A Flexible Framework for Asynchronous In Situ and In Transit Analytics for Scientific Simulations

Dreher Matthieu, Bruno Raffin

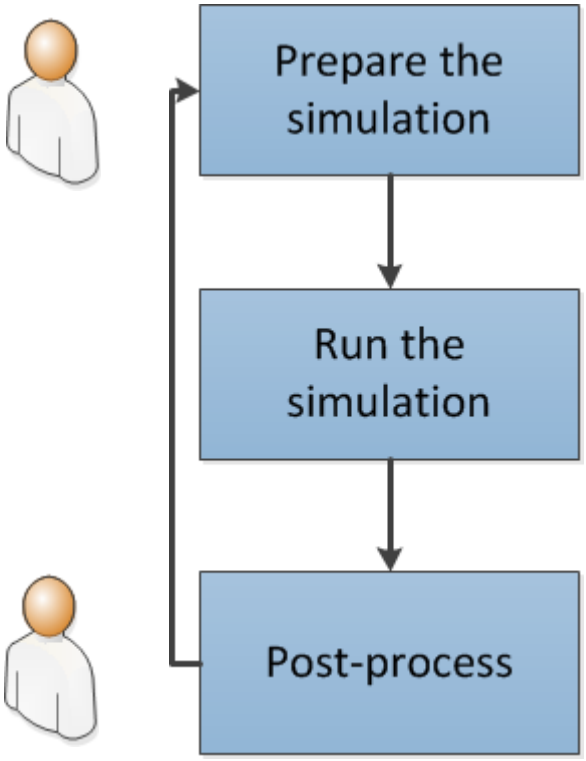
14 MAY 2014

Summary

1

Introduction to In Situ/In Transit concepts

Traditionnal scientist workflow



Model, configuration files, etc

Hours, days, week on parallel machine

Data transfert, analyse, etc

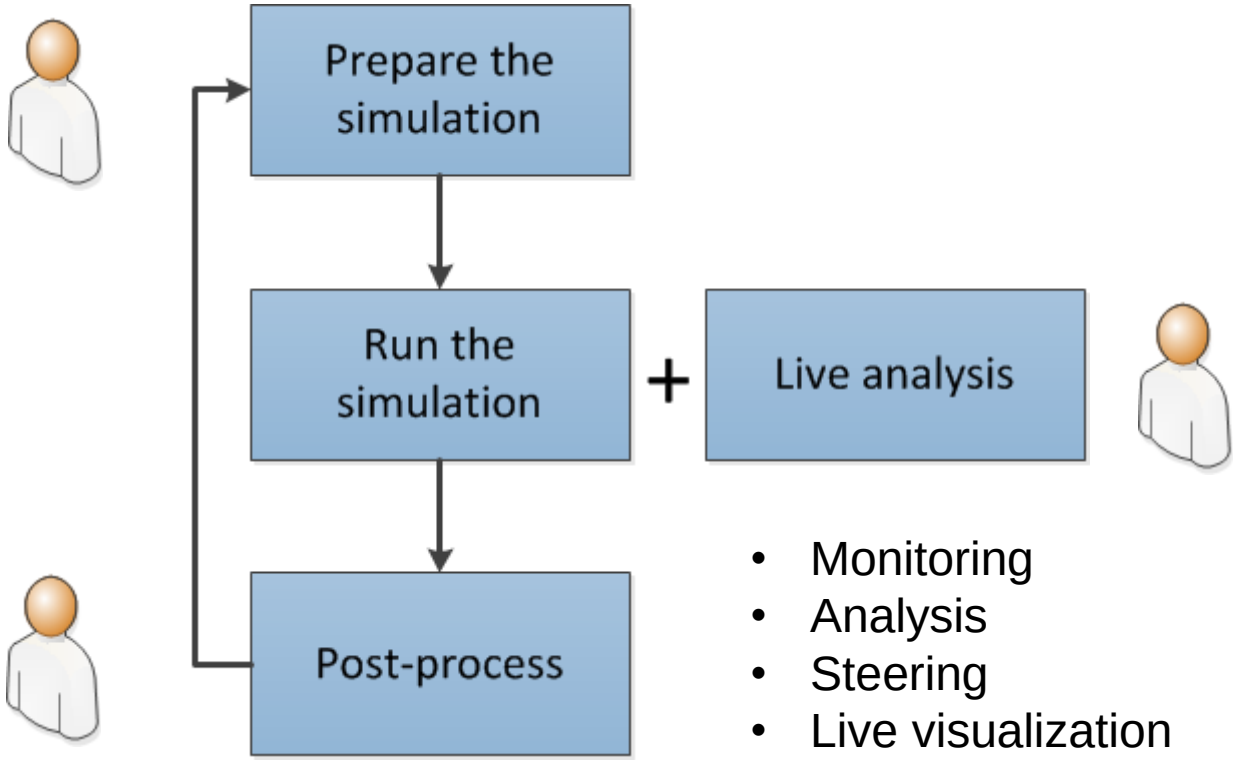
But what if something goes wrong on the way?

Problematics around the simulation

- The simulation is a black box
- Simulation well configured? well constrained? Unexpected event?
- Waste of computationnal time and ressource
- Data managment (size, transfert at the lab)
- Might not scale well on recent architectures

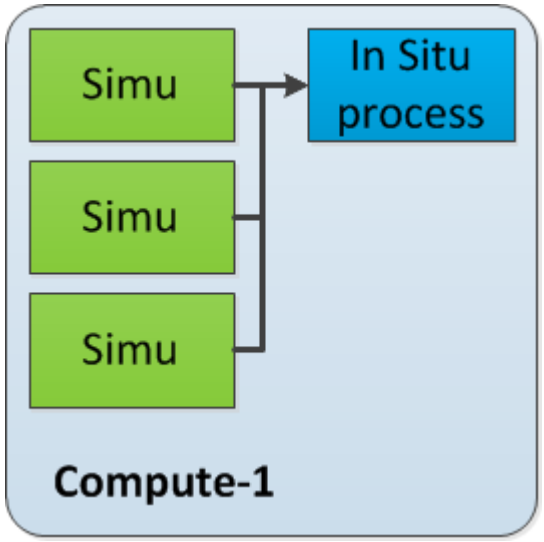
How to reintegrate the user into the simulation?

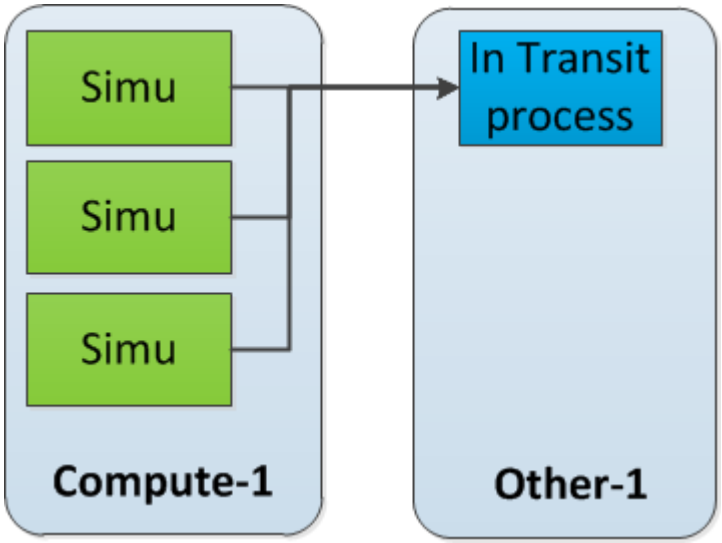
Scientist workflow with In Situ/In Transit capabilities



- Monitoring
- Analysis
- Steering
- Live visualization
- Advance writing
- …

In Situ and In Transit configurations





Pros :
- Data locality
- Fine grain parallelism

Cons:
- Concurrency on ressources

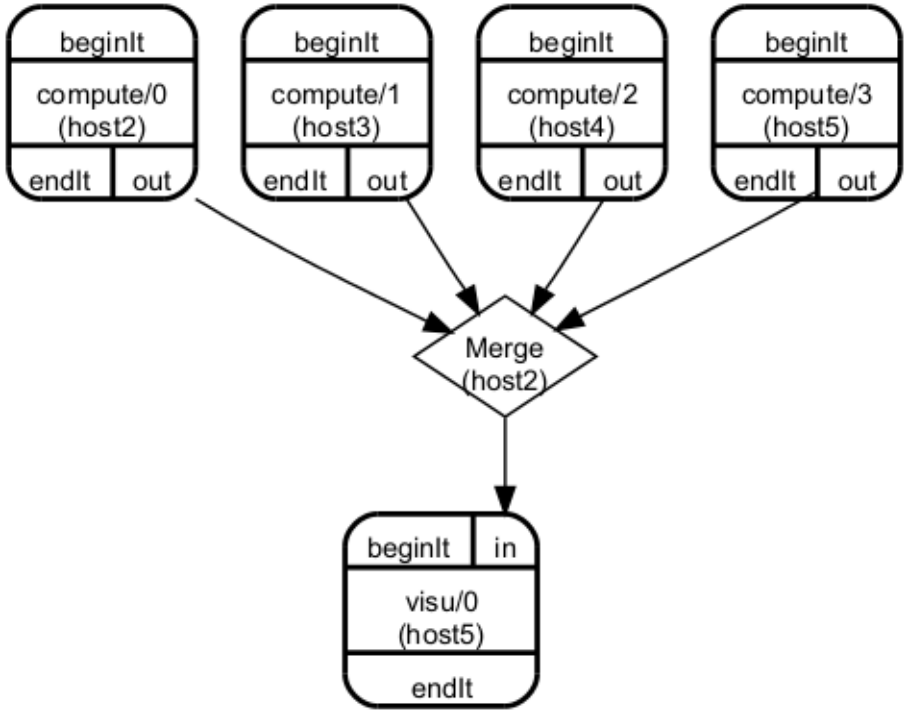
Pros :
- (Almost) No impact on the simulation ressources
- Lower parallelism

Cons:
- Extra ressources
- Require data transfert

2

The Vitamins Framework

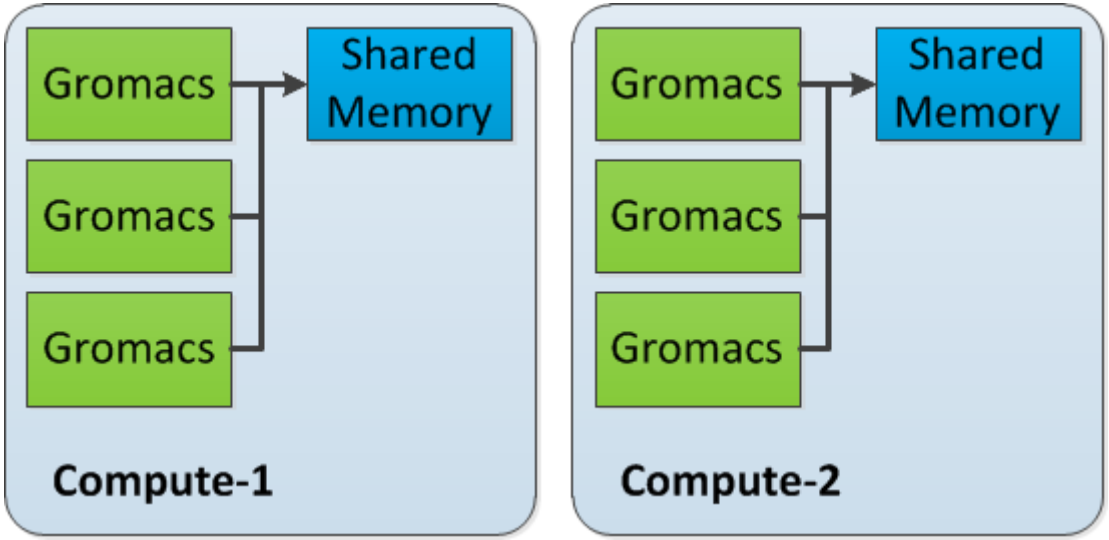
The FlowVR middleware



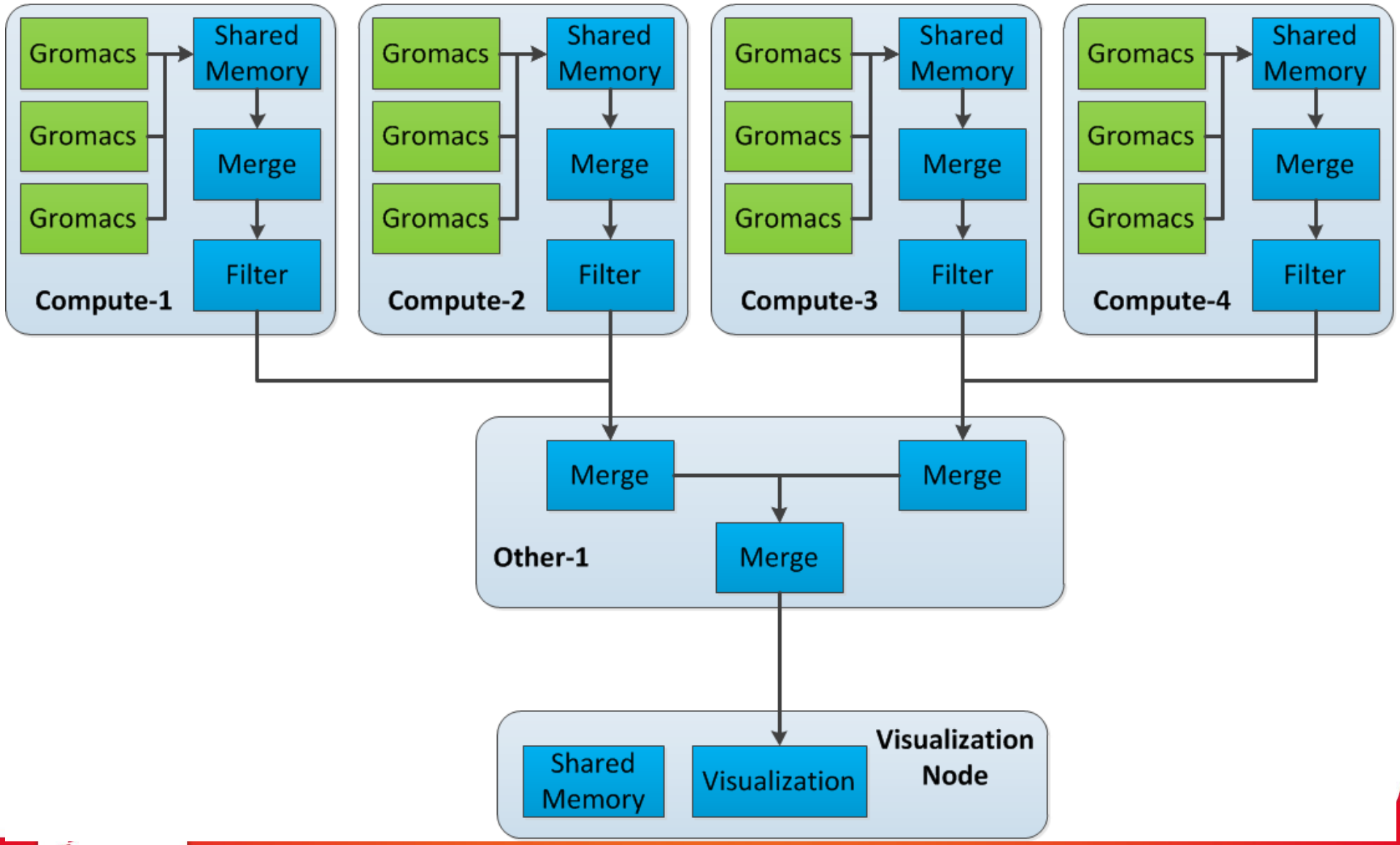
- Modules : indiviual codes encapsulated inside modules. A module can receive and send messages (C/C++/Python)
- Channel : Communication link between modules. By default FIFO but can create more complex policies. Message transport via shared memory or network.

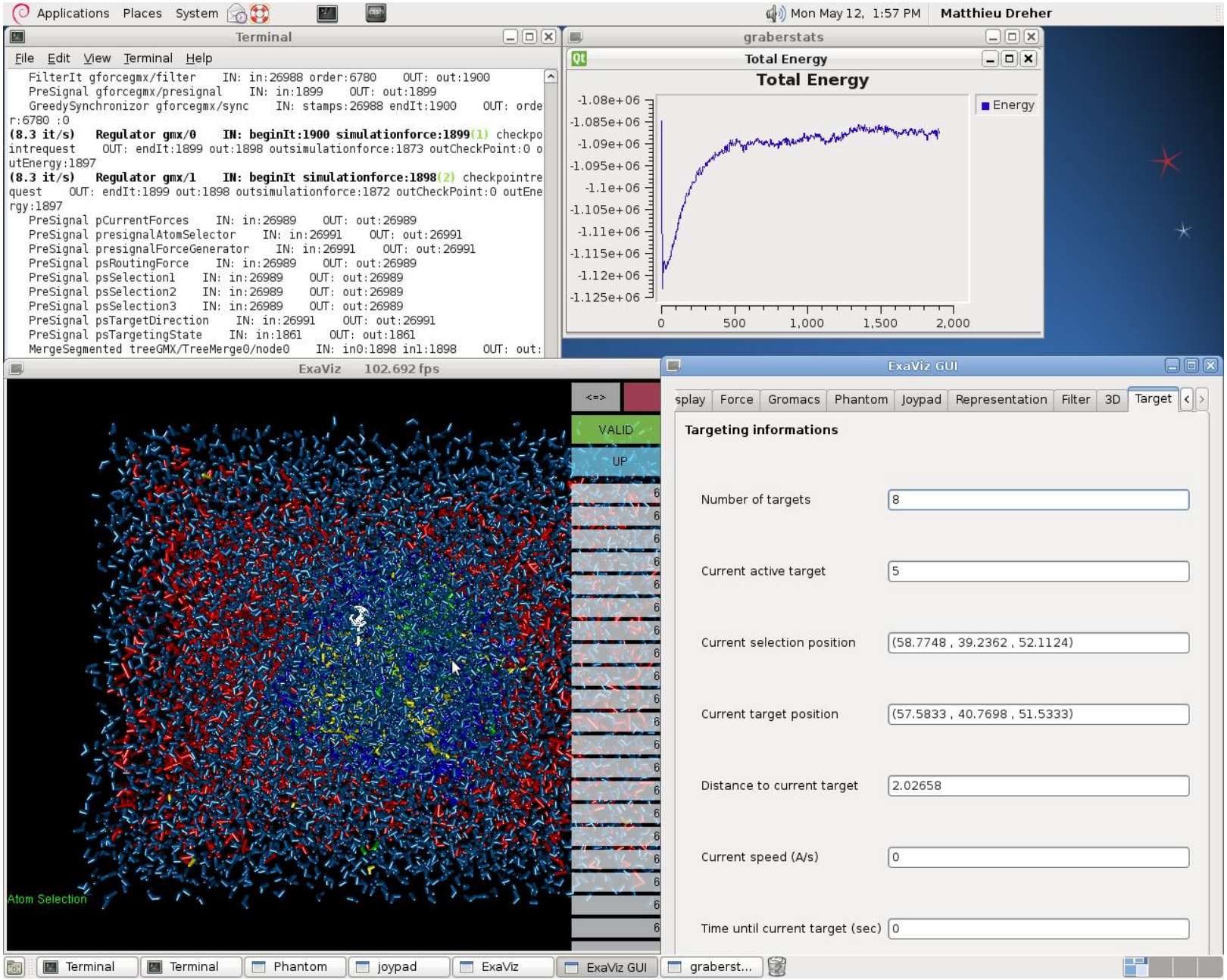
Focus on the Gromacs module

- Vitamins = collection of modules around the analyse of molecular dynamic simulation
- Gromacs is a scalable molecular dynamic simulation package.
- Hybrid parallelism with MPI + OpenMP + GPUs
- Scalability on several thousands of cores
- Instrumentation of the code for each MPI process (~50 lines)
- Asynchronous extraction of data

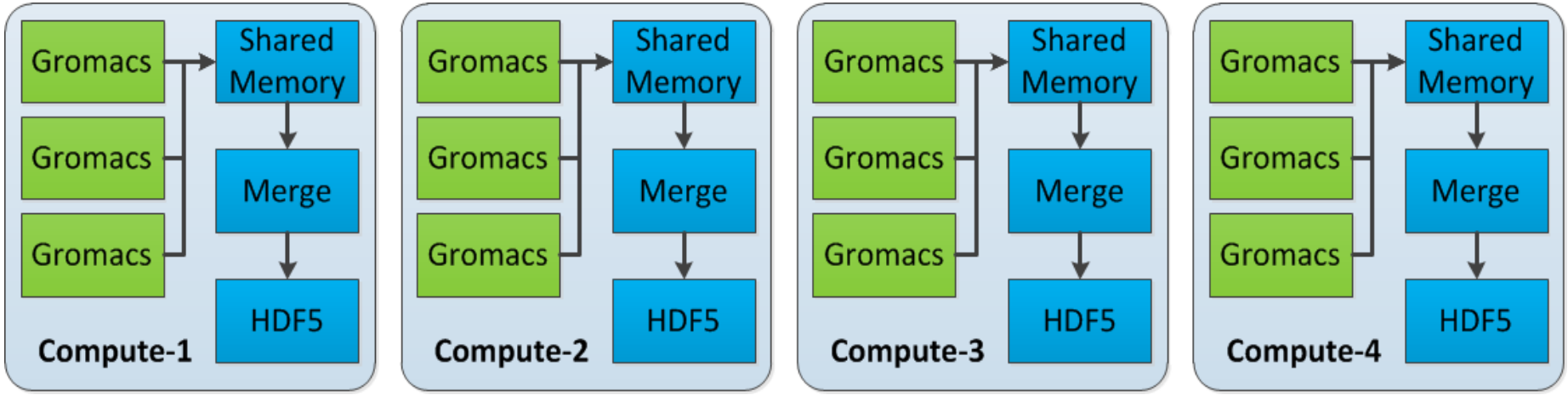


Live visualization exemple

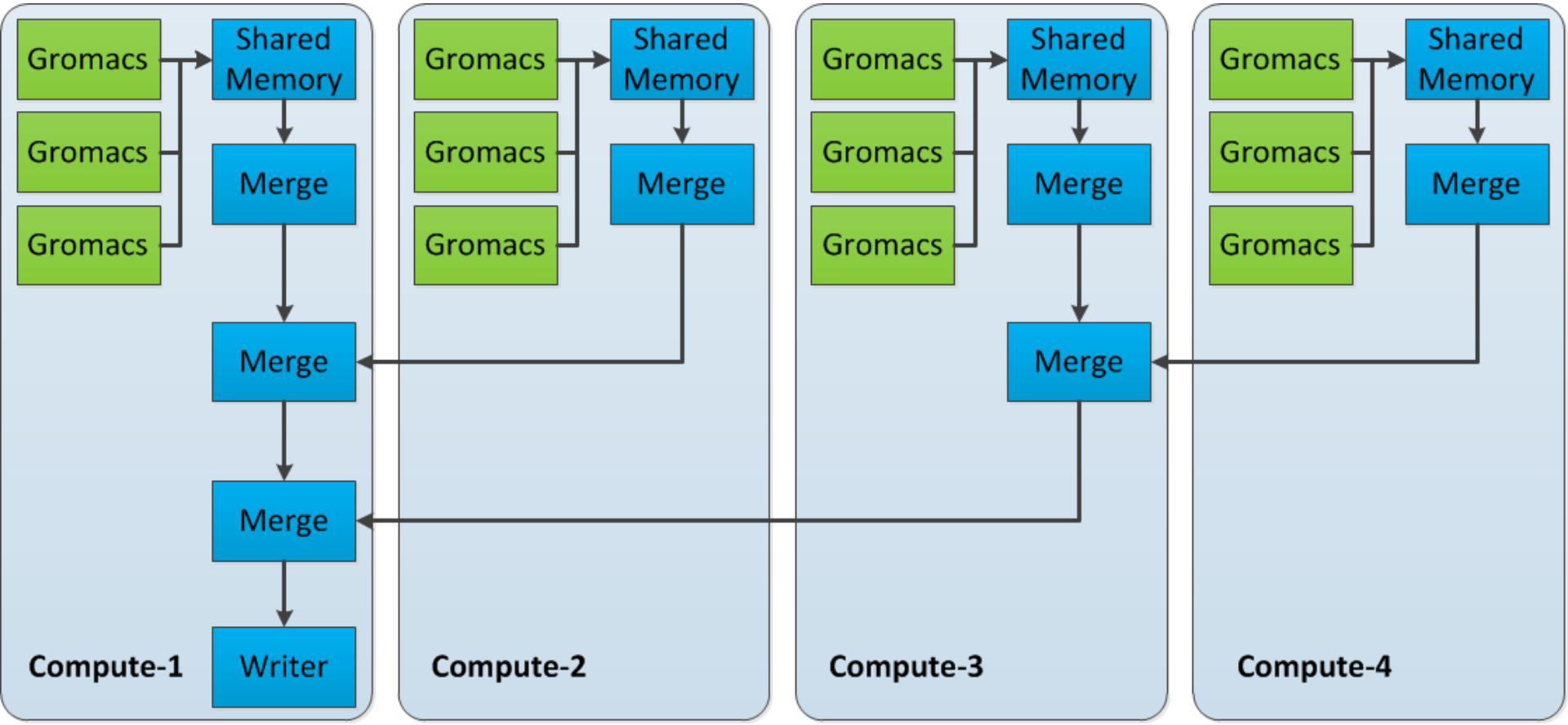



Applications Places System
Mon May 12, 1:57 PM
Matthieu Dreher
Terminal
File Edit View Terminal Help
FilterIt gforcegmx/filter IN: in:26988 order:6780 OUT: out:1900
PreSignal gforcegmx/presignal IN: in:1899 OUT: out:1899
GreedySynchronizor gforcegmx/sync IN: stamps:26988 endIt:1900 OUT: order:6780 :0
(8.3 it/s) Regulator gmx/0 IN: beginIt:1900 simulationforce:1899(1) checkpointrequest OUT: endIt:1899 out:1898 outsimulationforce:1873 outCheckPoint:0 outEnergy:1897
(8.3 it/s) Regulator gmx/1 IN: beginIt simulationforce:1898(2) checkpointrequest OUT: endIt:1899 out:1898 outsimulationforce:1872 outCheckPoint:0 outEnergy:1897
PreSignal pCurrentForces IN: in:26989 OUT: out:26989
PreSignal presignalAtomSelector IN: in:26991 OUT: out:26991
PreSignal presignalForceGenerator IN: in:26991 OUT: out:26991
PreSignal psRoutingForce IN: in:26989 OUT: out:26989
PreSignal psSelection1 IN: in:26989 OUT: out:26989
PreSignal psSelection2 IN: in:26989 OUT: out:26989
PreSignal psSelection3 IN: in:26989 OUT: out:26989
PreSignal psTargetDirection IN: in:26991 OUT: out:26991
PreSignal psTargetingState IN: in:1861 OUT: out:1861
MergeSegmented treeGMX/TreeMerge0/node0 IN: in0:1898 in1:1898 OUT: out:
graberstats
Total Energy
Total Energy
-1.08e+06
-1.085e+06
-1.09e+06
-1.095e+06
-1.1e+06
-1.105e+06
-1.11e+06
-1.115e+06
-1.12e+06
-1.125e+06
0
500
1,000
1,500
2,000
Energy
ExaViz 102.692 fps
<=>
VALID
UP
Atom Selection
ExaViz GUI
splay Force Gromacs Phantom Joypad Representation Filter 3D Target
Targeting informations
Number of targets 8
Current active target 5
Current selection position (58.7748 , 39.2362 , 52.1124)
Current target position (57.5833 , 40.7698 , 51.5333)
Distance to current target 2.02658
Current speed (A/s) 0
Time until current target (sec) 0
Terminal
Terminal
Phantom
joypad
ExaViz
ExaViz GUI
graberst...

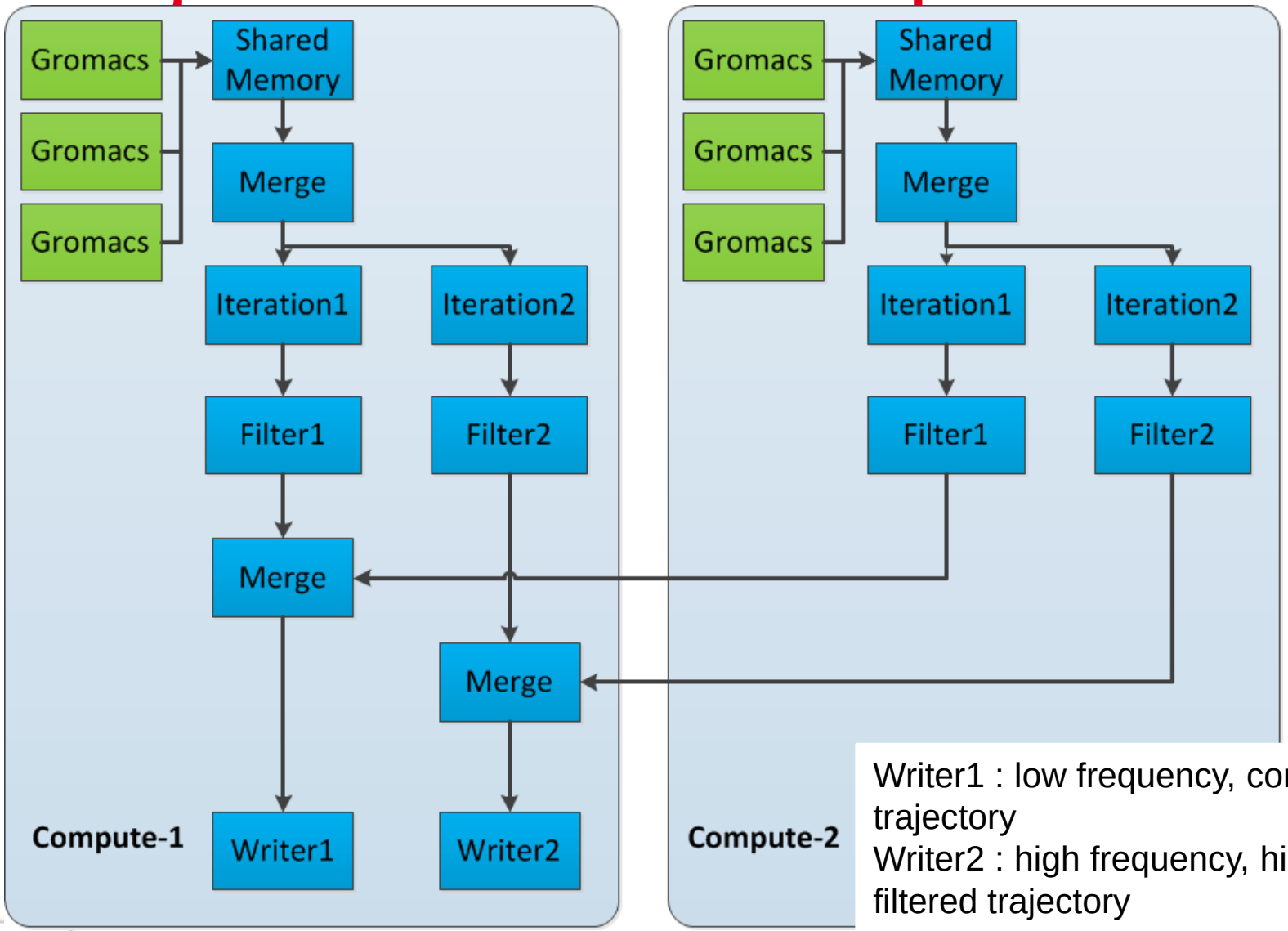
Writing scenario : local write



Writing scenario : global write



Advance Writing scenario : multiple trajectories with various frequencies



Writer1 : low frequency, complete trajectory
Writer2 : high frequency, highly filtered trajectory

3

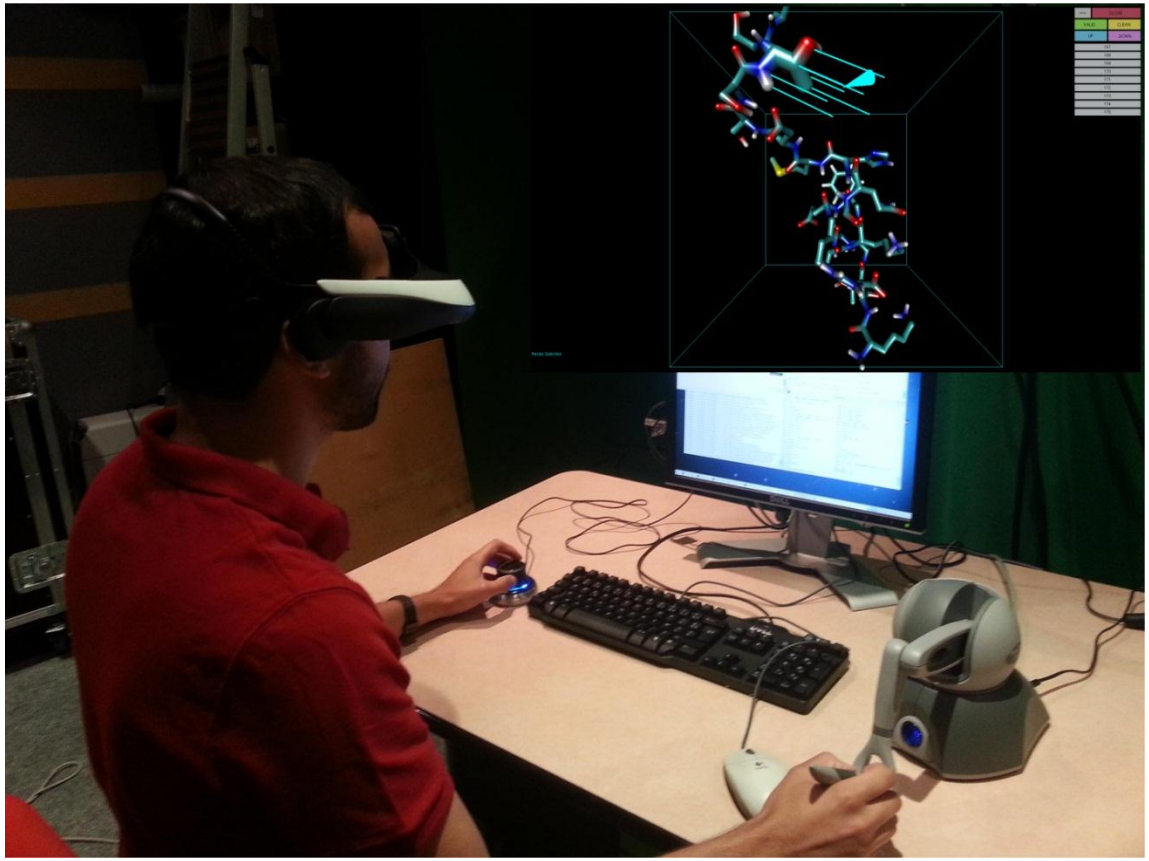
Framework usage on Ciment plateform

Development cycle stage 1 : Desktop

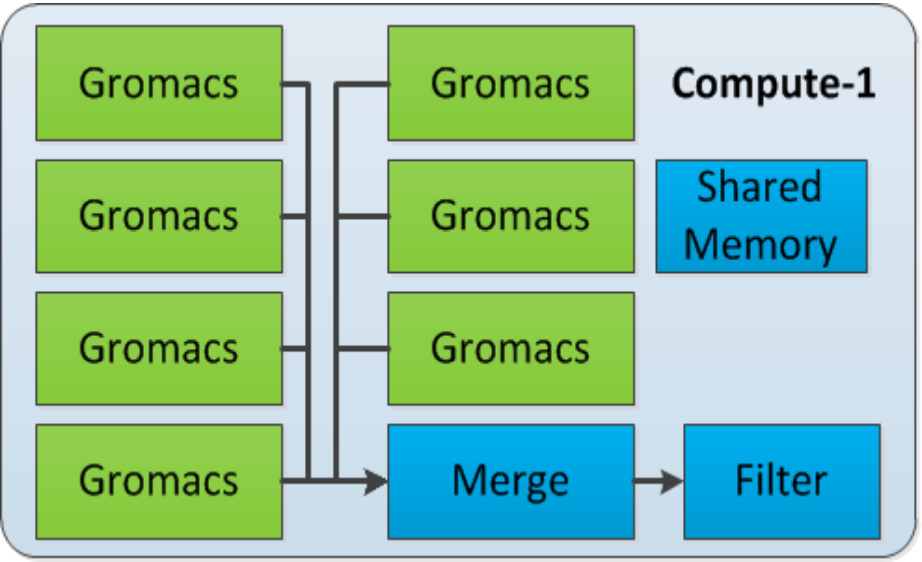
- Functionnalities development
- Test with low level parallelism (1-4 cores)
- Debugging
- Deployment on localhost

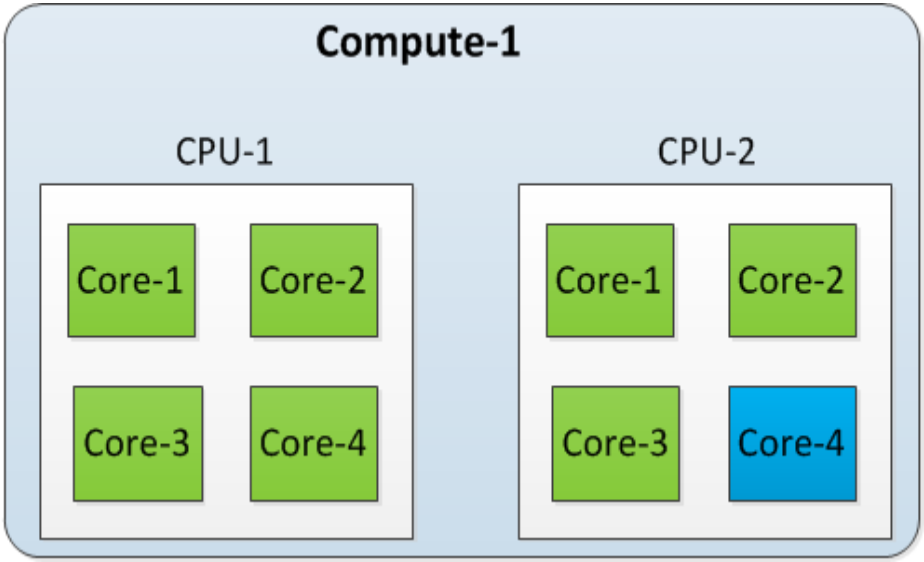
Development cycle stage 2 : Grid5000

- Experimental environment
- OAR + Kadeploy
- Root access
- Connection with a visualization node (Digitalis)
- Reservation mode with a user chart
- From 1 to 512 cores (64 nodes)
- Goal : **Detect scalability issues**



Exemple scalability issue : Process Mapping to Cores



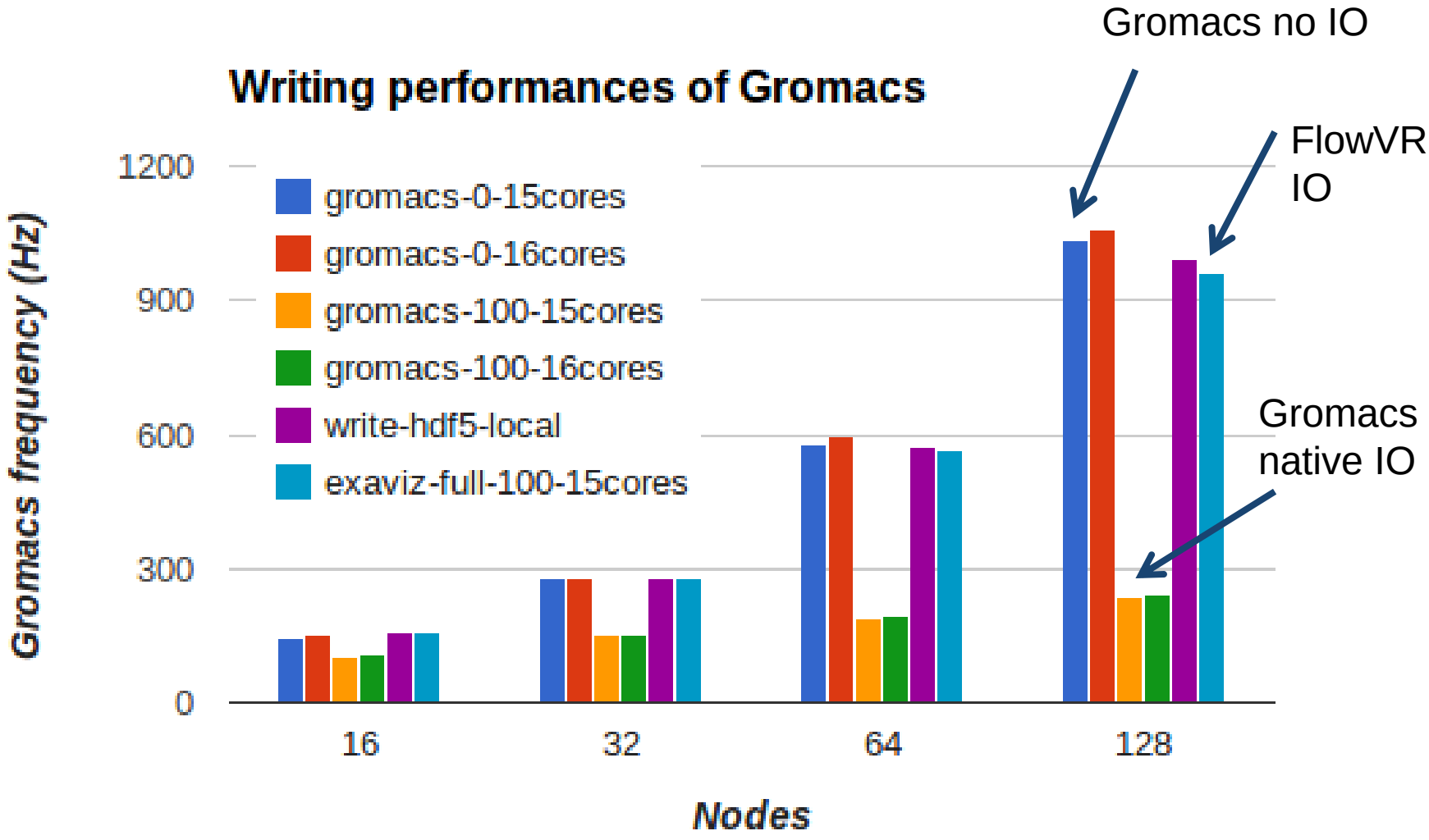


- Dedicated core for the extra operations (merge, filter)
- Force fixed mapping of processes to cores (mpirun –rankfile, taskset)
- Up to 50% performance drop if scheduler controlled mapping!

Development cycle stage 3 : Froggy

- Production environment
- No root access
- Visualization node
- Hardware close to Edel cluster (Infiniband, 2 sockets, Intel procs)
- From 1 to 2048+ cores
- Fair sharing policy
- Specific debugging for high level of parallelism
- Directly benefits from optimizations done at stage Grid5000!

Writing performances (higher is better)



Launching a FlowVR application on Froggy

Reservation:

- Oarsub –r « date » -l nodes=128,walltime=4 –k –e ~/froggy_key
- -r must be used with moderation (Interactive session)
- Don’t always get all the requested nodes
- Application tested up to 128 nodes (2048 cores)

Launching the application :

- Oarsub –C myJobID
- Configure ssh :

```
Host frog*
  User oar
   IdentityFile /home/mdreher/froggy_key
   Port 6667
```

- Generate the hostfiles based on $OAR_NODEFILE
- Mpirun –np 128 –machinefile myDaemonHosts flowvrd –top –s 1G
- Python fvnanopython –gmx mySimulationHosts
- Flowvr fvnanopython –l -s

4

Conclusion

Conclusion

Conclusion:

- In Situ is a promising solution to help solving the data management problem (size, transfert, etc…)
- Can save a lot of ressources and time by reintroducing the user into the simulation
- The Froggy and Grid5000 plateforms gives a suitable environment to develop such applications
- Application tested up to 128 nodes (2048 cores)
- Vitamins project : http://vitamins.gforge.inria.fr/doku.php

On going work :

- Experimentations on Hyperthreading and Infiniband usage
- Terminaision criteria based on analysis

Conclusion

Conclusion:

- In Situ is a promising solution to help solving the data management problem (size, transfert, etc…)
- Can save a lot of ressources and time by reintroducing the user into the simulation
- The Froggy and Grid5000 plateforms gives a suitable environment to develop such applications
- Application tested up to 128 nodes (2048 cores)
- Vitamins project : http://vitamins.gforge.inria.fr/doku.php

On going work :

- Experimentations on Hyperthreading and Infiniband usage
- Terminaision criteria based on analysis