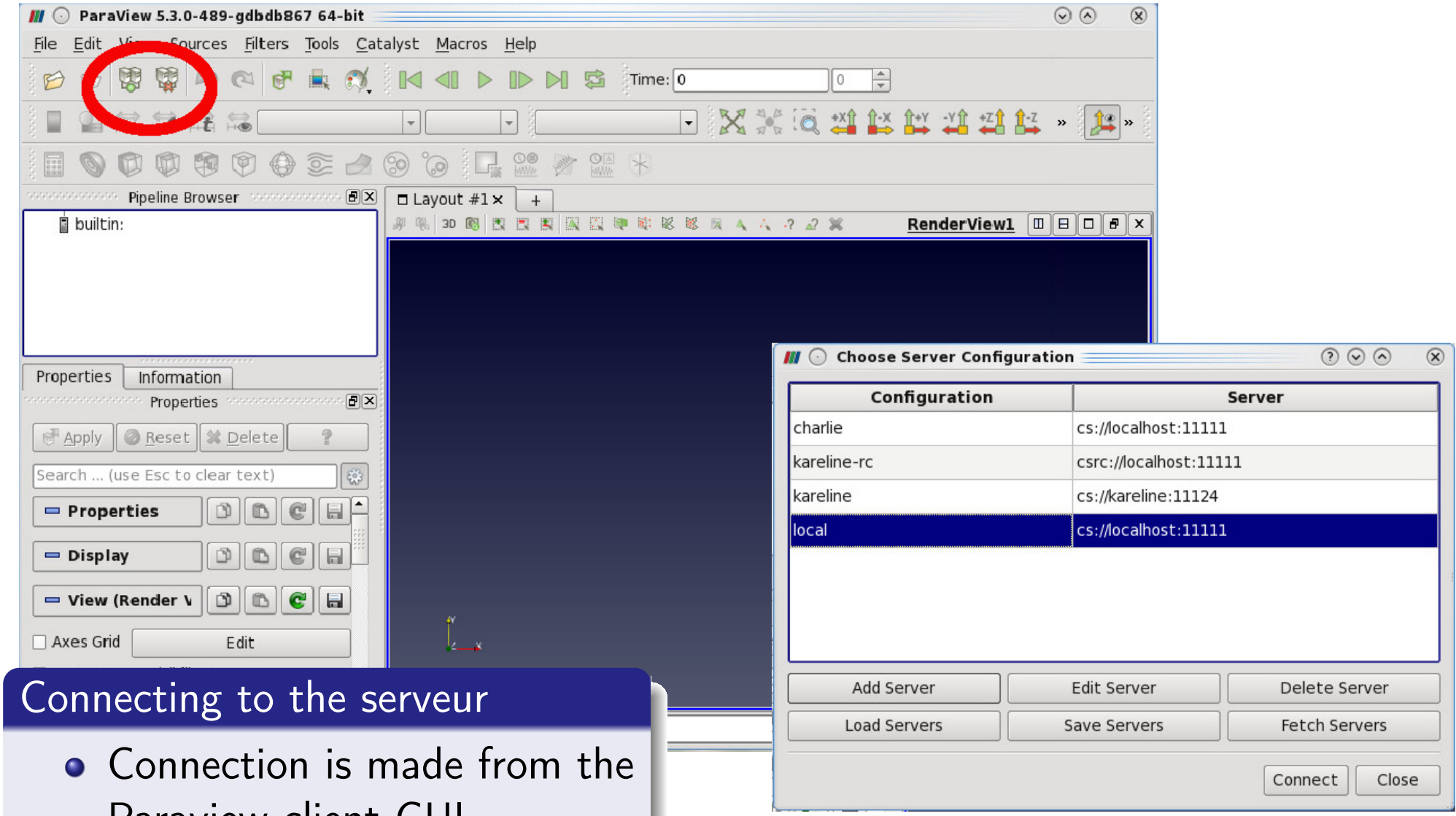
```
bash-4.1$ mpirun -np 10 pvserver
Waiting for client...
Connection URL: cs://charlie.legi.grenoble-inp.fr:11111
Accepting connection(s): charlie.legi.grenoble-inp.fr:11111
```

Launching several pvserver instances

- On the "*fat node*" we launch **10** pvserver occurences
- **mpirun -np 10 pvserver**
- They are waiting for the client connection on the 11111 default port (can be configured)

Fat node graphique
512Go RAM / 20 cores

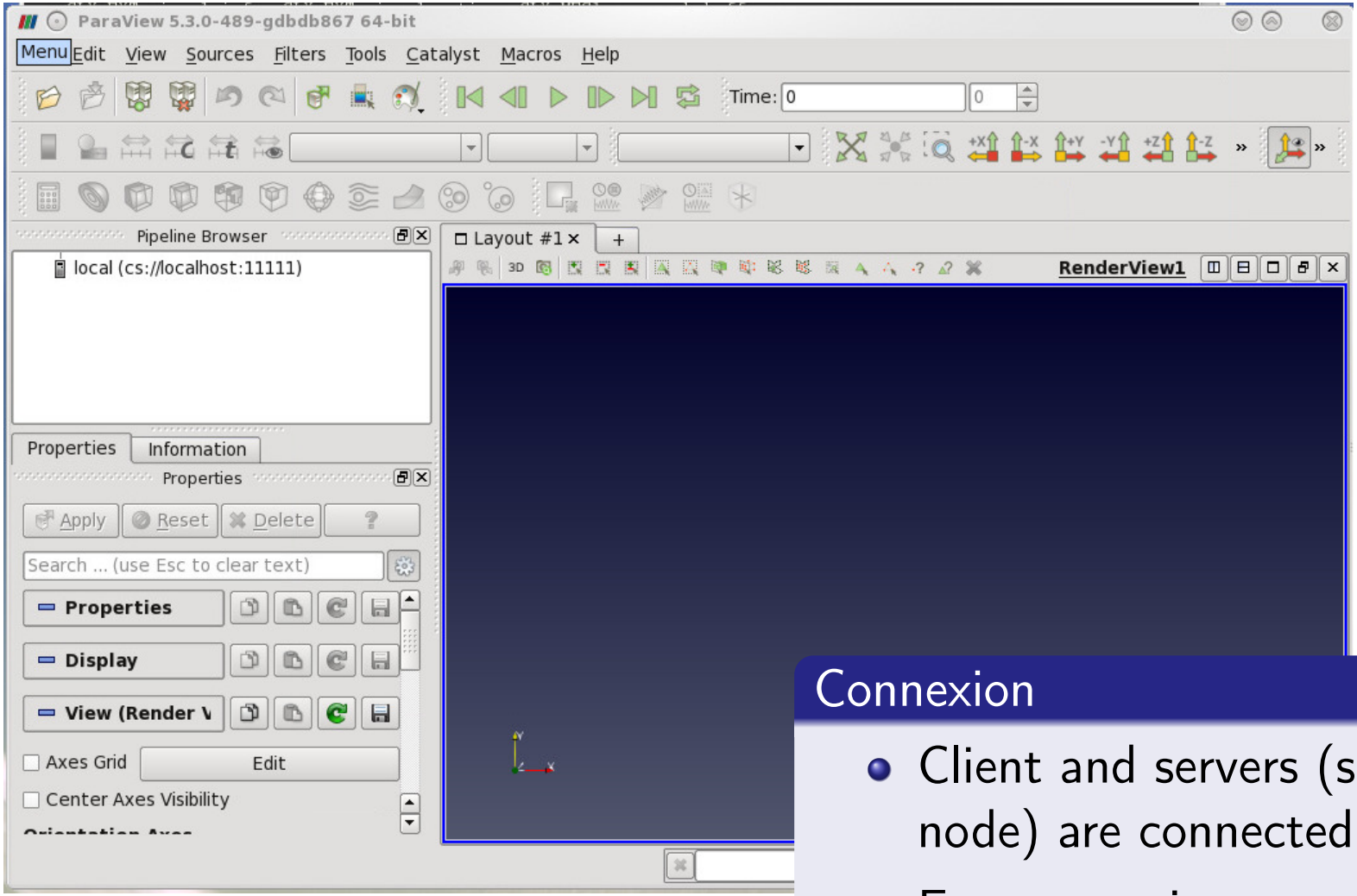
Let's talk parallel



Connecting to the serveur

- Connection is made from the Paraview client GUI
- Server is **localhost:11111**

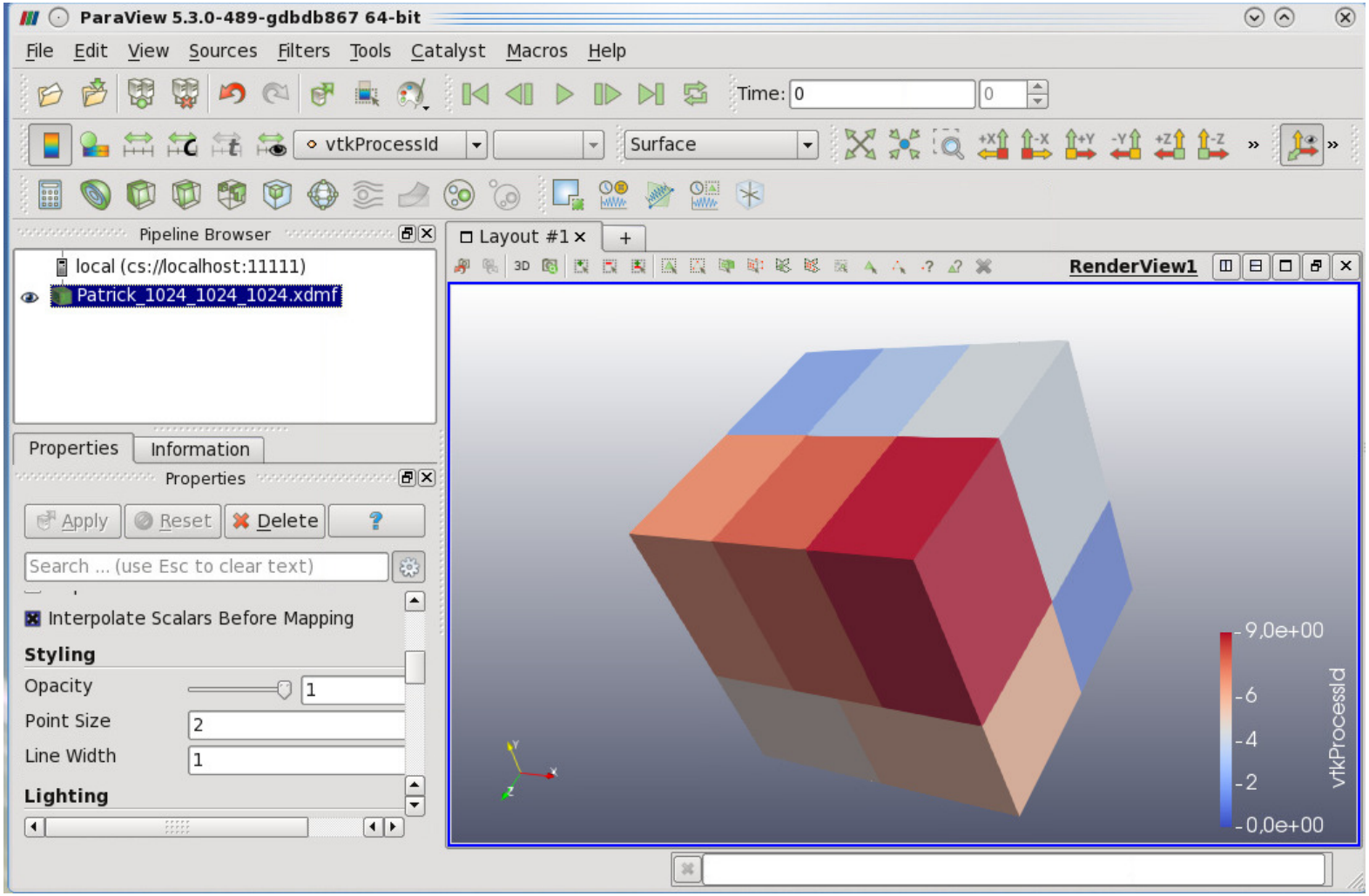
Let's talk parallel



Connexion

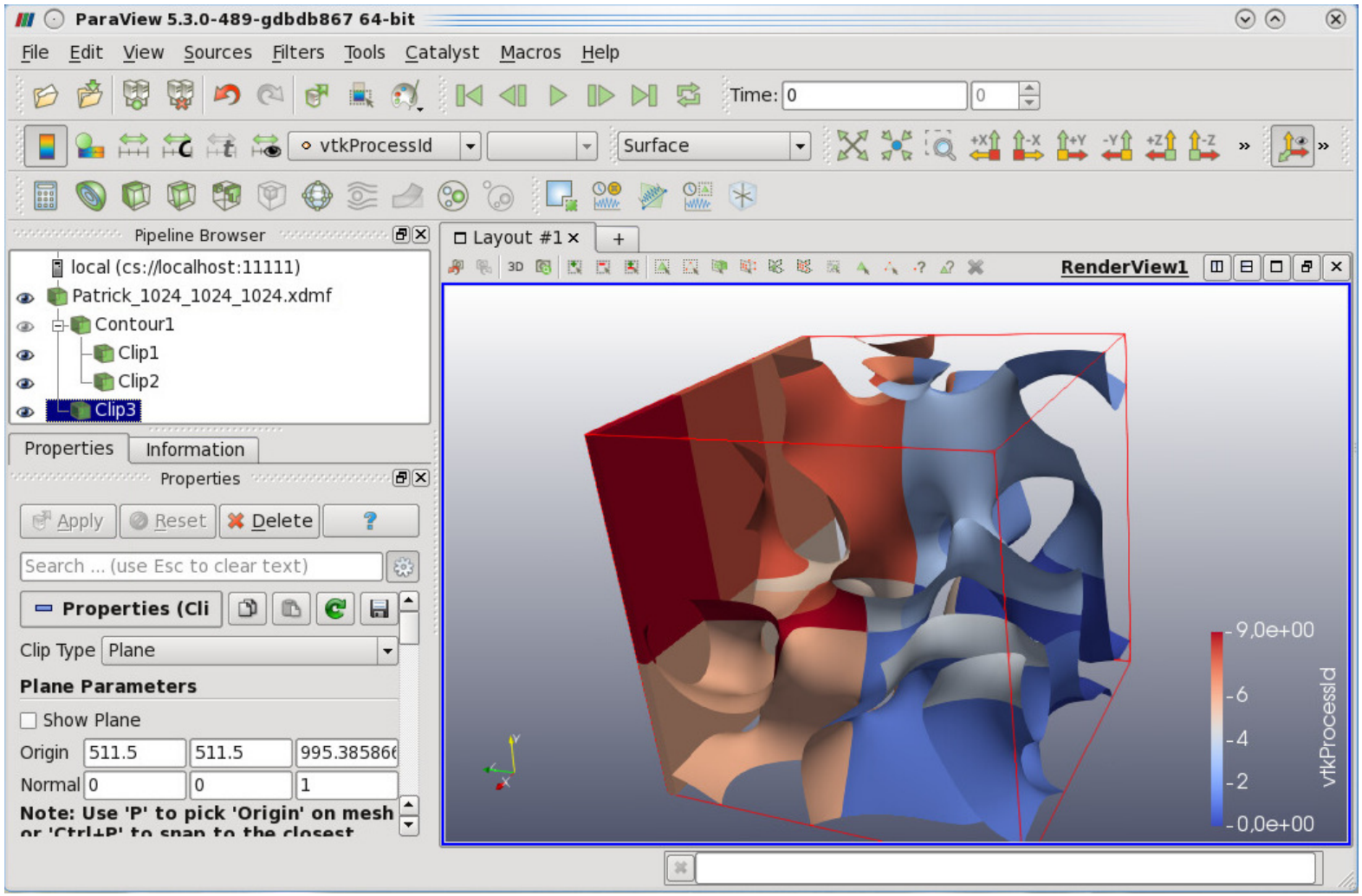
- Client and servers (same node) are connected
- Every paraview processing is now parallel

Let's talk parallel



- Parallel access to the data files for all the pvservers
- Rendering is parallel
- Blocs are colored by the pvserver id to show the distribution

Let's talk parallel



- Isosurfaces, sections. . . computations are done in parallel on the server and are faster
- Isosurfaces are colored by the pvserver id to show the parallel setup

Let's talk parallel

Bu

Poste utilisateur

```
build-charlie-manual : mpirun
File  Edit  View  Scrollback  Bookmarks  Settings  Help
bash-4.1$ mpirun -np 10 pvserver
Waiting for client...
Connection URL: cs://charlie.legi.grenoble-inp.fr:11111
Accepting connection(s): charlie.legi.grenoble-inp.fr:11111
Client connected.
Exiting...
Exiting...
Exiting...
Exiting...
Exiting...
Exiting...
Exiting...
Exiting...
```

...ie-manual : paraview

Fat node graphique
512Go RAM / 20 cores

Shutdown of the pvservers

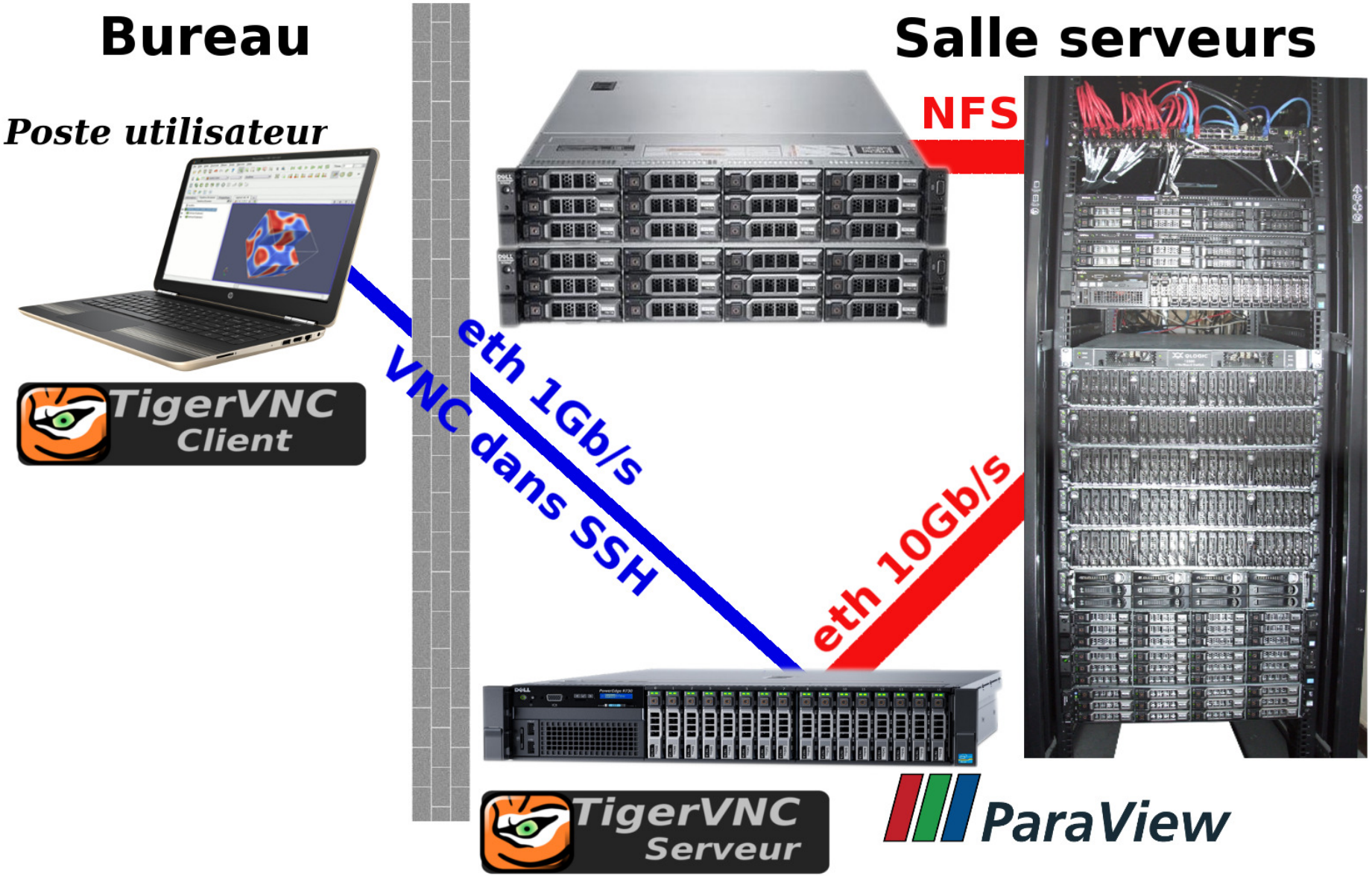
- Disconnecting the server or closing the GUI ends all the pvservers processes
- **TIP:** It is possible to start the paraview client on the laptop without the VNC session but...

But. . .

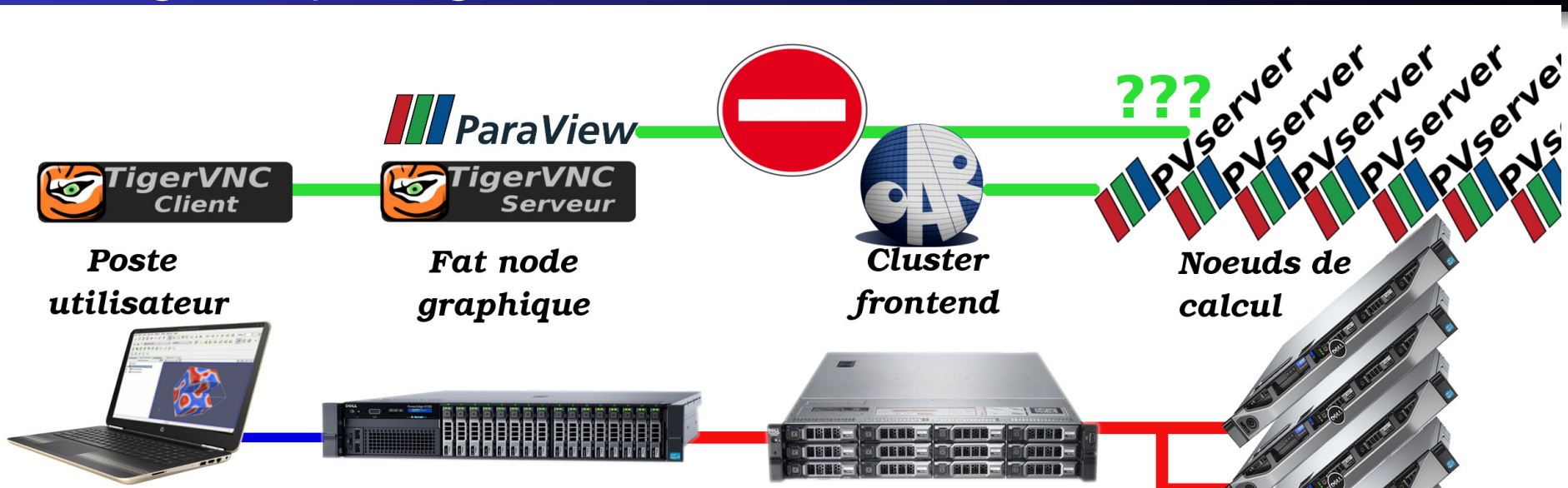
. . . and when the data becomes really big ?

- How to aggregate more resources?
- How to use the cluster nodes ?
- Where are the limits ?

Using computing nodes on the cluster



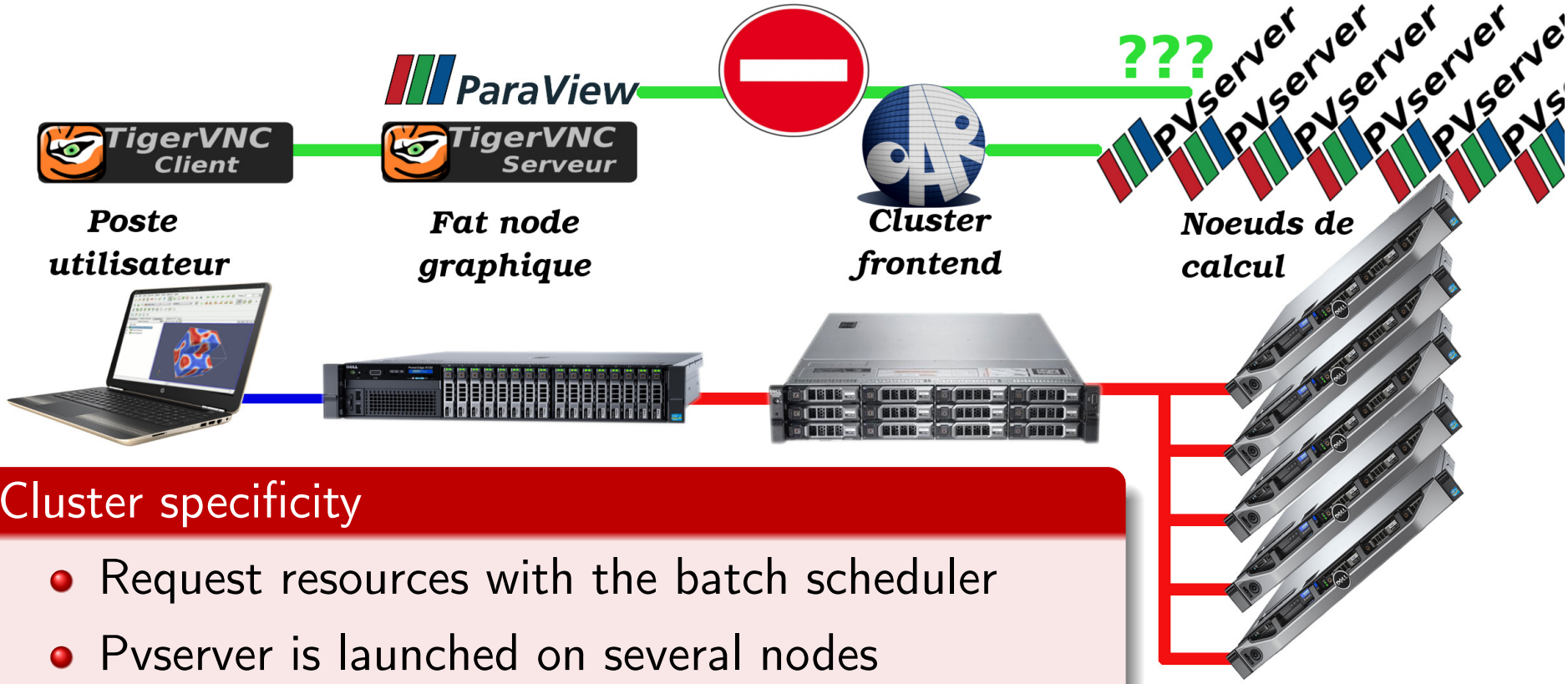
Using computing nodes on the cluster



Cluster specificity

- Request resources with the batch scheduler
- Pvserver is launched on several nodes
- Nodes are hidden behind the frontend

Using computing nodes on the cluster



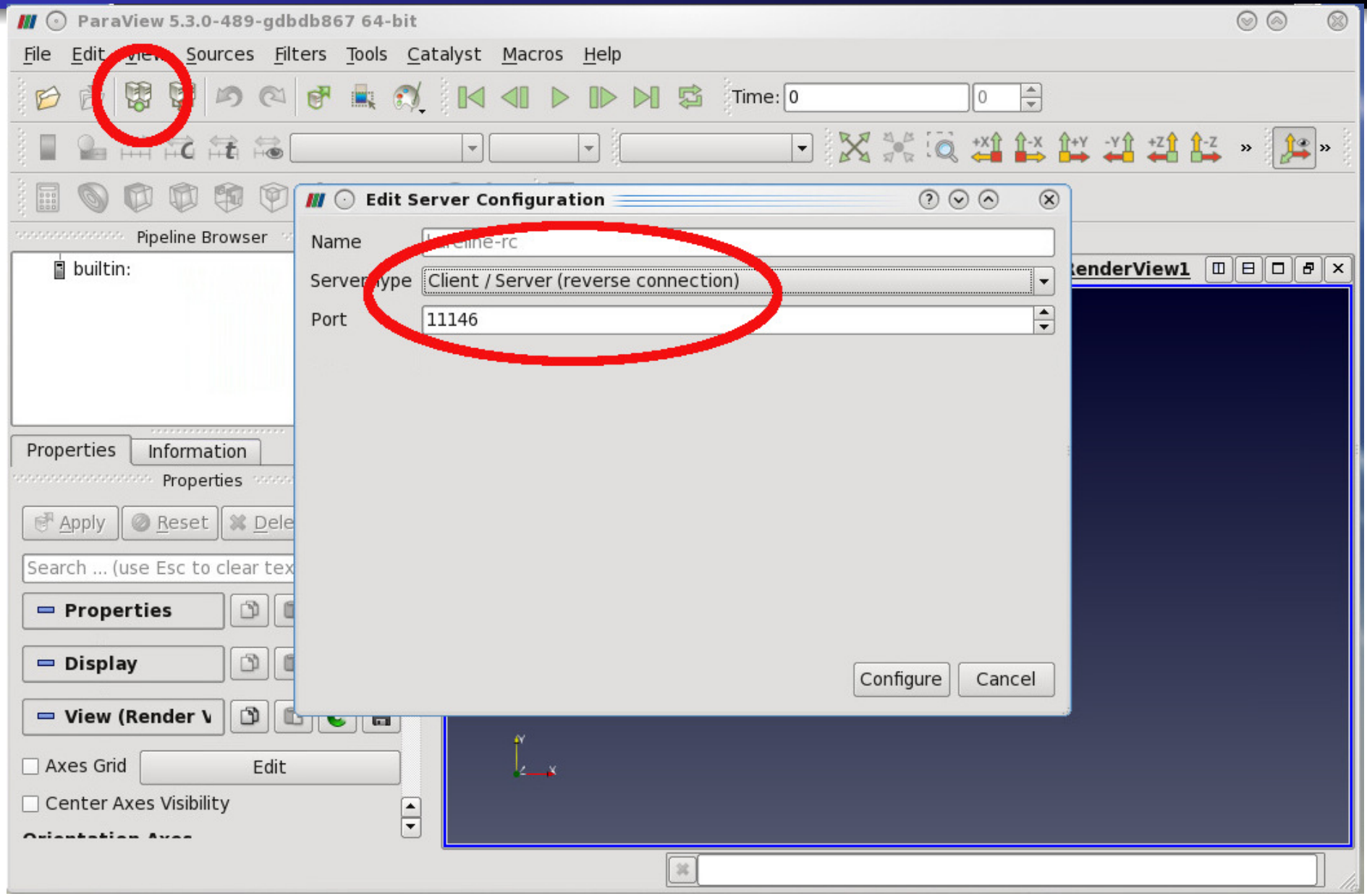
Cluster specificity

- Request resources with the batch scheduler
- Pvserver is launched on several nodes
- Nodes are hidden behind the frontend

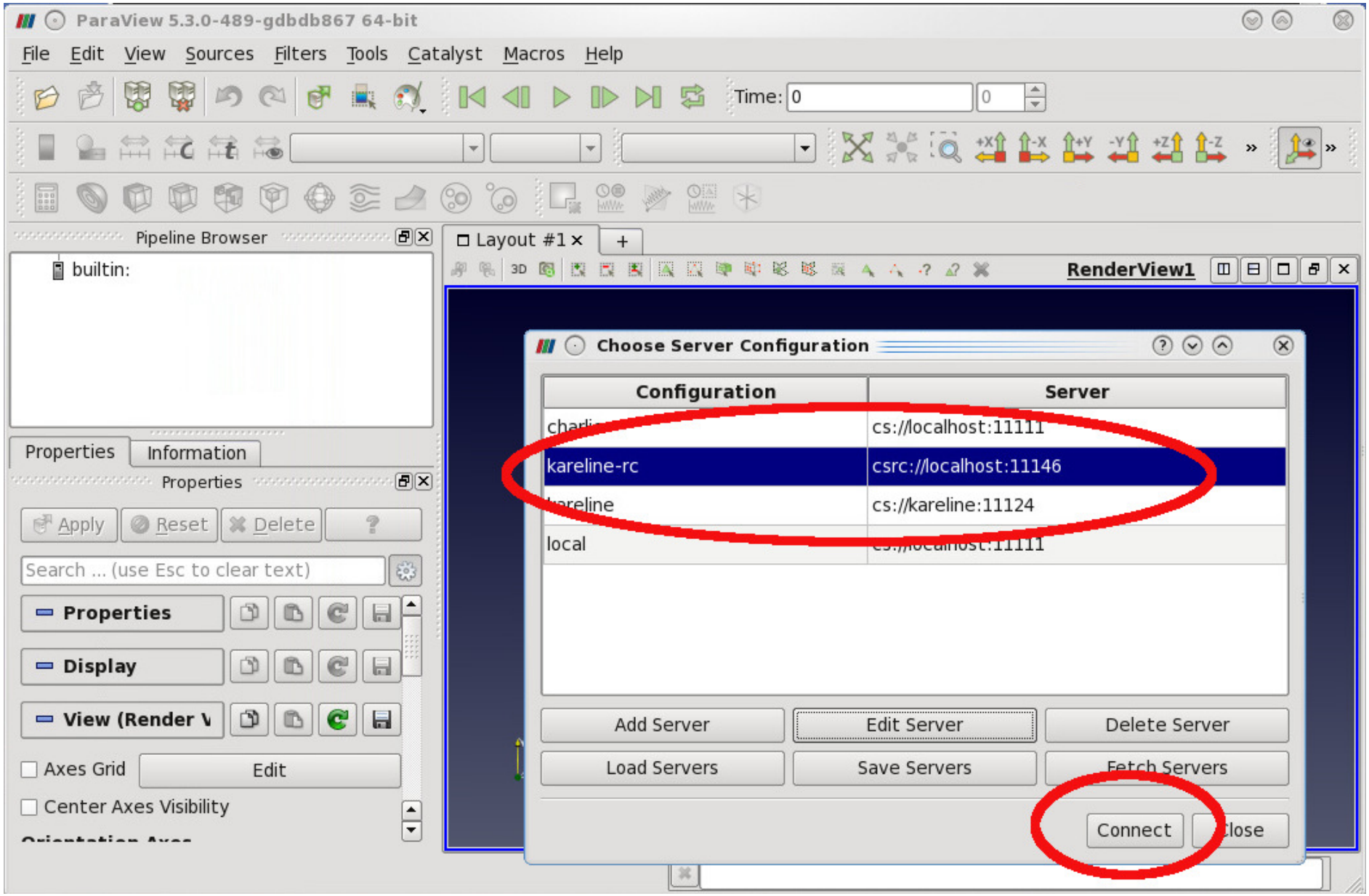
Solution: reverse connection

- pvservers processes connect to the Paraview client
- **mpirun -np 64 pvserver -rc -ch=foo.bar.fr -sp=11146**

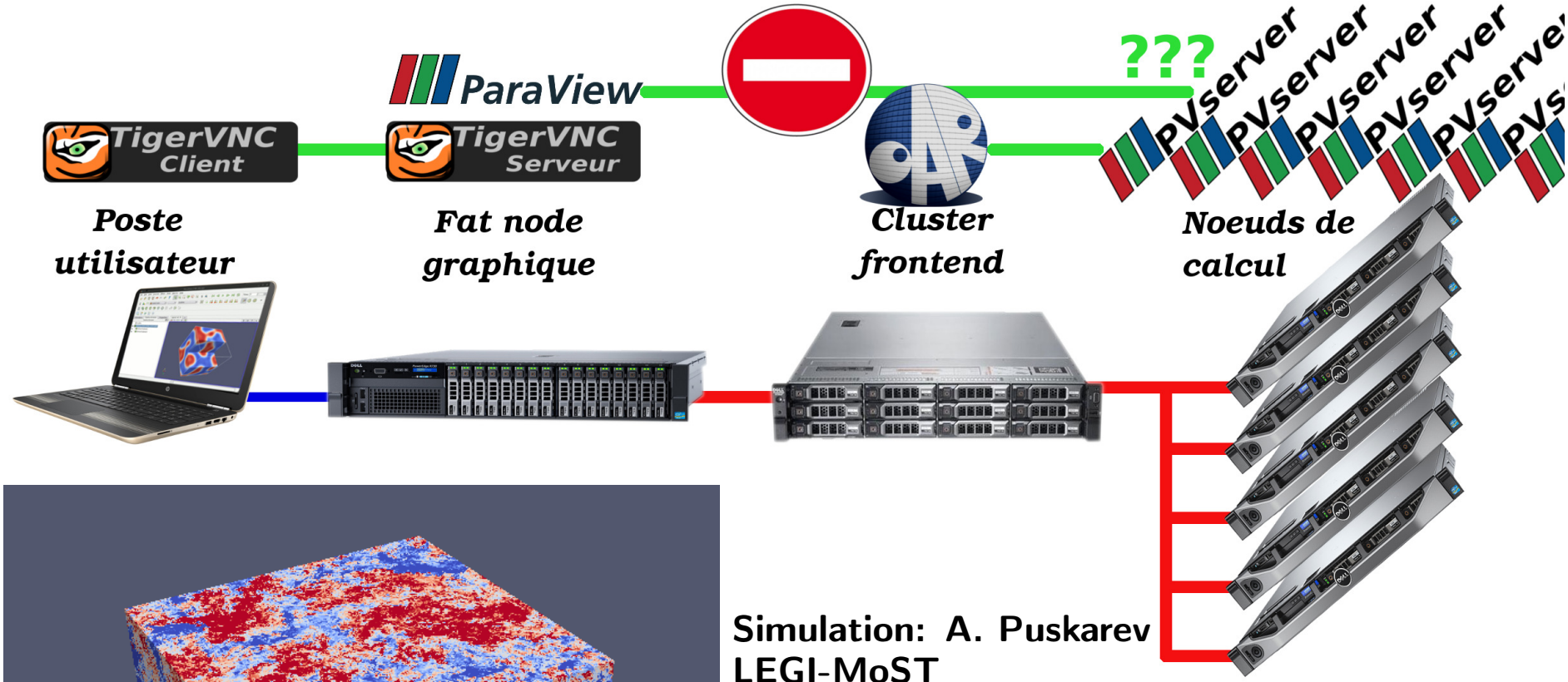
Reverse connection set up on the client side

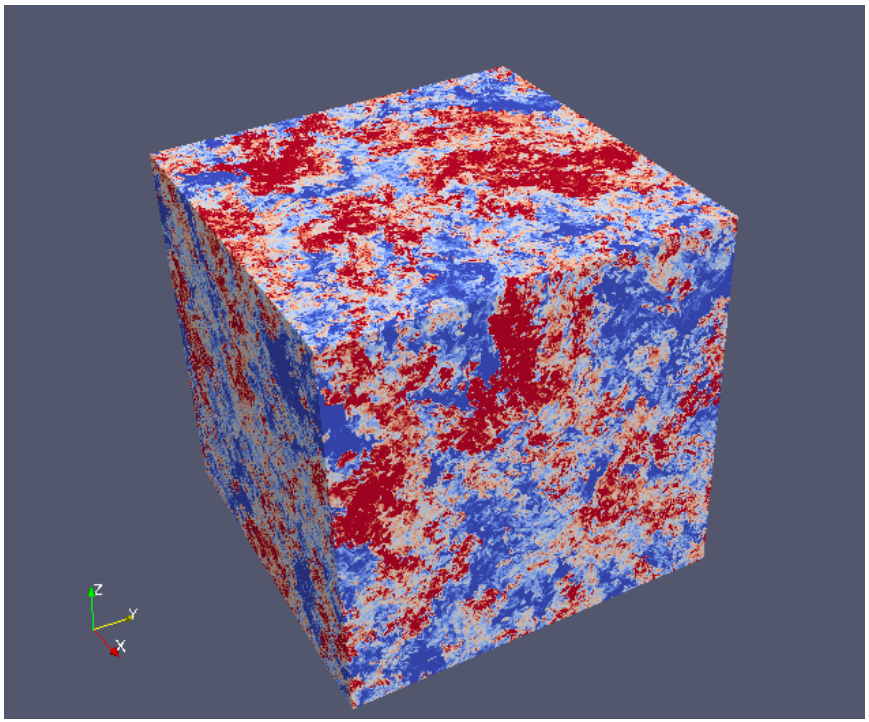


Reverse connection set up on the client side



Large data set visualization

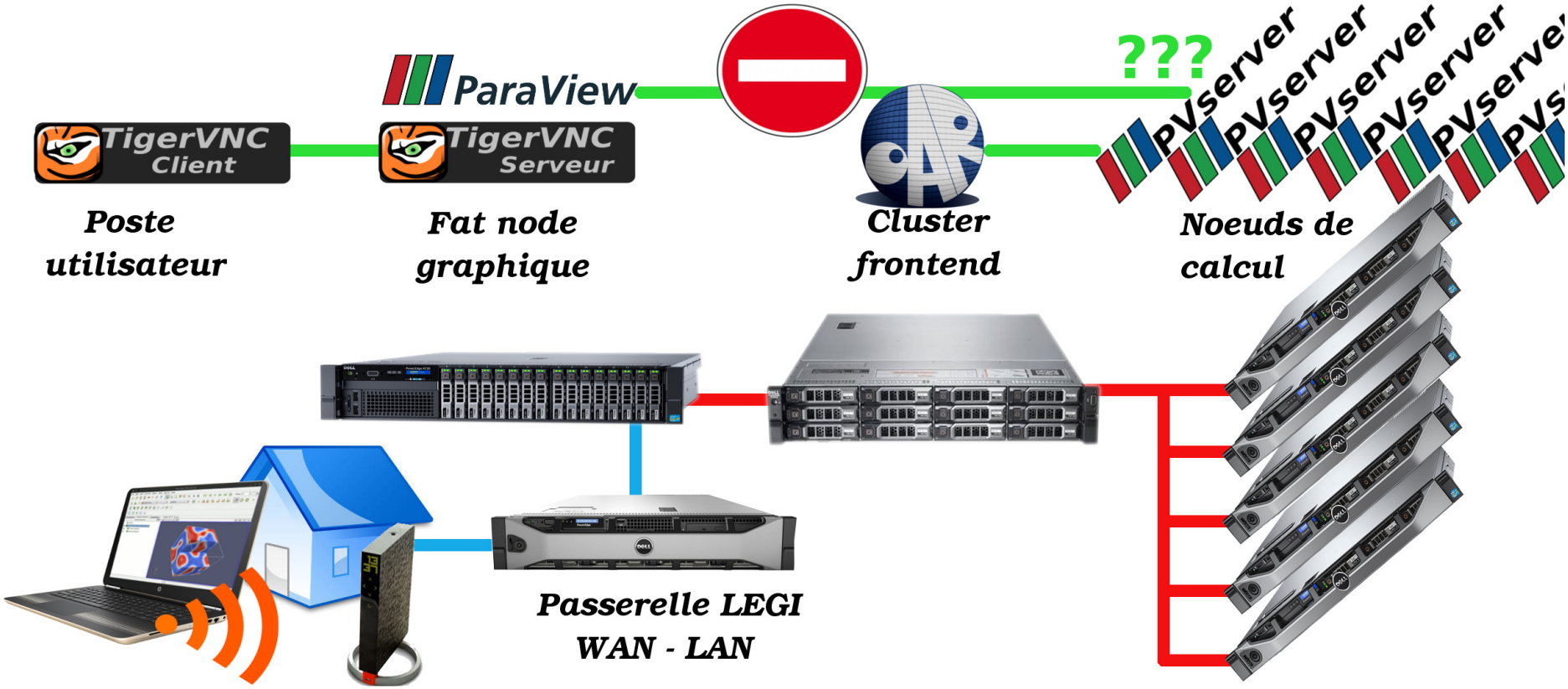




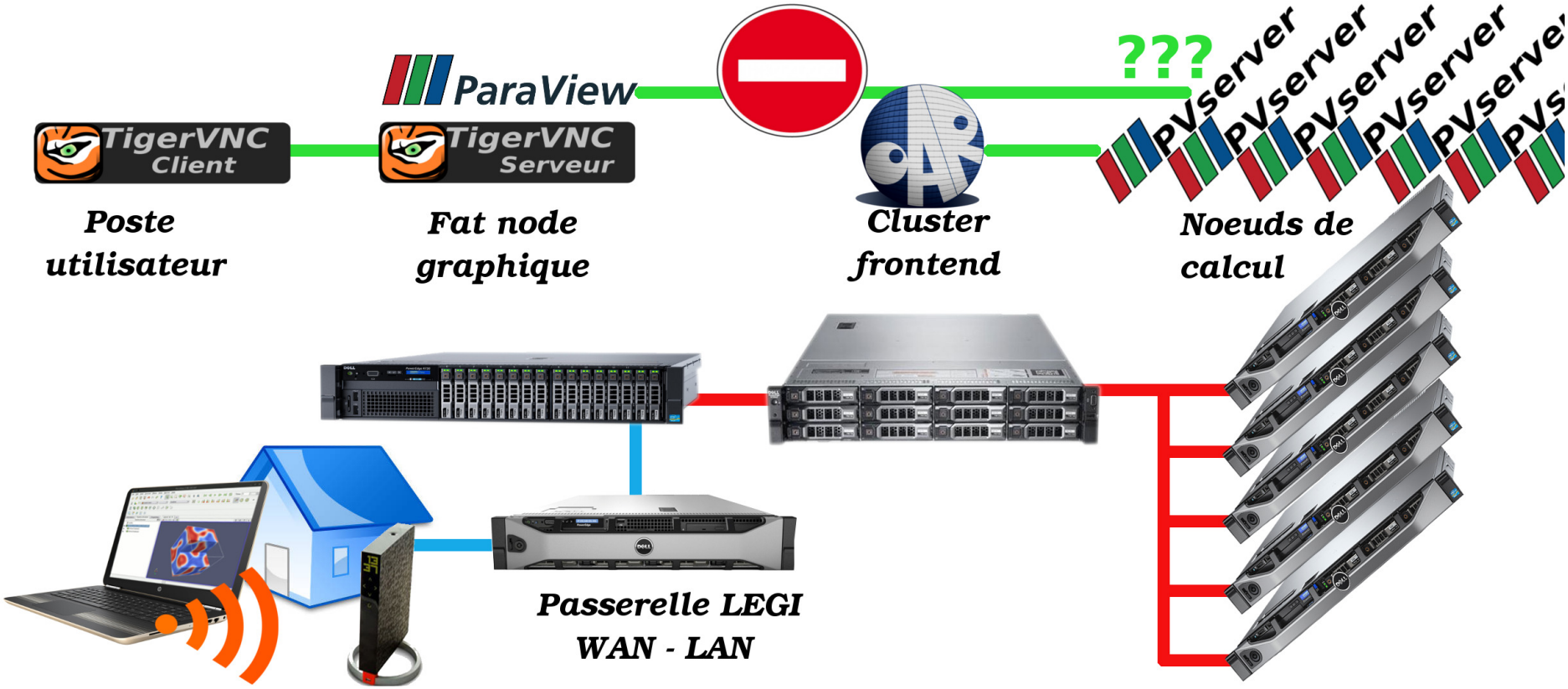
Simulation: A. Puskarev
LEGI-MoST

- Three-dimensional Scalar field, velocities, Q Criterion...
- 8,6 billions cells geometry
- 65GB $\times$ 5 fields = 325GB to load
- 14 nodes / 280 pvserver processes

But can I do this from home now ?

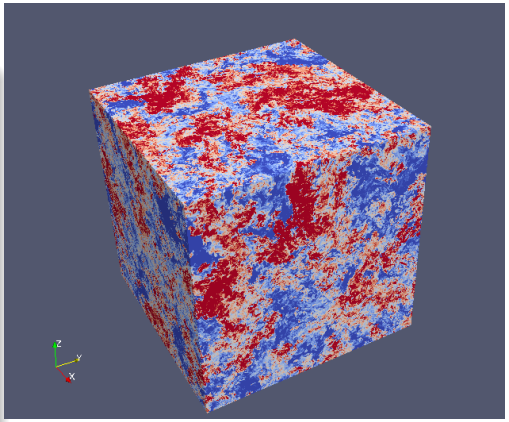


But can I do this from home now ?



Yes, it works!

- Option -**via** when launching tigerVNC client
- Automatic set up of the ssh tunnel from the laptop to the fat node



But it is not a bed of roses tough...

Loading data takes a lot of time

- **280** `PVServers` processes simultaneously access the file server !
- Loading **326GB** in small pieces from a file server with only 32GB of RAM
- Nearly **40mn** needed to map the data in RAM on the `PVServers`

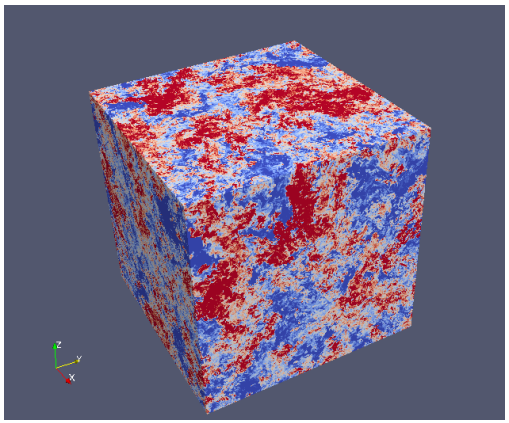
But it is not a bed of roses tough...

Loading data takes a lot of time

- **280** `PVServers` processes simultaneously access the file server !
- Loading **326GB** in small pieces from a file server with only 32GB of RAM
- Nearly **40mn** needed to map the data in RAM on the `PVServers`

Possible improvements

- Preload of the data on each node's disks
- Loading time falls to **10mn** (only **20 concurent processes** for the I/Os)
- Performances could be improved with fastest devices (SSD, burst-buffer...) or a more powerful file server



But it is not a bed of roses tough...

Sometime rendering could be slow

- No GPU available on the cluster nodes
- Rendering is done by the CPU

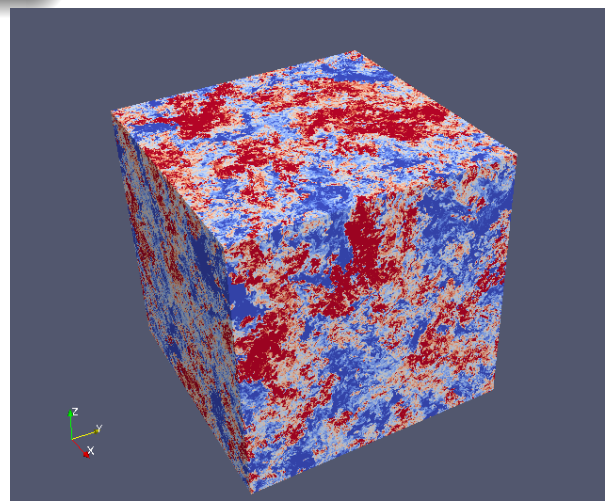
But it is not a bed of roses tough...

Sometime rendering could be slow

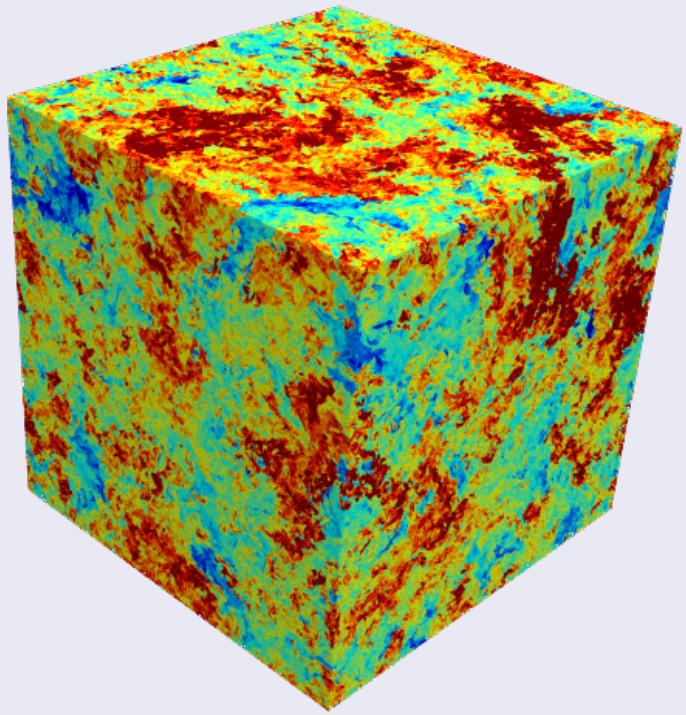
- No GPU available on the cluster nodes
- Rendering is done by the CPU

Possible improvements

- Using nodes with GP-GPU devices (Nvidia, AMD...) to speed up rendering



Large data visualization becomes possible on the laptop



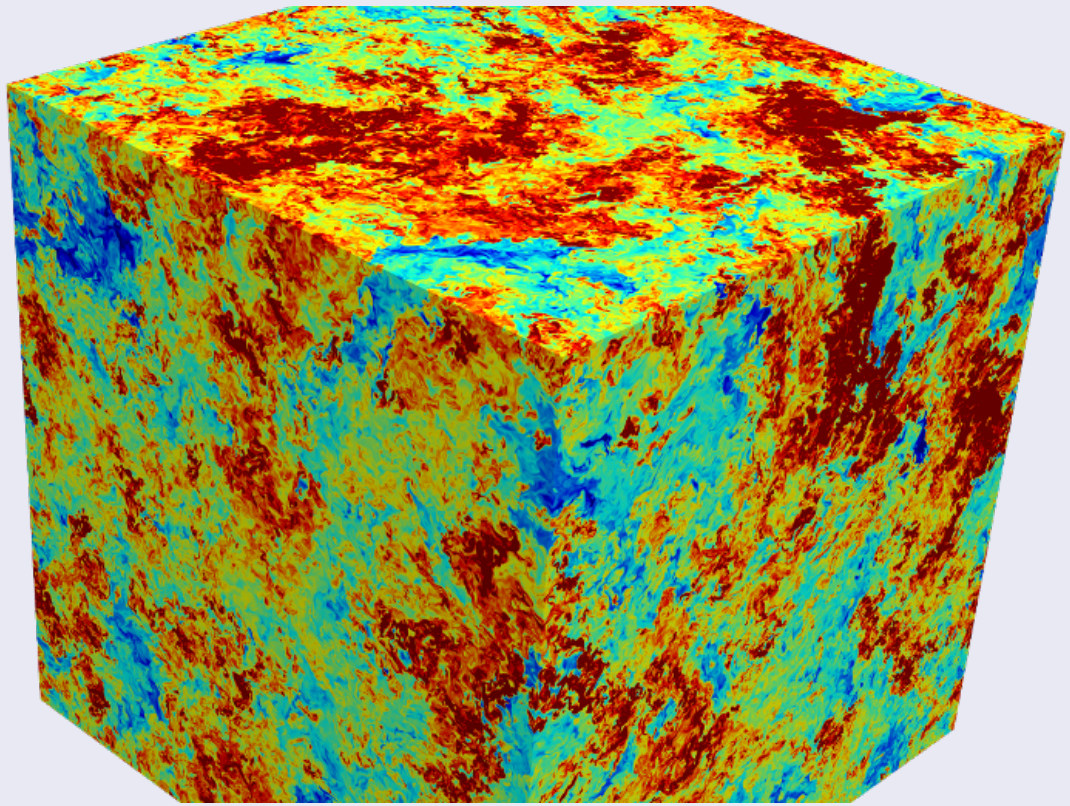
Dive into the details!

However...

This workflow opens access to [easy] visualization of [very] large datasets

- Without requiering powerfull desktop on each user desk
- Using existing computational resources in the data-center

Large data visualization becomes possible on the laptop



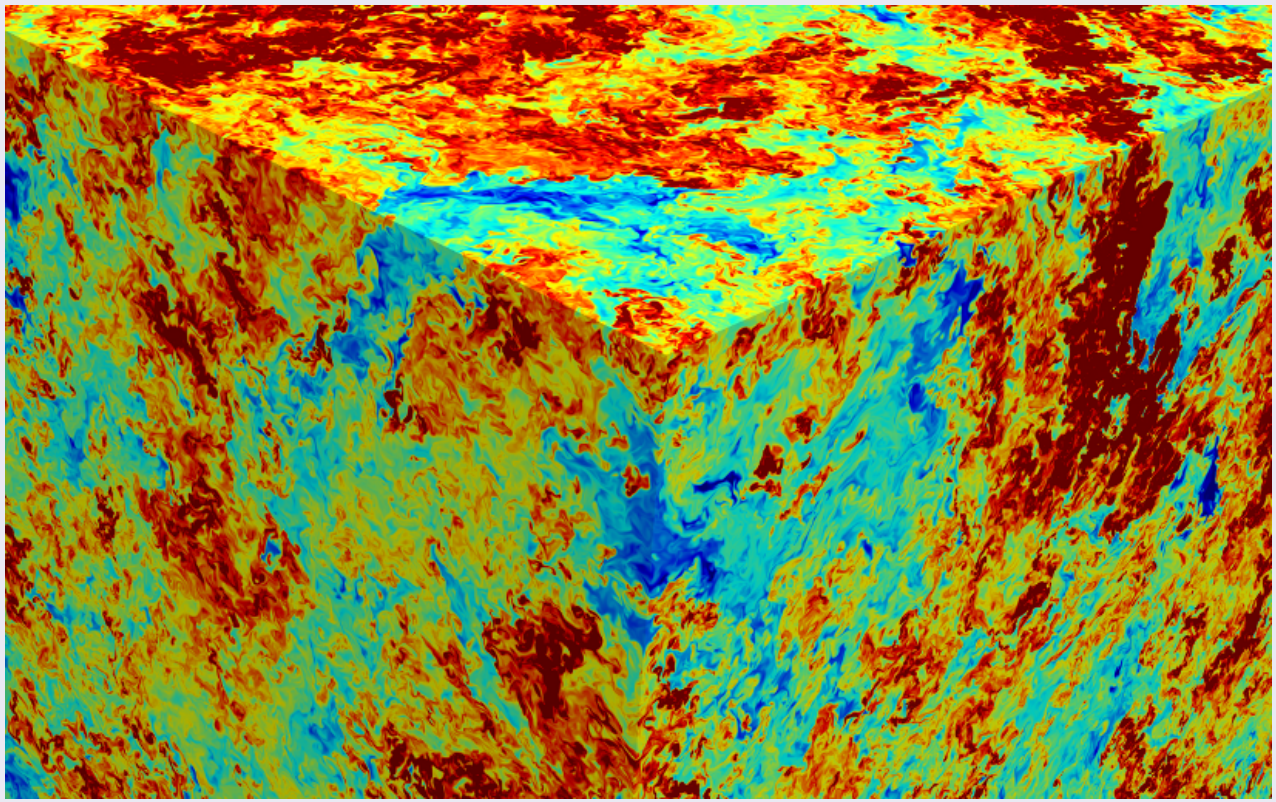
Dive into the details!

However...

This workflow opens access to [easy] visualization of [very] large datasets

- Without requiering powerfull desktop on each user desk
- Using existing computational resources in the data-center

Large data visualization becomes possible on the laptop



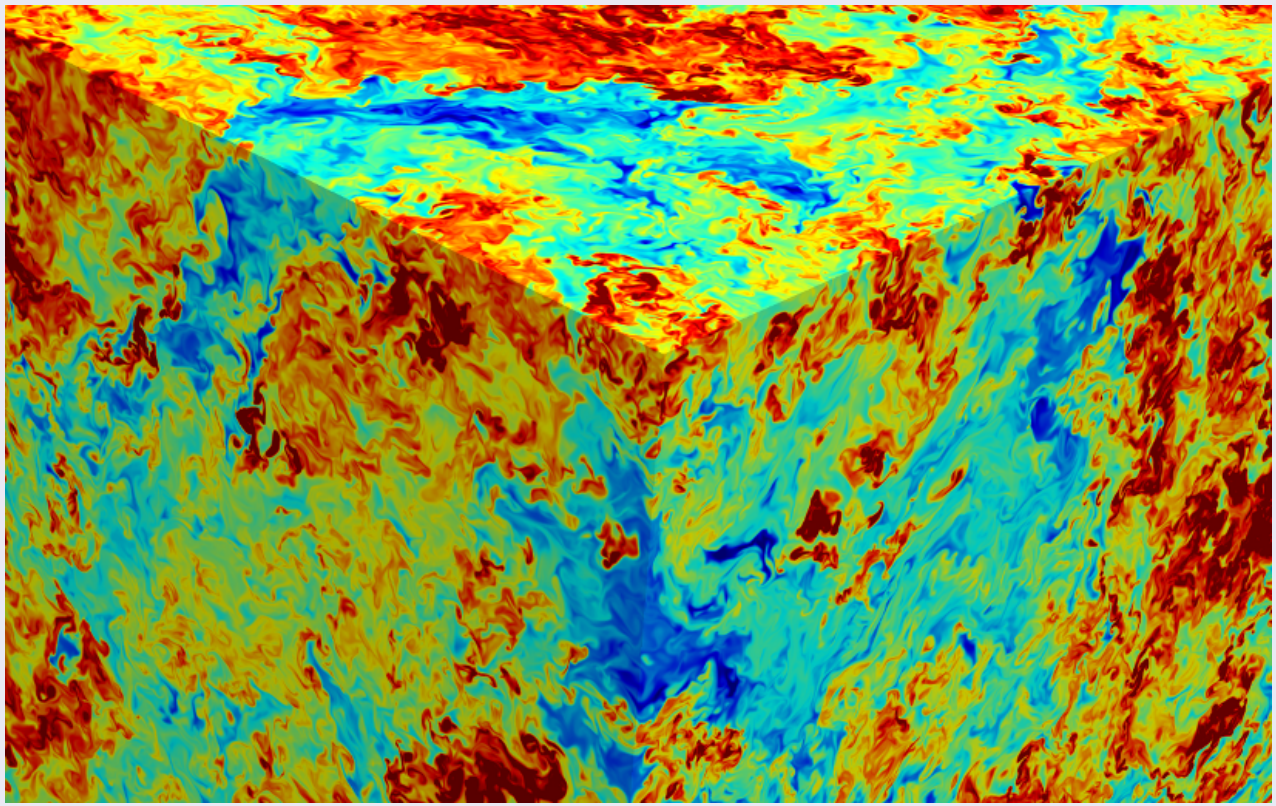
Dive into the details!

However...

This workflow opens access to [easy] visualization of [very] large datasets

- Without requiering powerfull desktop on each user desk
- Using existing computational resources in the data-center

Large data visualization becomes possible on the laptop



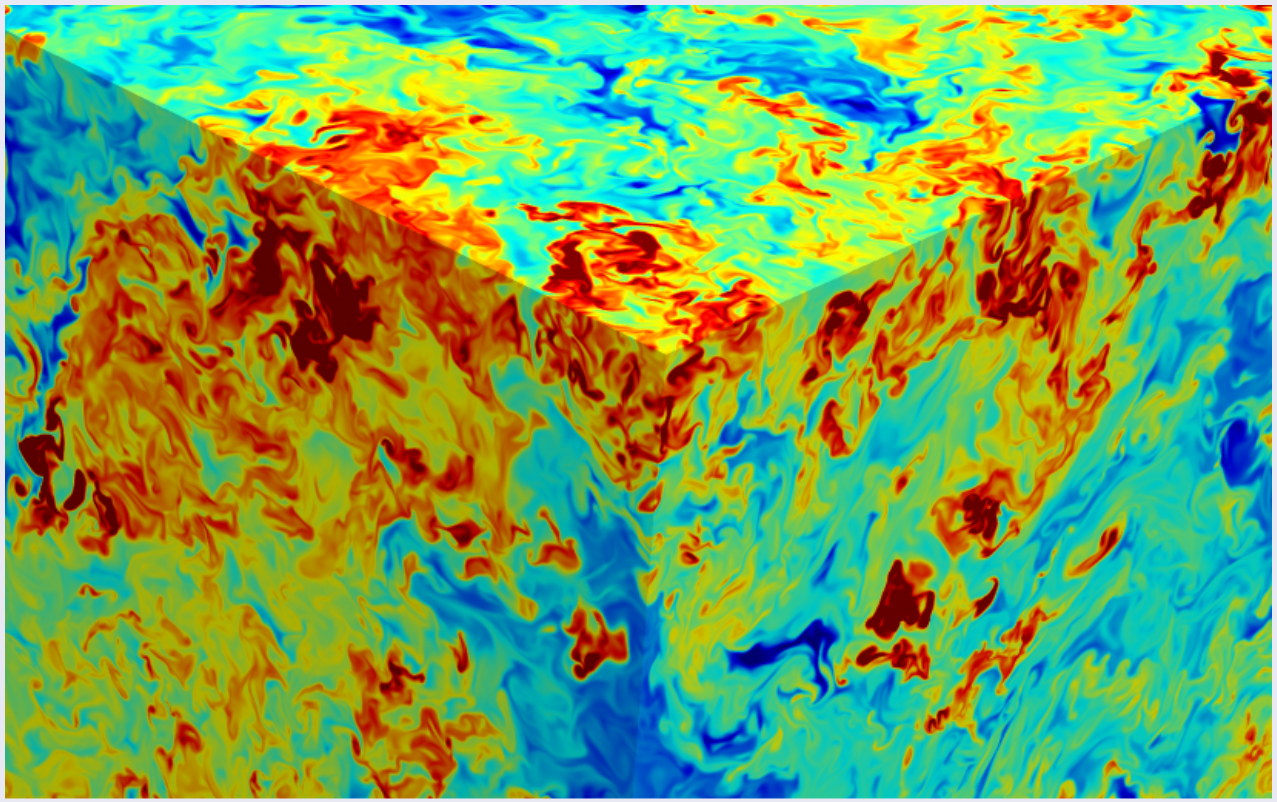
Dive into the details!

However...

This workflow opens access to [easy] visualization of [very] large datasets

- Without requiering powerfull desktop on each user desk
- Using existing computational resources in the data-center

Large data visualization becomes possible on the laptop



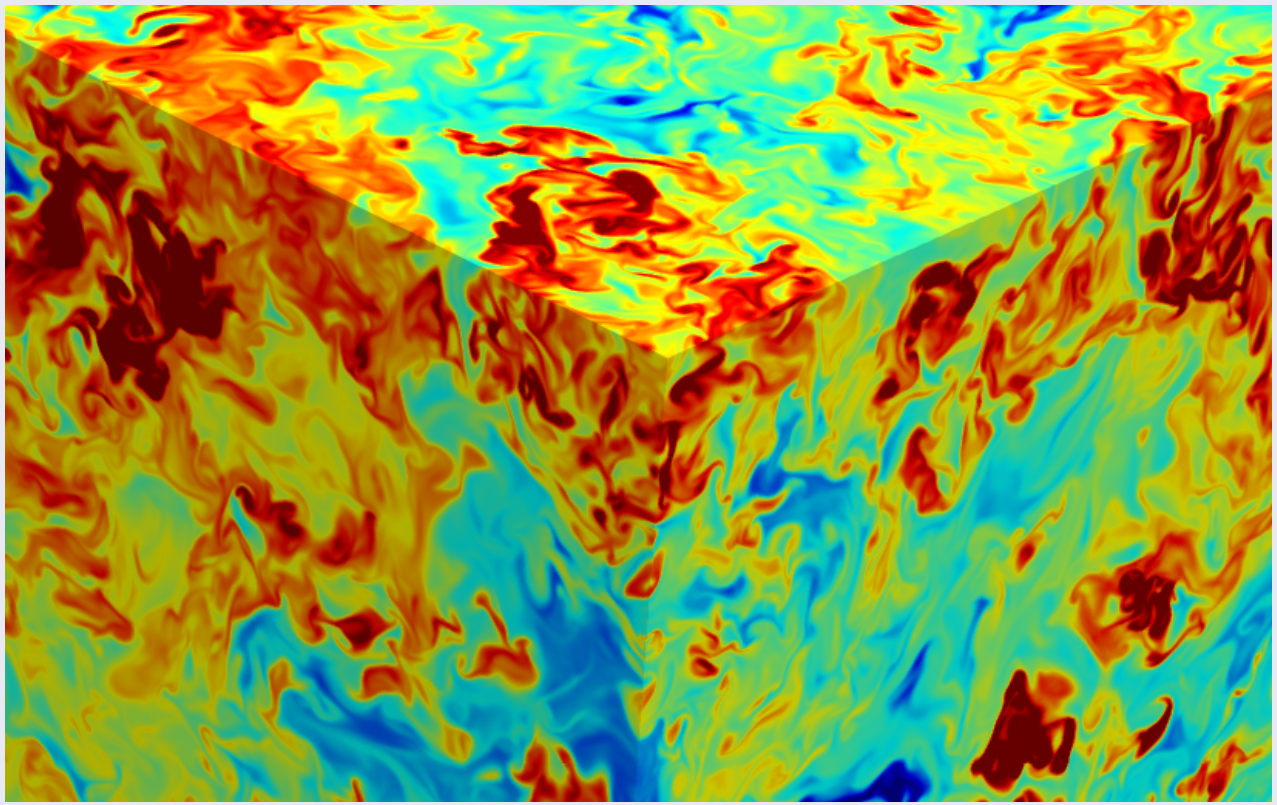
Dive into the details!

However...

This workflow opens access to [easy] visualization of [very] large datasets

- Without requiering powerfull desktop on each user desk
- Using existing computational resources in the data-center

Large data visualization becomes possible on the laptop



However...

This workflow opens access to [easy] visualization of [very] large datasets

- Without requiering powerfull desktop on each user desk
- Using existing computational resources in the data-center

Conclusions

- Paraview is an "open-source" powerful tool for scientific visualizations
- Highly parallel client/serveur simple setup for large datasets
- Associated to TigerVNC these tools provide flexibility and ubiquity for scientific visualization
- Similar solutions are available at CIMENT mesocenter (Grenoble), at TGCC/CEA...

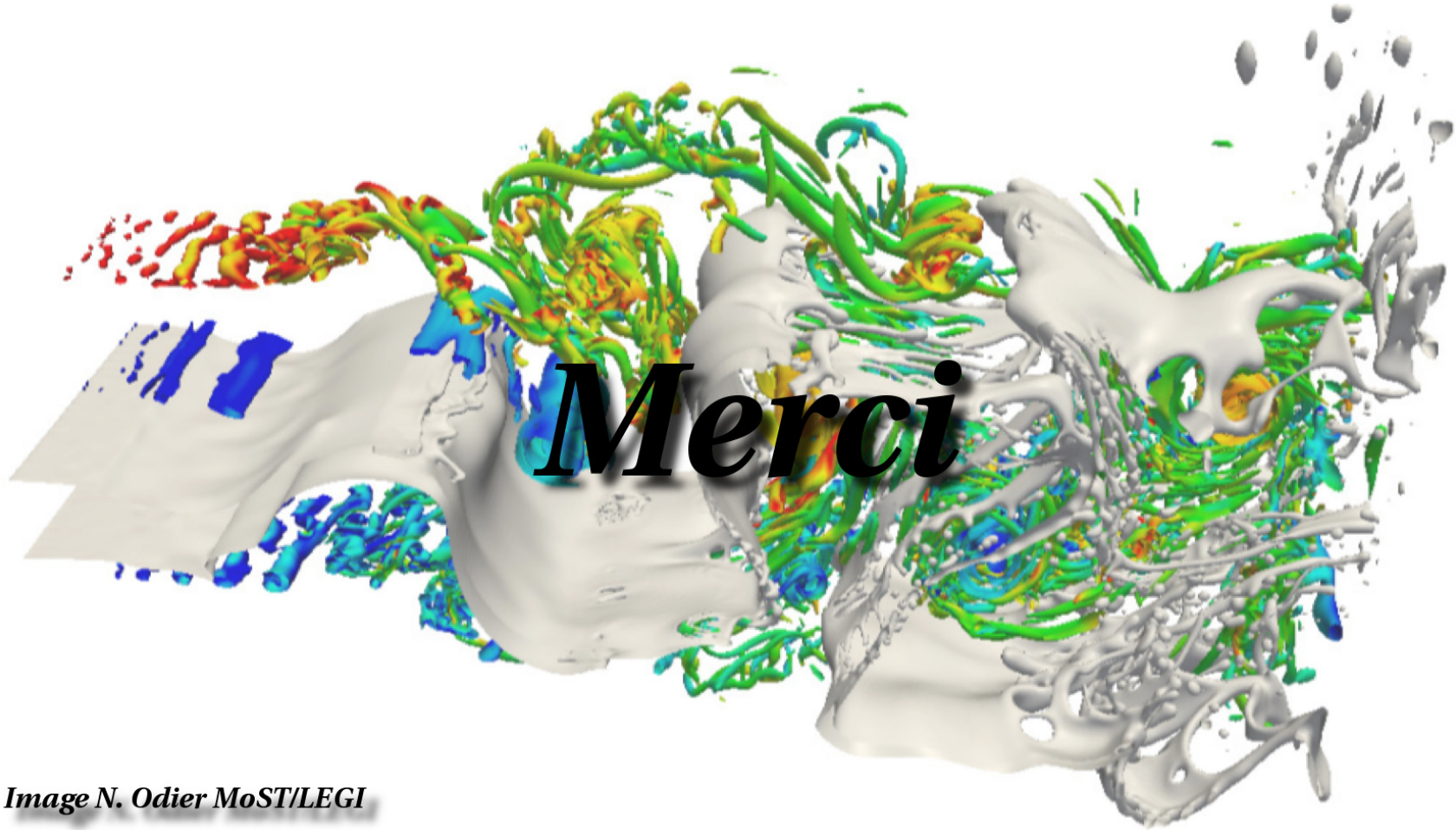


Image N. Odier MoST/LEGI