From Live Multi-Camera 3D Modeling to In Situ Processing for HPC

Bruno Raffin
DataMove

bruno.raffin@inria.fr

November 2016

What this talk is about

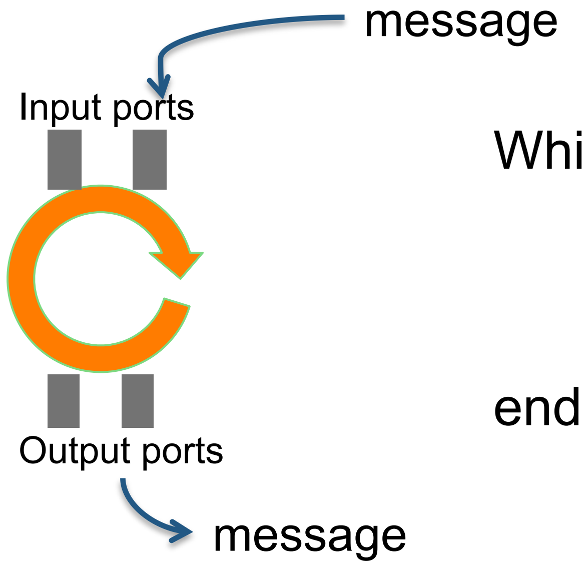
Data flow programming for data intensive high performance applications

Tool: FlowVR

Evolved from applications to interactive applications (virtual reality, 3D modeling) to computational steering and in situ processing for large scale numerical simulations

FlowVR : Data Flow Programming

Base object: components




```
While ( wait(inputs) )
        get()
        compute()
        put()
end
```

Get/put messages from/to ports: no explicit origin/destination

A simple API to limit code intrusion
Direct access to memory: no hidden copies
Adapted to develop domain specific layers

http://flowvr.sf.net

FlowVR : Data Flow Assembly

Python scripting:

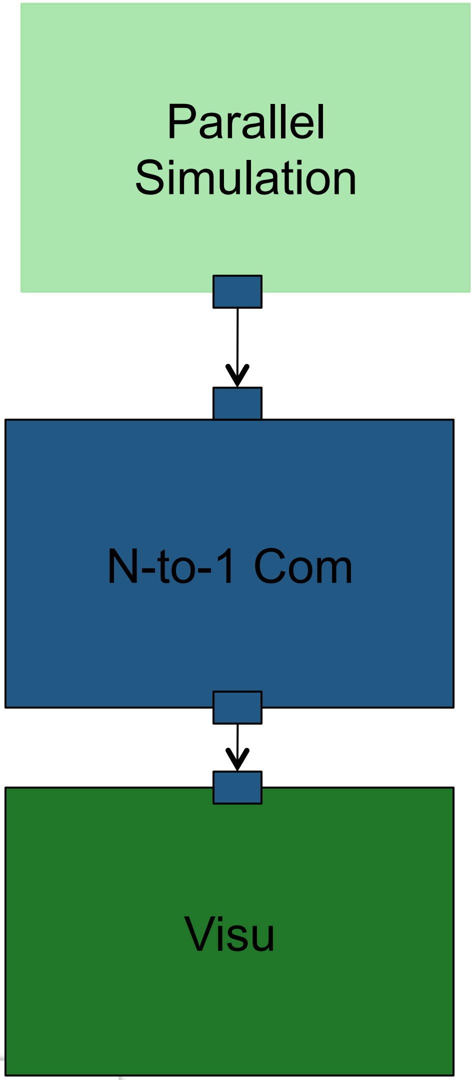
1. Assemble components (data flow graph)
2. Instantiate parameters: component mapping,…

Script execution

Launching commands:
mpirun app1….
mpirun app2….

Configuration commands for daemons
(message routing,….)

FlowVR : Example



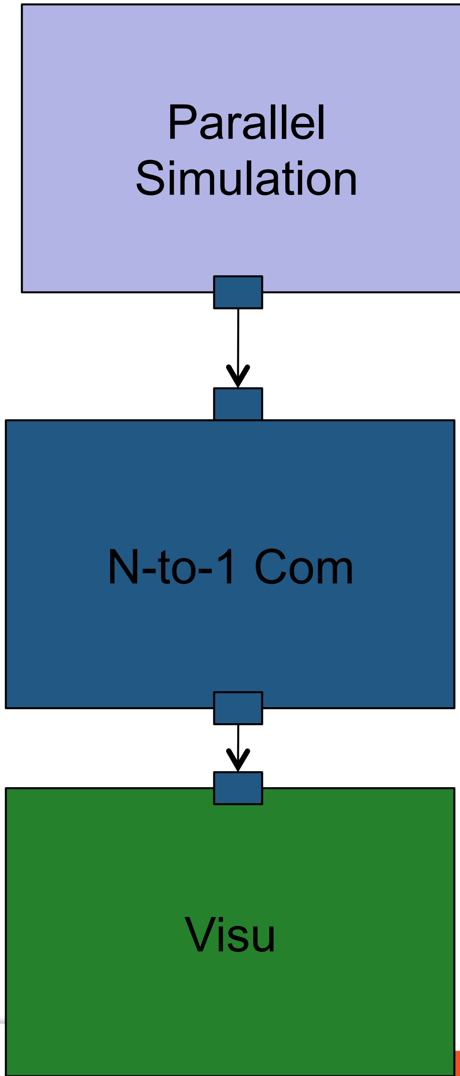
FlowVR : Example

N-to-1 Communication Pattern
(tree with arity k - unkown)
(N unknown – get value from incoming component)

Merge filter
(K inputs –unkown)

bruno.raffin@inria.fr

FlowVR : Example

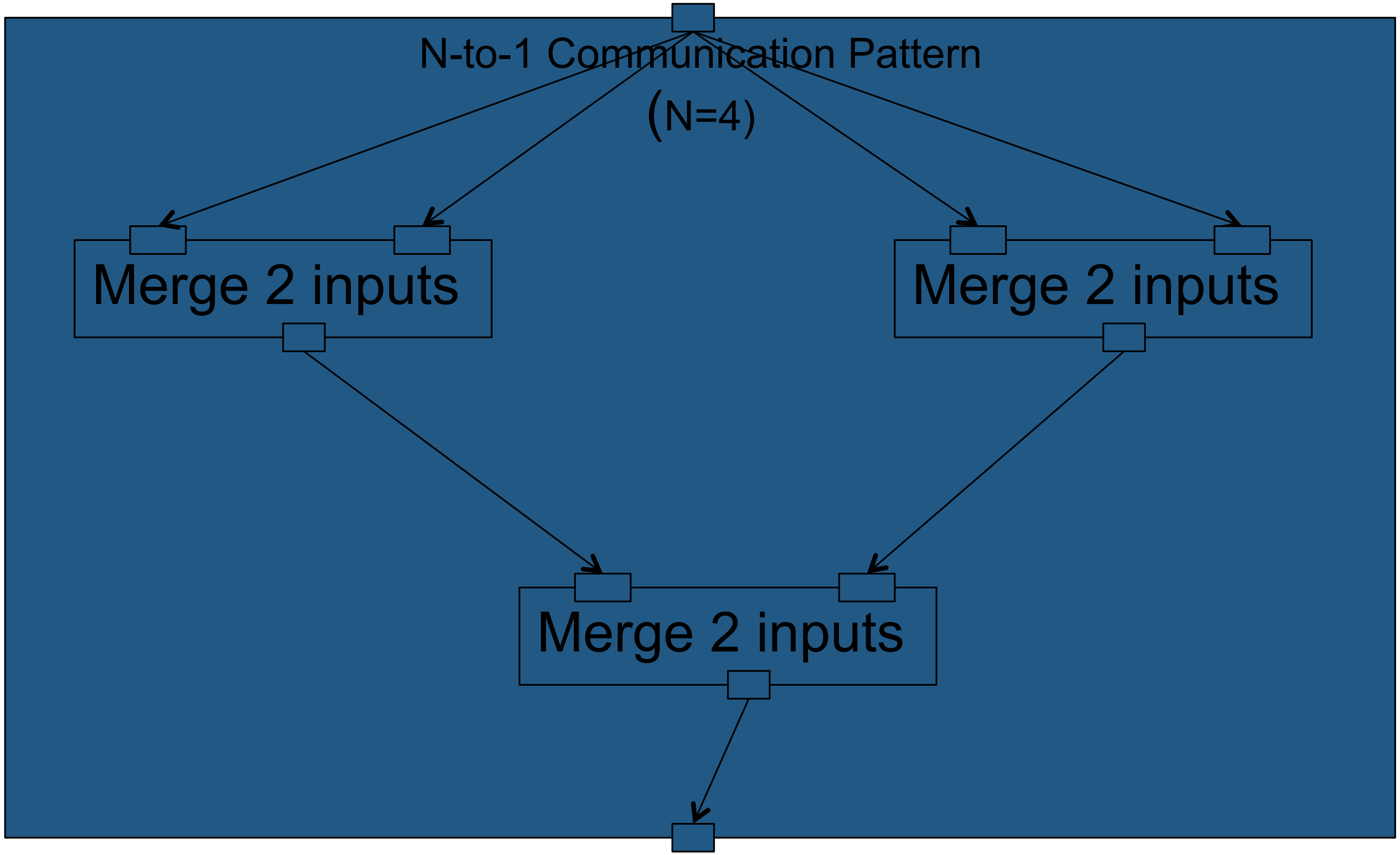


N=4

K=2

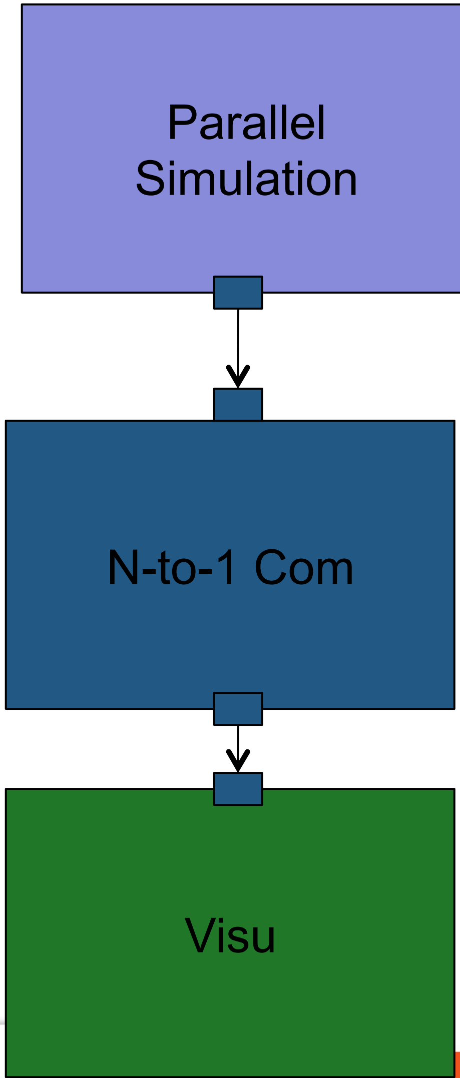
bruno.raffin@inria.fr

FlowVR : Example



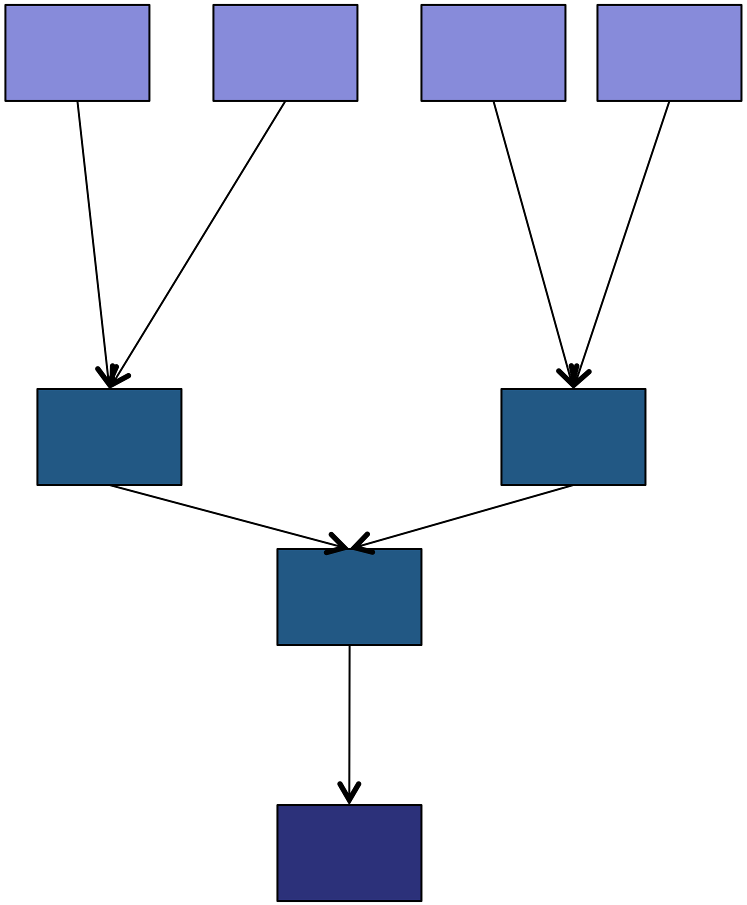
bruno.raffin@inria.fr

FlowVR : Example



N=4

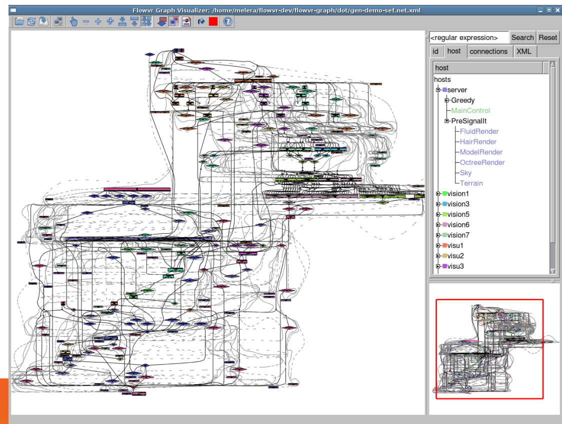
K=2



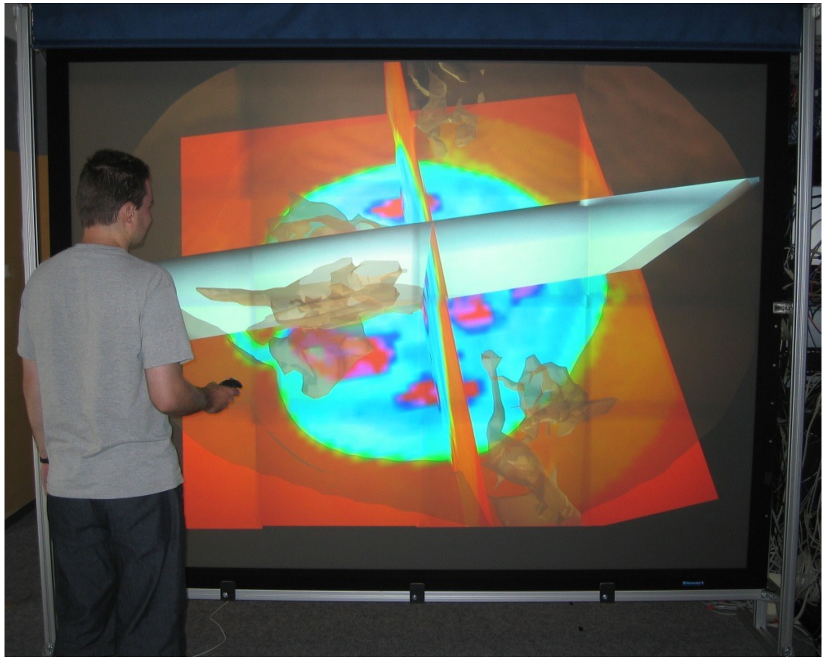
FlowVR Runtime

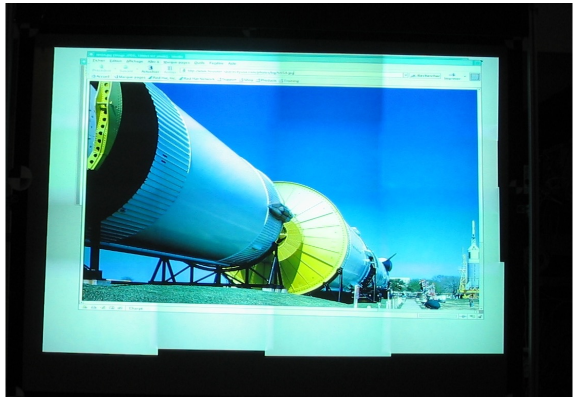
Run one daemon per node

- Components (applications) attach to the daemon when starting
- Daemons in charge of routing messages and triggering component executions:
  - Pointer exchange inside a node (shared memory)
  - MPI or TCP between nodes



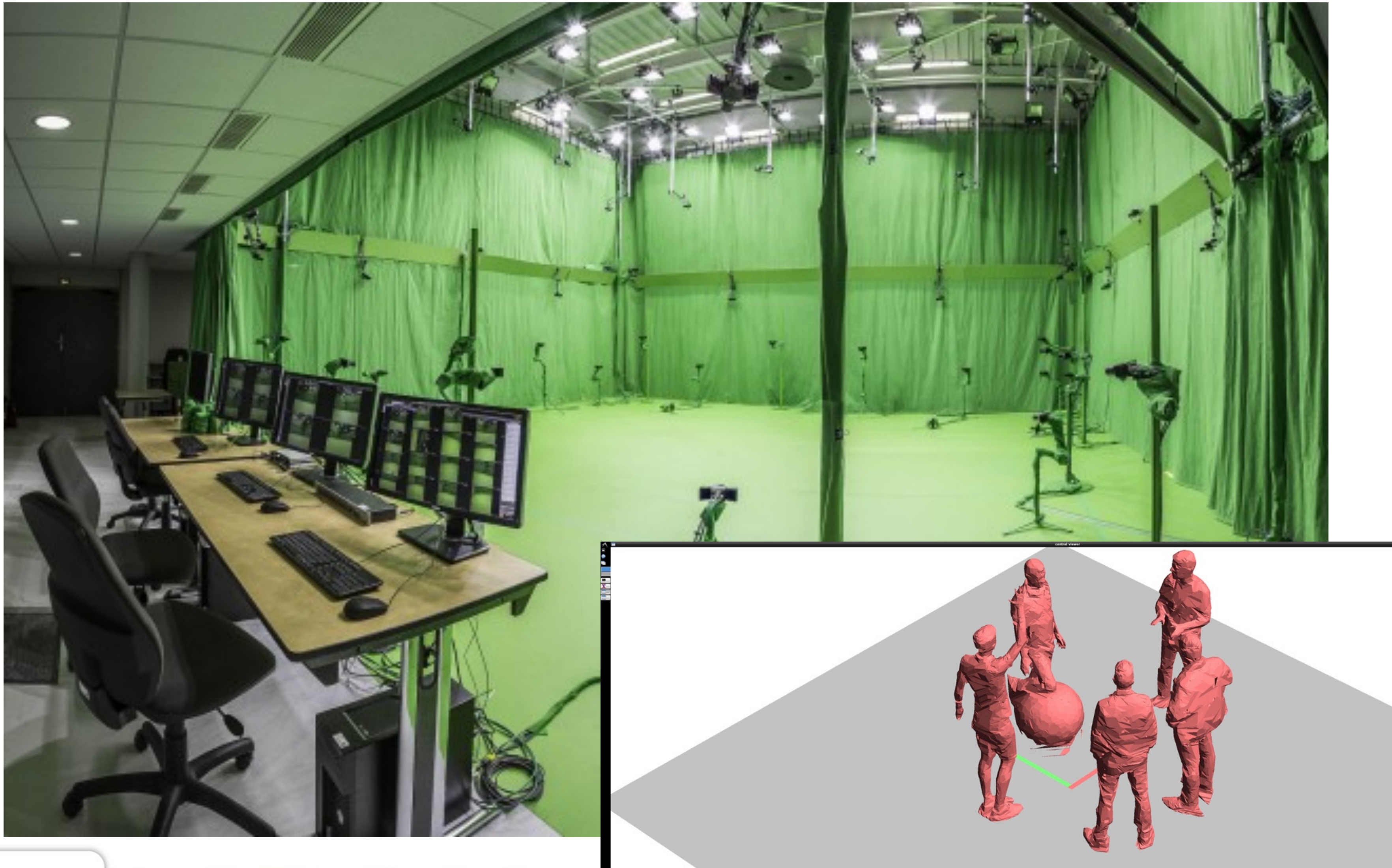
Interactive Applications





Interactive Applications

Equipex Kinovis: 70 cameras



Itermediate Summary

- Data Flow Programming well adapted to this type of interactive data intensive applications
- Avoid moving data, process them as closely as possible to their source
- No intermediate storage to disk, process data as soon as available (some buffering possible)

Actually an approach that is today relevant for more traditional large scale numerical simulations !

The Data Challenge

More compute capabilities -> larger simulations -> more data

Usability Challenge:

- How to extract meaningful information from this huge amount of data in a reasonable time
- Analysis tools have not been considered as first class citizen so far. They did not receive the same attention as simulation codes. Today analysis codes are either:
  - In the simulation codes
  - Scripts (with limited parallelism)
  - Rely on on scientific visualization tools like Paraview/VTK or Visit (reasonable parallelism support)

Performance Challenge:

- Moving data becomes the bottleneck for simulation as well as data analytics
- Compute capabilities increase faster than data transfer ones
- Data movements and storage consume 50%-70% of total energy (ScidacReview 1001)

A Data Challenge Already Present

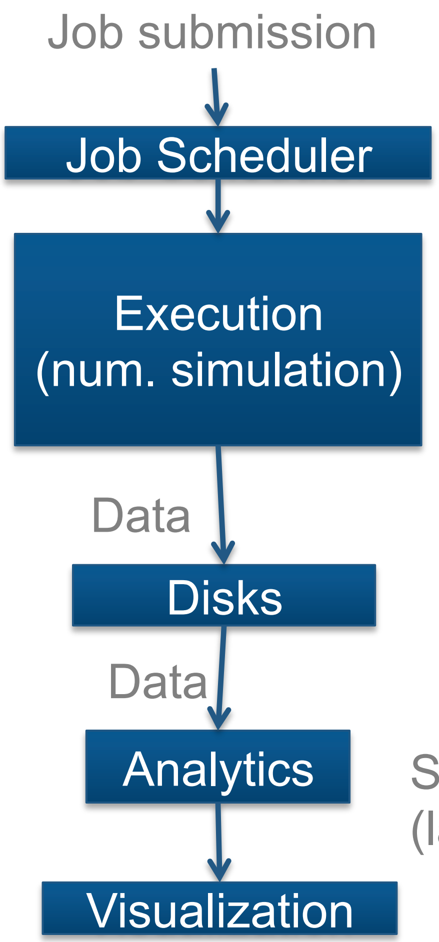
Scientists already spend a significant part of their efforts in the data analysis

A simple but classical strategy to limit the impact of the data challenge:
Reduce output frequency

Need for more advanced strategies to better manage the available I/O and storage budget

In Situ Processing!!

Traditional Workflow



Not sustainable at Exascale !

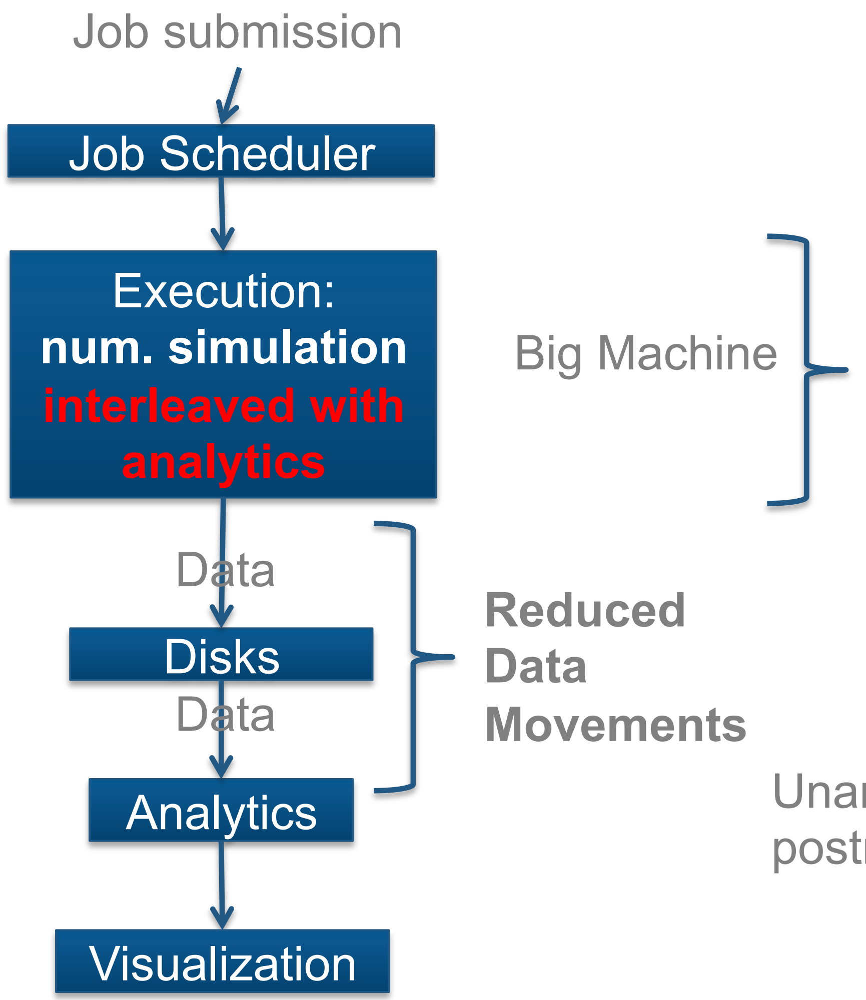
Big Machine

Simulation codes may include some analysis

Small Machine (laptop)

Limited support for parallelization

WorkFlow with In situ Analytics



In situ analytics:

Pros:

- Data reduction
- Large scale parallel analytics
- On-line monitoring

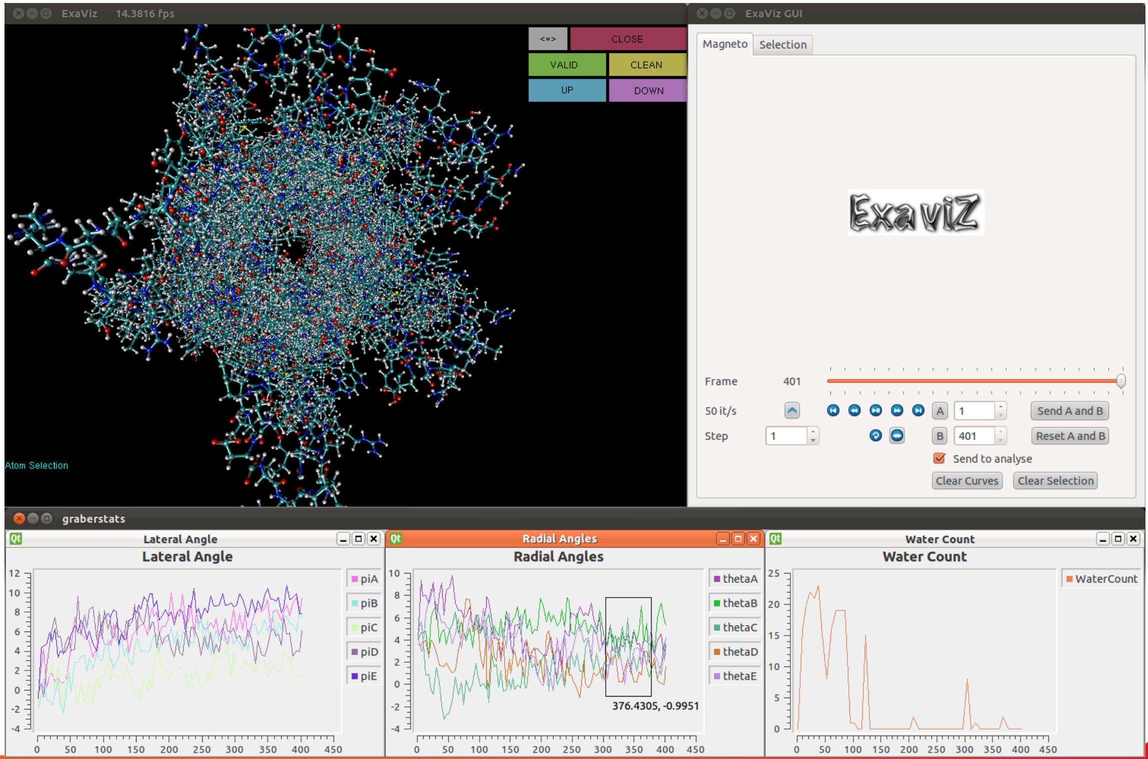
Cons:

- Need to anticipate analytics
- Programming model ?

Unanticipated analysis still need to be performed postmortem

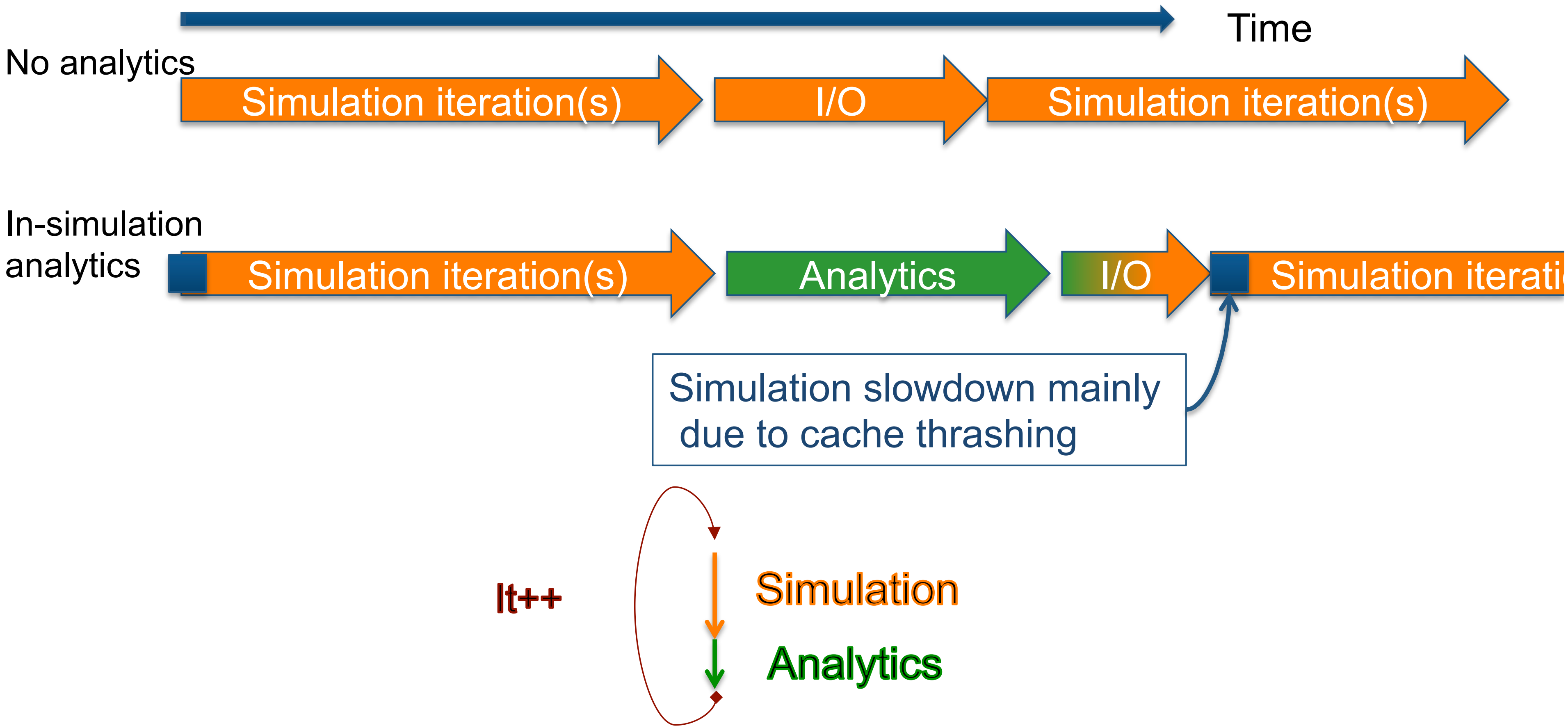
In Situ Processing: What for ?

Data compression (Isabela [Lehmann & al. LDAV'14])
Indexing (FastBit, Dirac [Lakshminarasimhan & al. HPDC'13])
Analytics (1D, 2D, 3D descriptors)

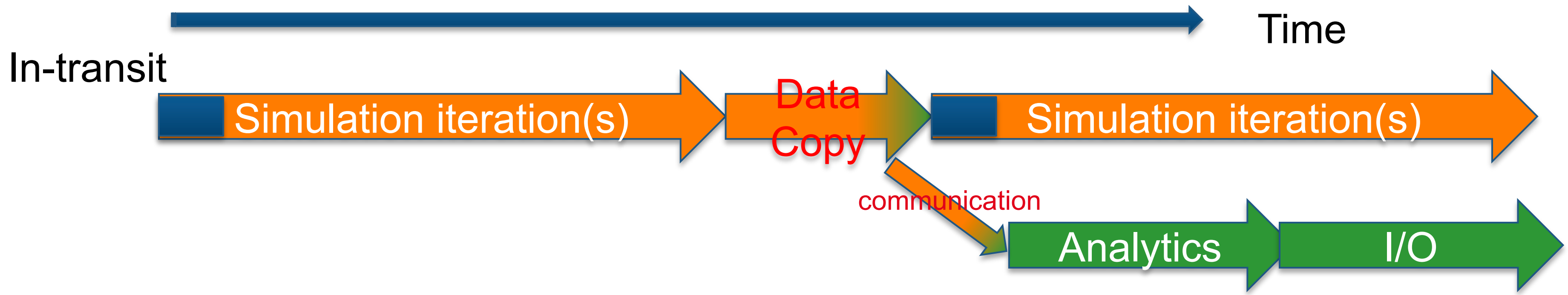


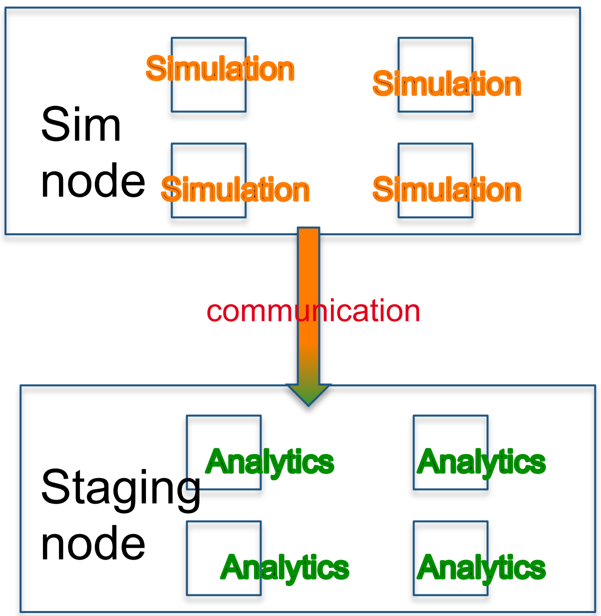
[Dreher & al.
Faraday Discussion'14]

In Simulation Processing



In Transit Processing

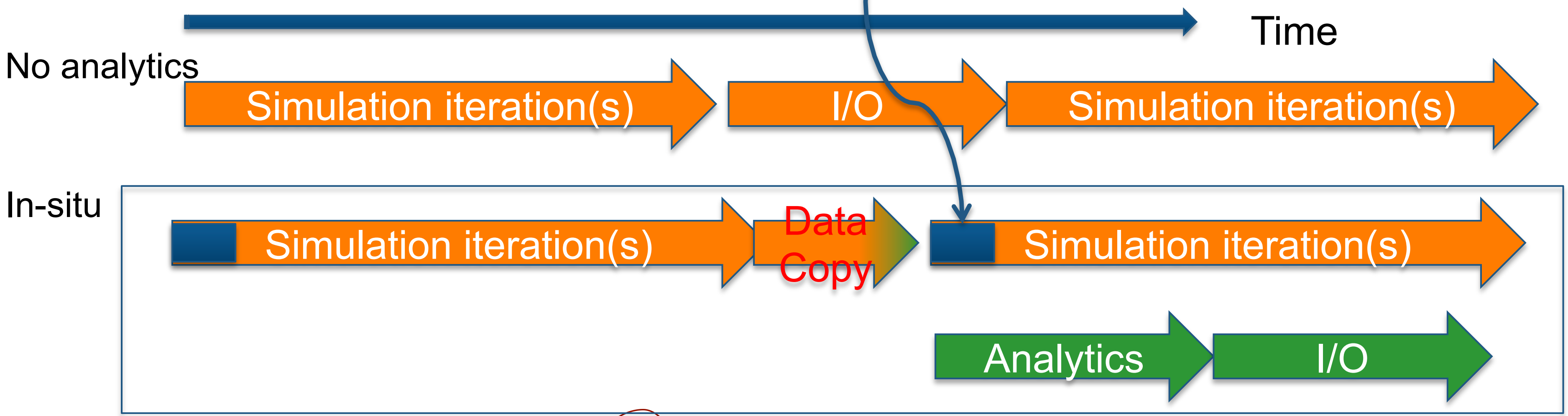


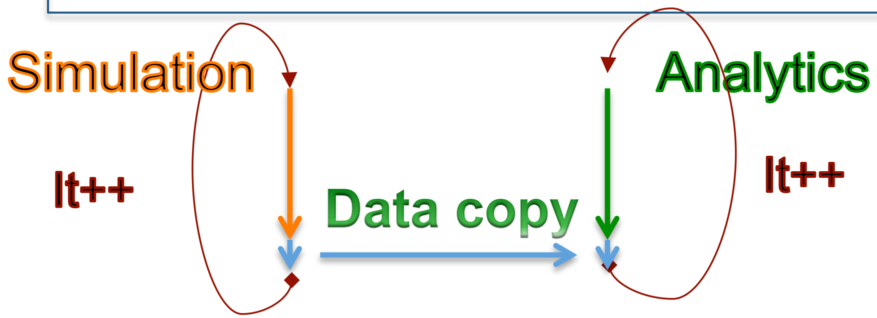


In-transit: simulation and analytics run on different nodes (staging nodes)

In Situ Processing

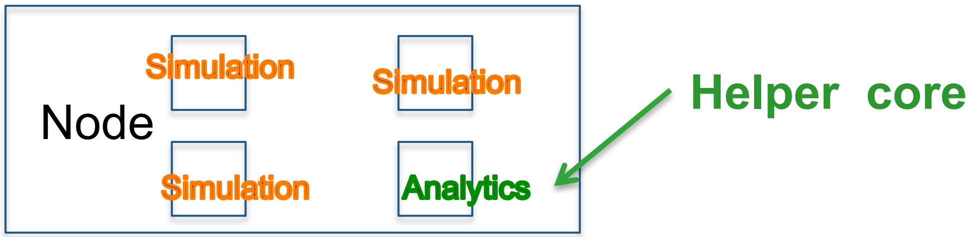
Simulation slowdown
due to concurrent use of some
resources with analytics and I/Os



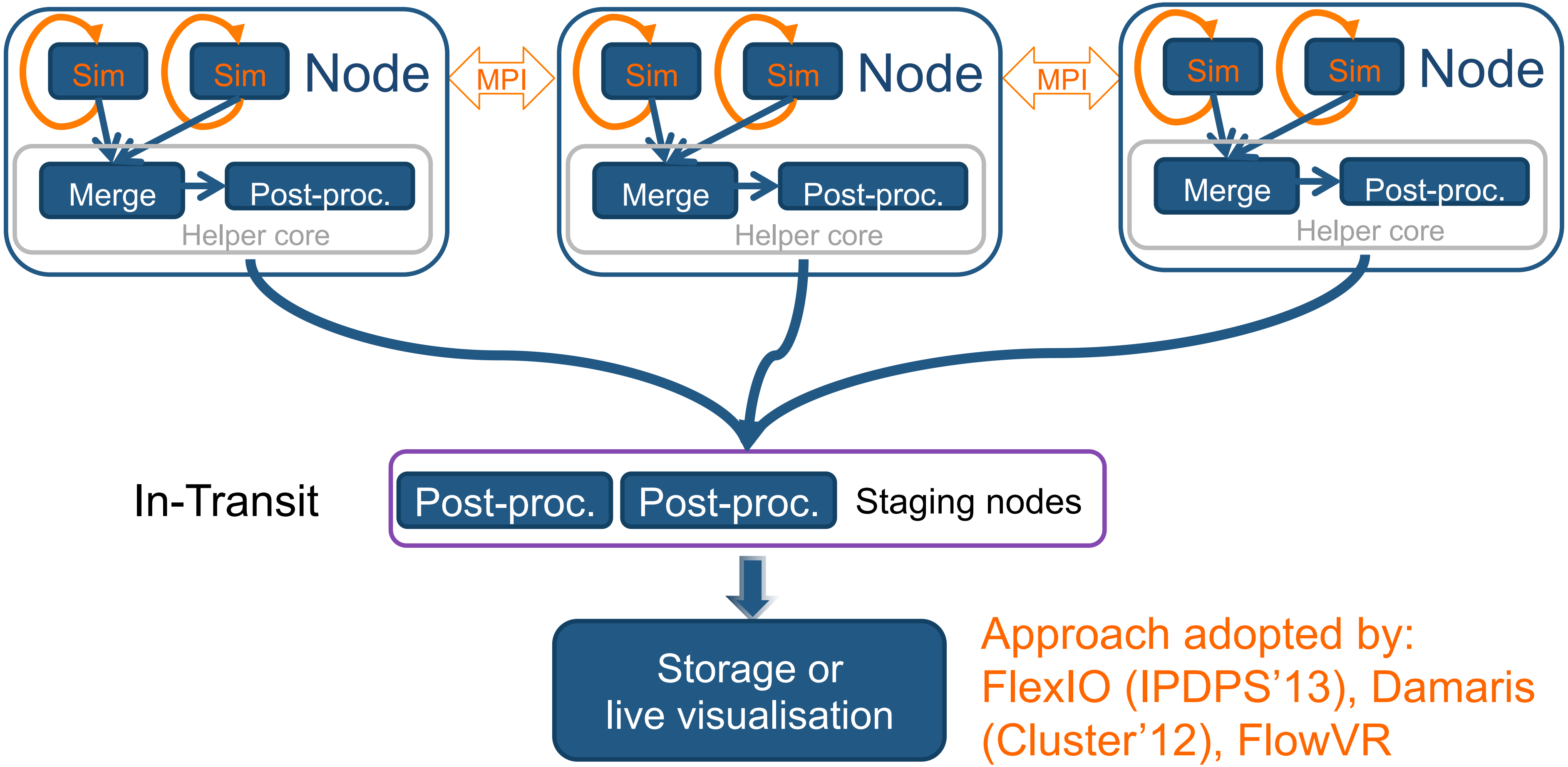


In situ:
simulation and analytics share the same nodes

Resource allocation strategies:
time sharing or space sharing
(dedicated helper core)

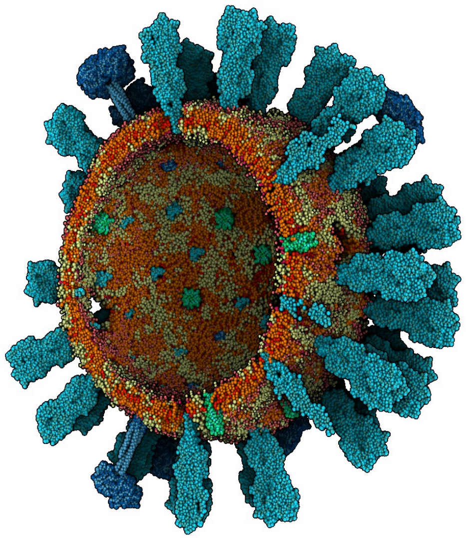


Asynchronous In Situ + In Transit Processing



Approach adopted by:
FlexIO (IPDPS'13), Damaris (Cluster'12), FlowVR (CCGrid'14)

Molecular Dynamics



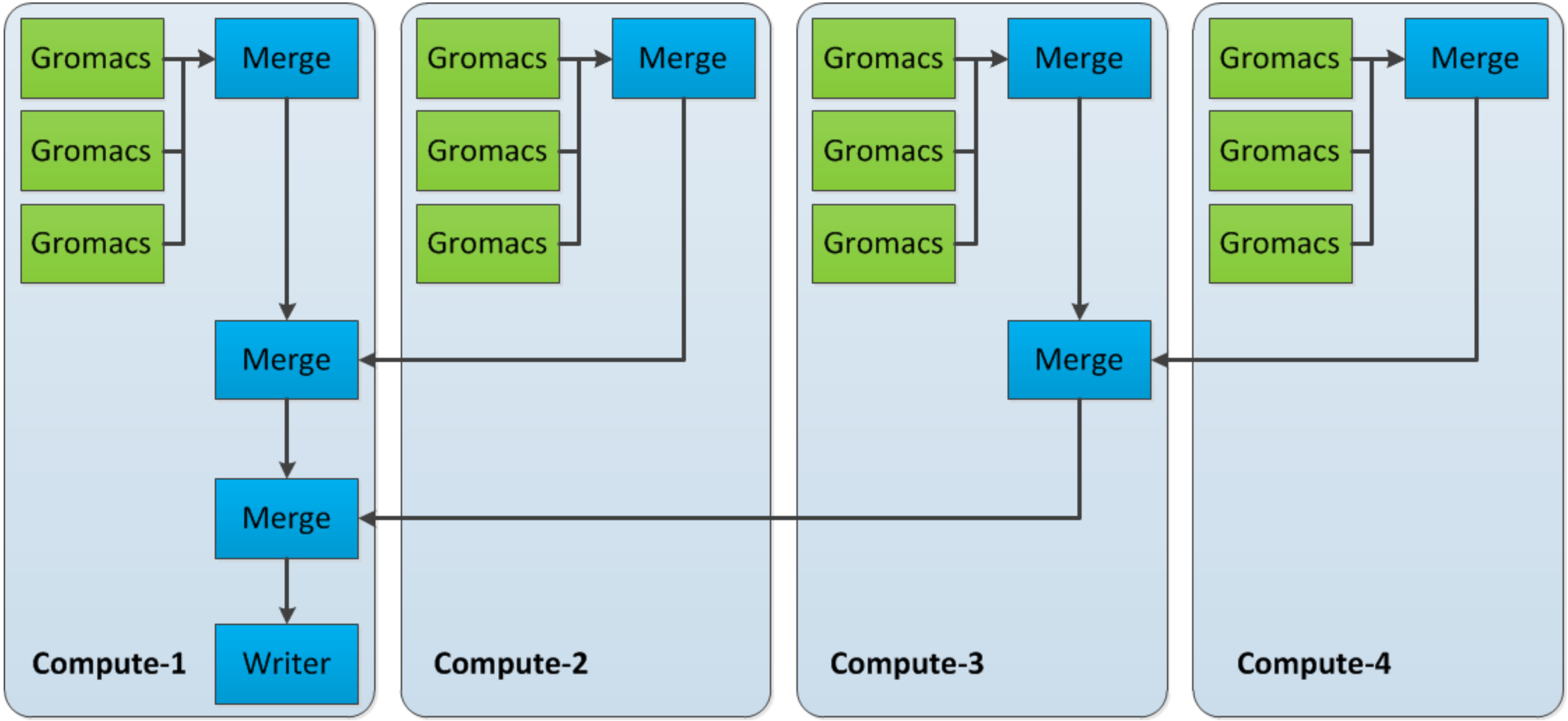
Collaboration with Marc Baaden (IBPC):

- Realistic molecular systems
- Realistic analytics scenarios

Gromacs:

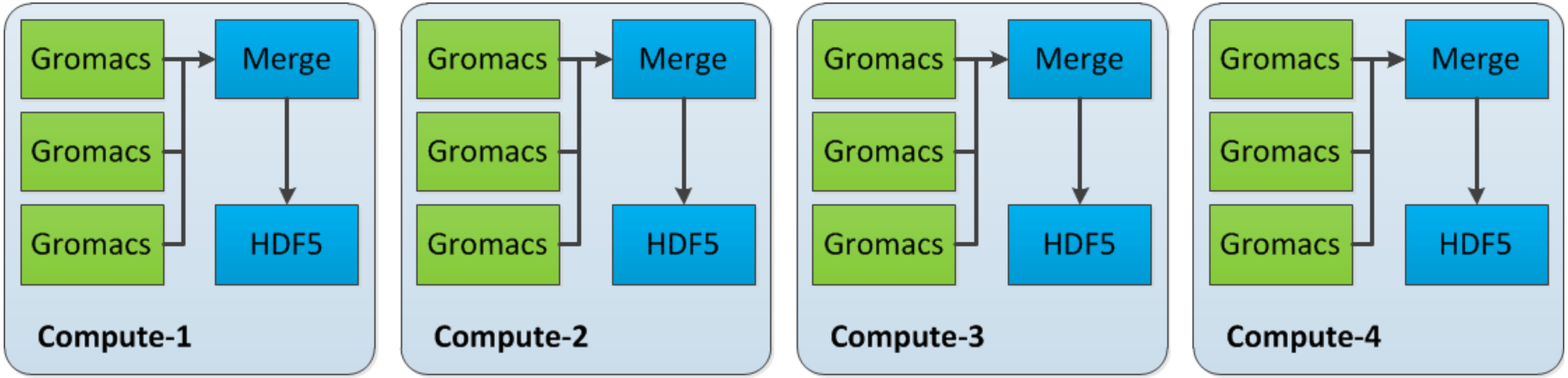
- Standard open source molecular dynamics simulation code
- Hybrid MPI+openMP+GPU parallelism, with integrated dynamic load balancing strategies
- Heterogeneous: long range interactions handled through dedicated processes (PME nodes)
- Very high iteration frequency (500 Hz)
- Often a benchmark in exascale related publications

Exemple: Benefit of the In Situ Helper Core Strategy



Aggregate the results in situ on helper-core (1 per node) to the master node

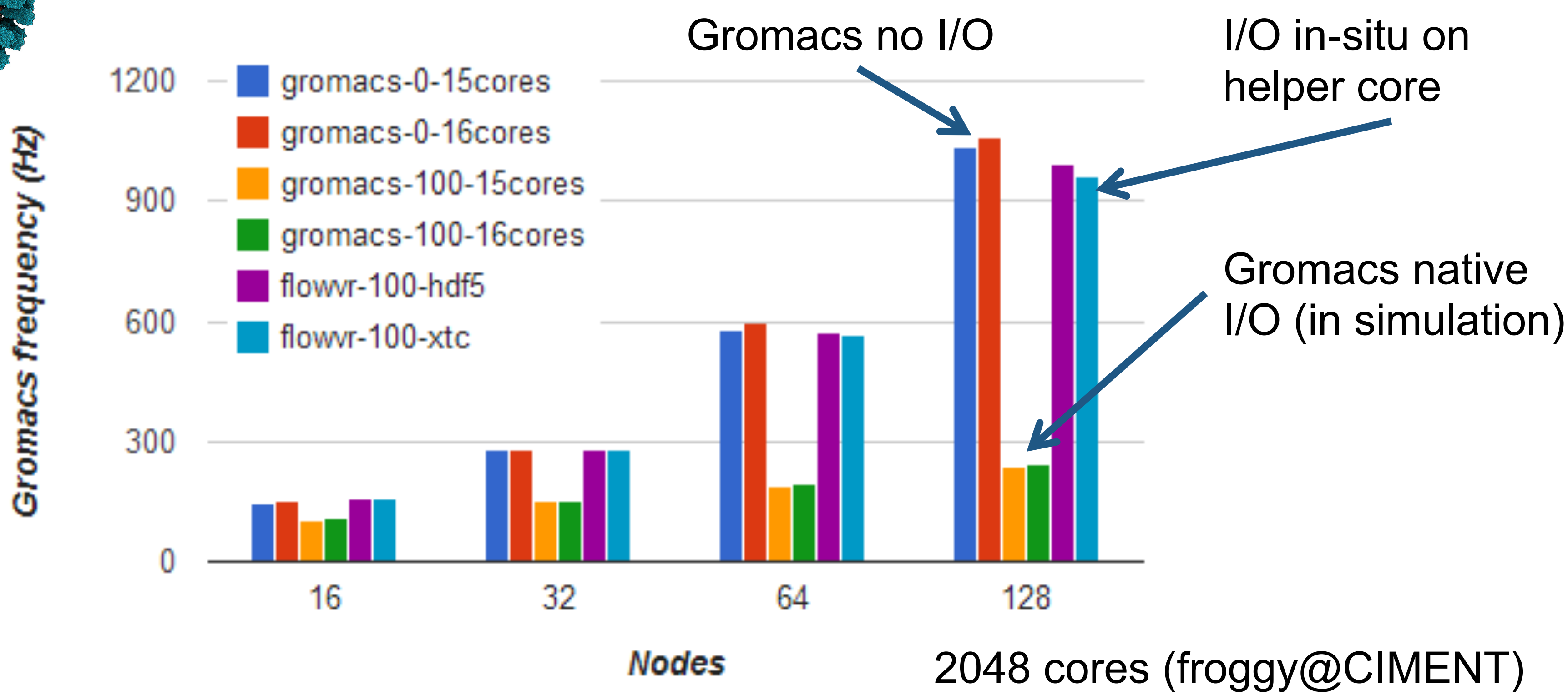
Exemple: Benefit of the In Situ Helper Core Strategy



Aggregate local results in situ on helper-core (1 per node) and write to disk

FlowVR proved well adapted to develop these strategies

In Sim vs. In Situ I/O [Dreher,CCGRID'14]

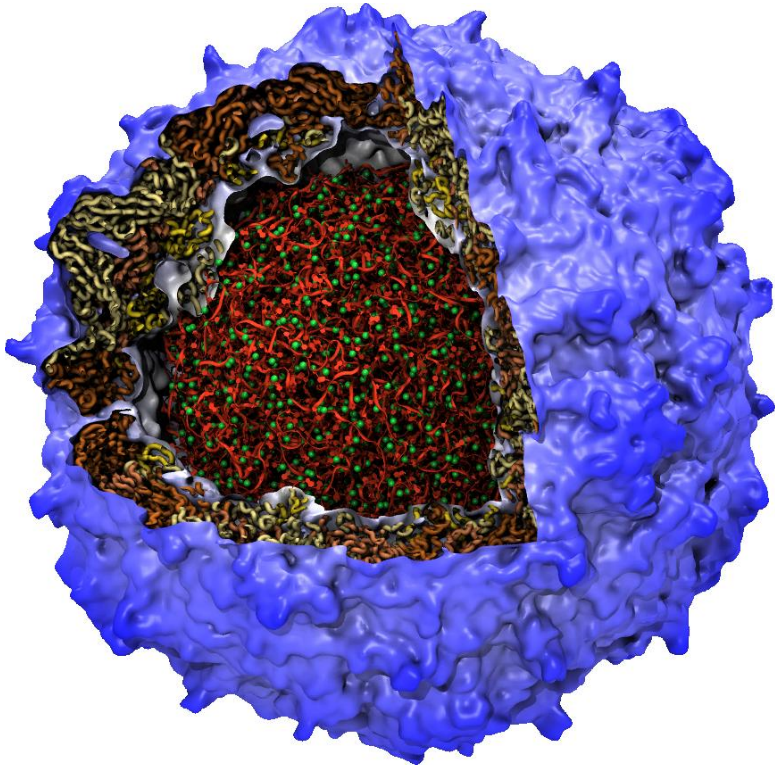


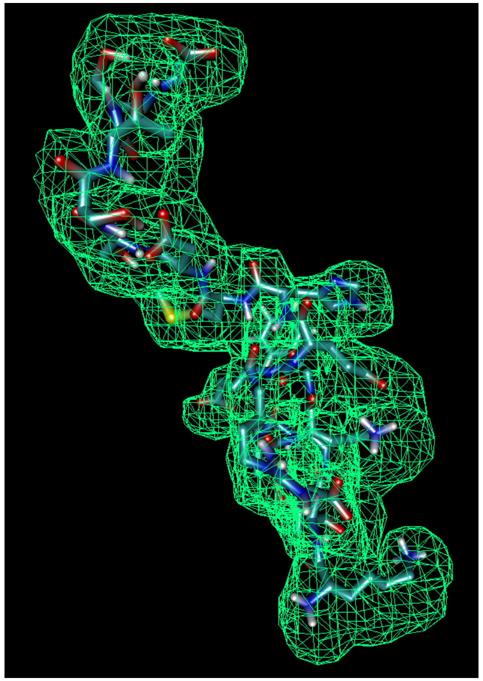
2048 cores (froggy@CIMENT)

Gromacs without I/O: 15 cores/node 3% slower than 16 cores/node

(- 6% if scalability would have been perfect)

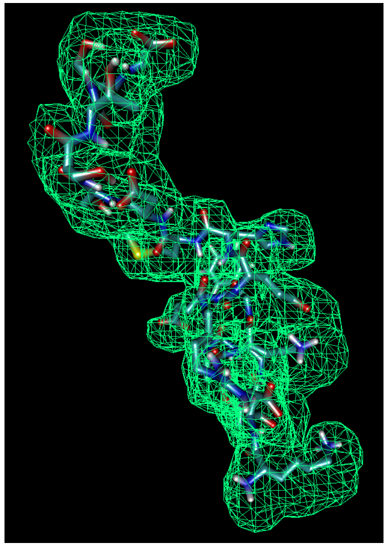
Parallel In Situ Quicksurf



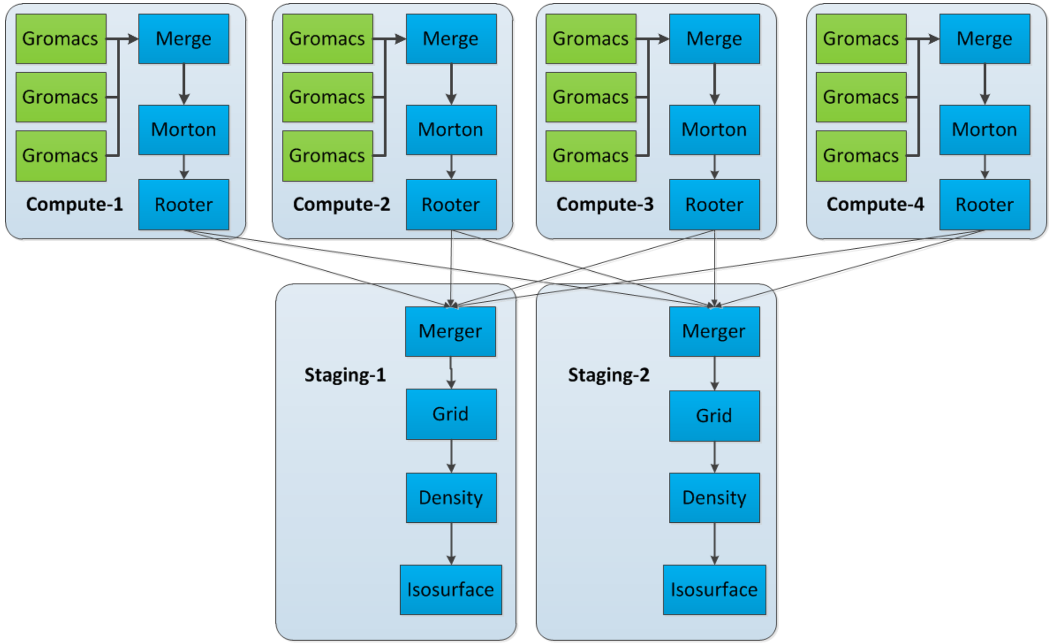


Classical way to visualize a molecule surface (isosurface based on atom density)

Parallel In Situ Isosurface Extraction [Dreher,CCGRID'14]

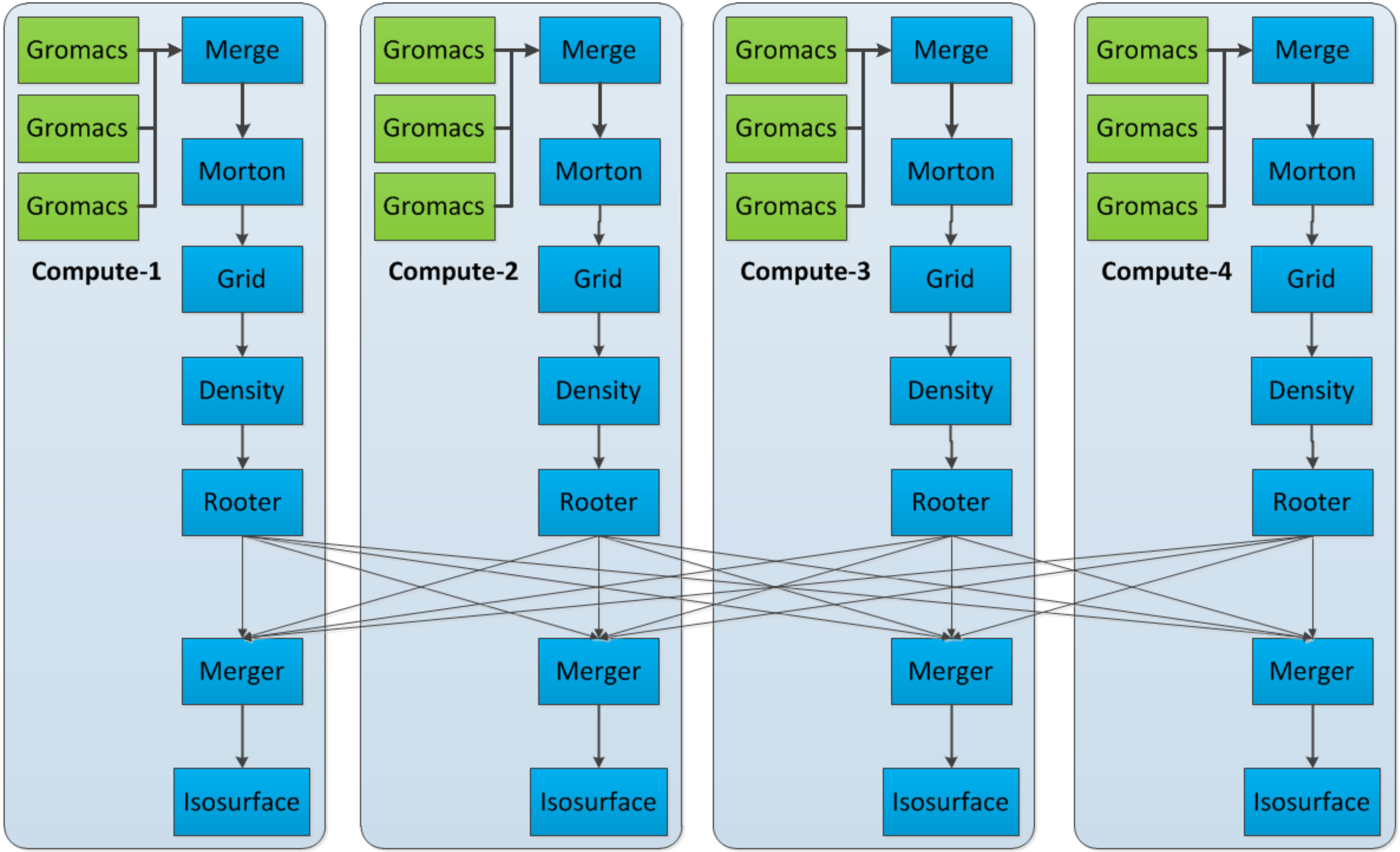


Compute a molecule surface based on atom density

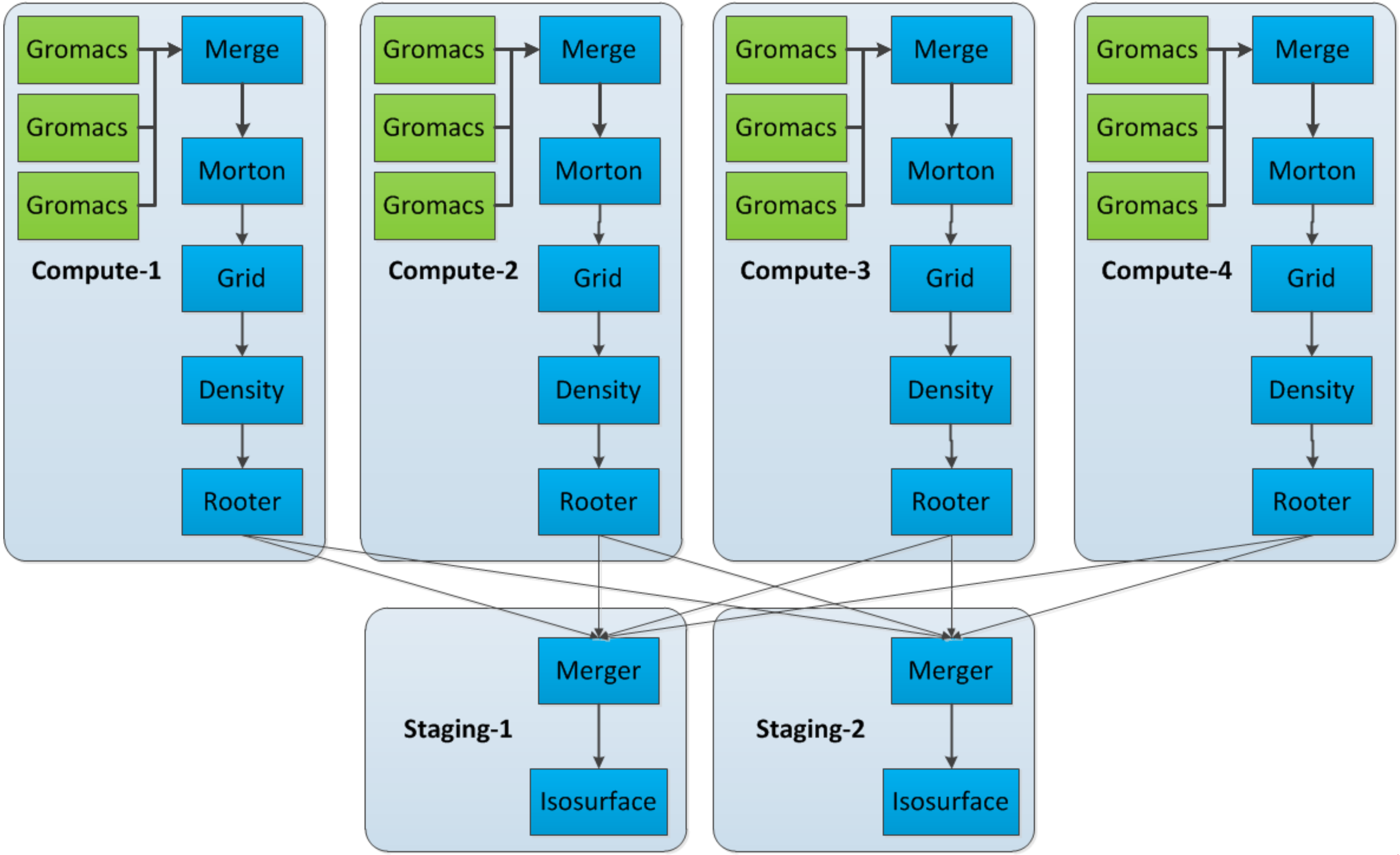


Tested different distributions of processing steps to in situ and in transit nodes.

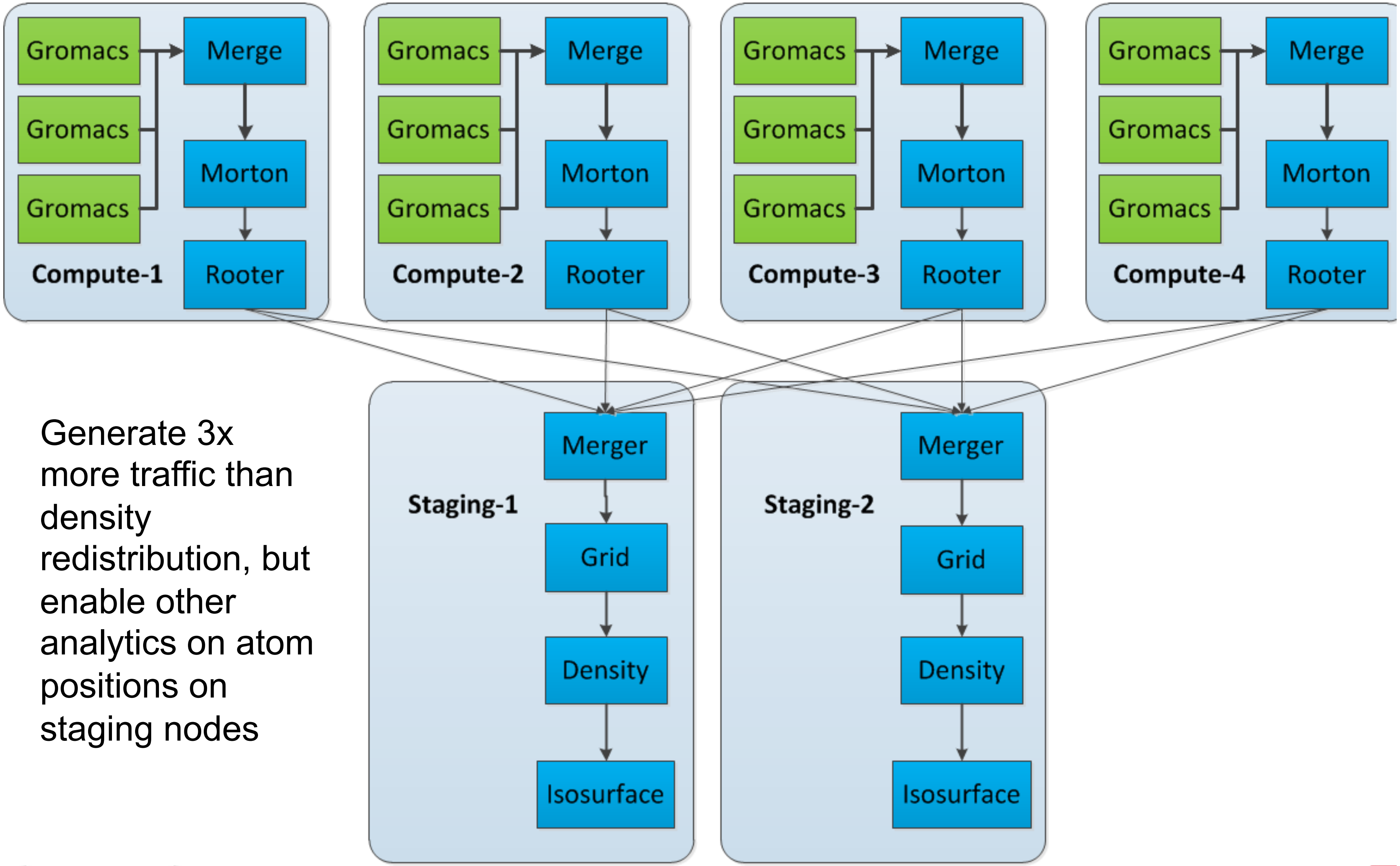
Densities Redistributed In Situ



Densities Redistributed In Transit

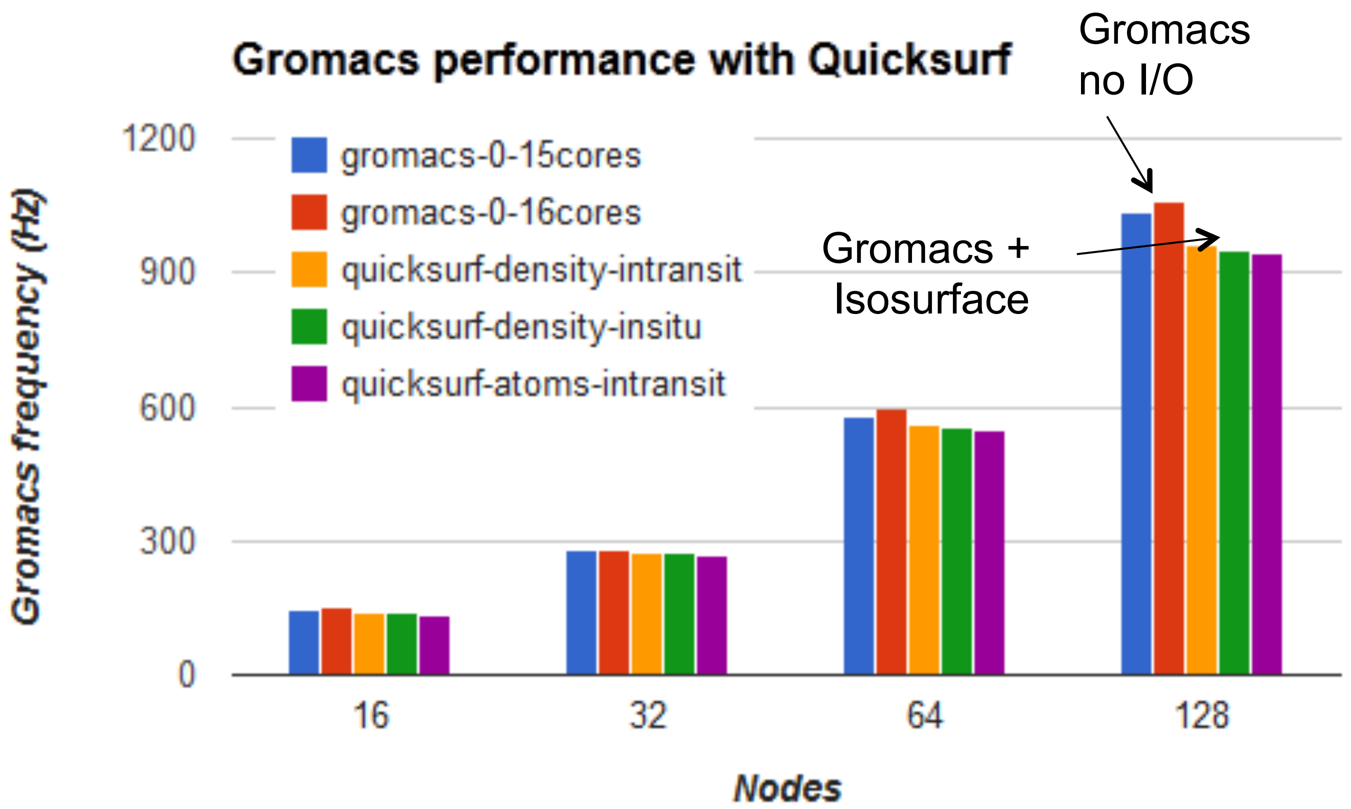


Atom Positions Redistributed In Transit



Generate 3x more traffic than density redistribution, but enable other analytics on atom positions on staging nodes

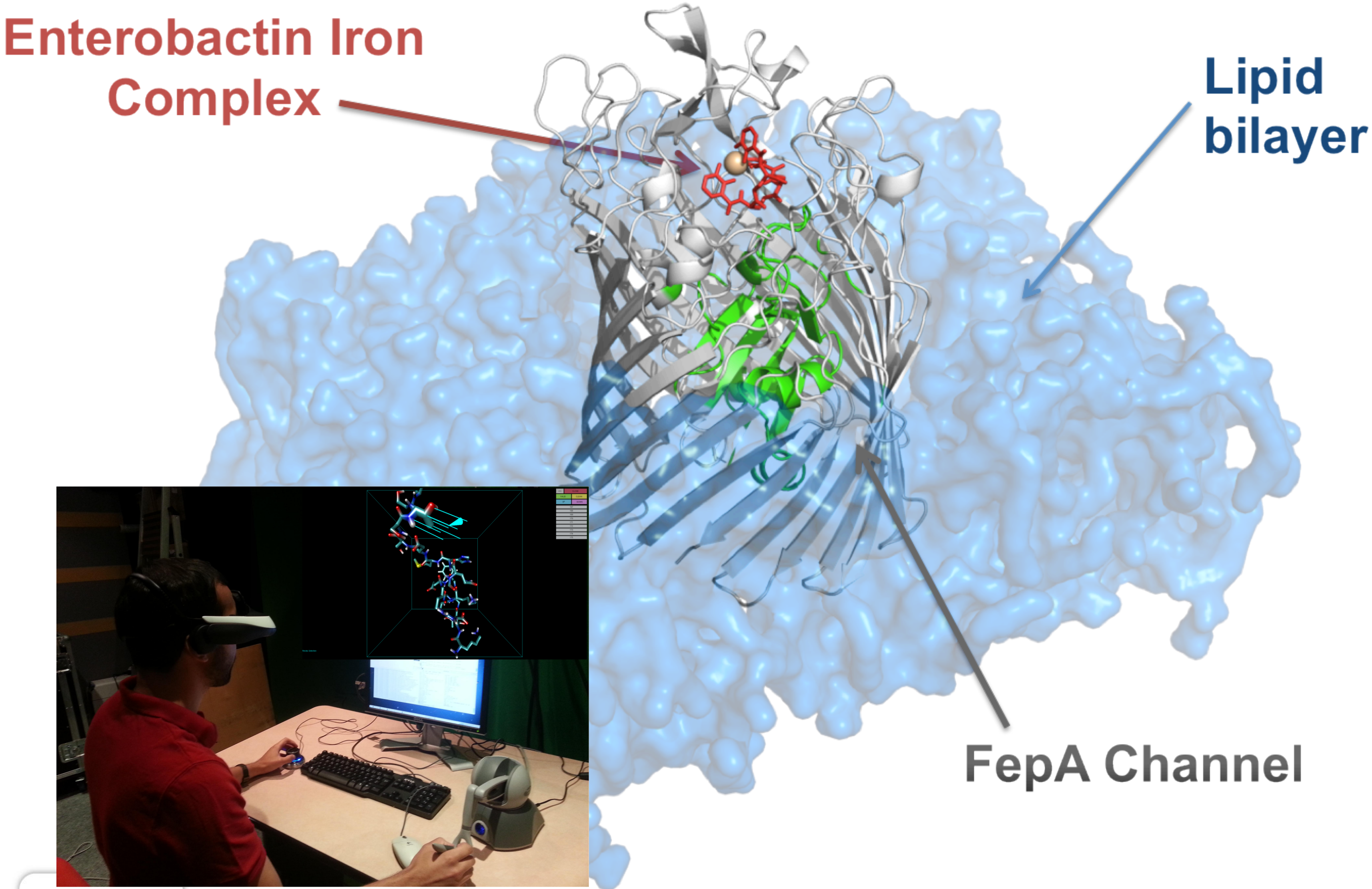
Performance [Dreher,CCGRID'14]



(froggy@CIMENT)

- In transit: 1 staging node every 64 compute nodes
- Density-intransit: costs 7% comp. to gromacs 15 cores
- Density-insitu costs 8% but use 1.5% less nodes than density-intransit
- Atoms-intransit costs 8.6% but enables other in transit analytics (3x more data to move on stagging nodes than Density-intransit)

From In Situ Analytics to Computational Steering



Conclusion

DataFlow Programming:

- FlowVR
- Google TensorfFlow (Machine learning)
- Apache Flink (Big Data stream + batch processing)

In-situ analytics:

- A paradigm shift that will very likely influence the HPC ecosystem (SW and HW)
- Not standard framework yet
- An opportunity to rethink the use of the I/O budget
- Bring some interactivity into the HPC world. Put the "user in the loop"

HPC and Big Data Convergence ?

Infratructures

- Dedicated platforms combining compute (Cluster + GPUs), data acquisition (Cameras) and visualization (display wall, VR headset) resources
  - Grimage (2002-2013)
  - Equipex Kinovis (since 2014)
  - Led to experiment sharing compute resources for batch and interactive computing through Grid'5000.

- Grid'5000 multi-site experiments (Bordeaux,Grenoble, Orléans)
- Grid'5000 transcontinental experiments (Japan, France)

- Mésocentre Ciment
- EDF & GENCI machines

References

A Flexible Framework for Asynchronous In Situ and In Transit Analytics for Scientific Simulations. Matthieu Dreher, Bruno Raffin. CCGrid'14, May 2014, Chicago

In Situ Statistical Analysis for Parametric Studies. Théophile Terraz, Bruno Raffin, Alejandro Ribes, Yvan Fournier. ISAV2016, Nov 2016, Salt Lake City

Damaris: How to Efficiently Leverage Multicore Parallelism to Achieve Scalable, Jitter-free I/O. Matthieu Dorier, Gabriel Antoniu, Franck Cappello, Marc Snir, and Leigh Orf. In CLUSTER - IEEE International Conference on Cluster Computing. IEEE, September 2012.

Synergistic Challenges in Data-Intensive Science and Exascale Computing. Technical report, DOE ASCAC Data Subcommittee Report, 2013.

FIN