

Toward a Computational Steering Environment for Legacy Coupled Simulations

Nicolas Richart, Aurélien Esnard and Olivier Coulaud

INRIA Futurs - ScAIApplix Project - Bordeaux France

ISPDC'07

1 Introduction

- General Notions about Computational Steering
- Introduction to EPSN Model

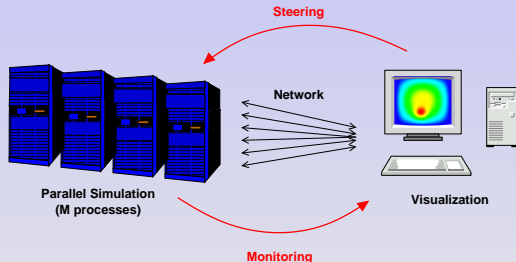
2 Coupled Simulations

- Main issues
- Unique description for each coupled simulations kind

3 Conclusion

Numerical Simulations and Computational Steering

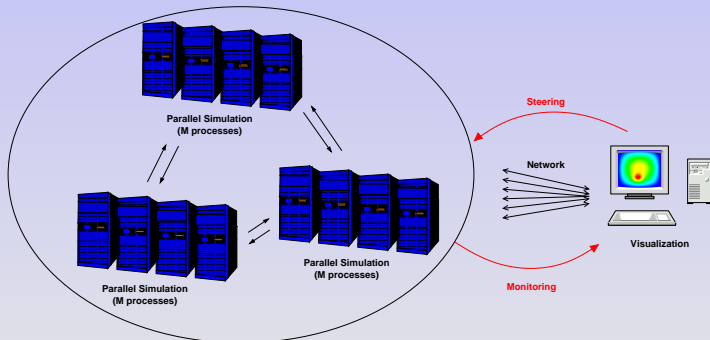
- Traditionally: Numerical simulation → Post-processing (Visualization)
- Computational steering: a more interactive approach
 - Monitoring: Online visualization of intermediate results
 - Steering: Change simulation parameters and data on-the-fly



→ Drive the simulation more rapidly in the right direction

Steering of Coupled Simulations

- Coupled simulations: different models coupled in different distributed codes
- Simulation representation and interaction mechanism
→ how, when, and where safely interacts with the simulations?

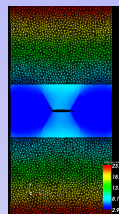


→ Two main kinds of coupled simulations: **M-SPMD** and **Client-Server**

“M-SPMD” Simulations

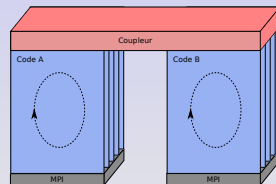
LibMultiScale

- Simulation of crack propagation in materials
- Code A : elastodynamic (macroscopic)
data : mesh (displacement, velocity, ...)
- Code B : molecular dynamics (microscopic)
data : atoms (position, velocity, ...)
- The coupling program is distributed on the 2 codes



M-SPMD

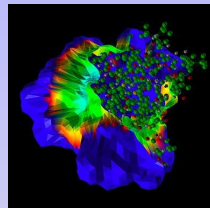
- Multiples SPMD codes with the roughly the same execution flow
- As many time loops as codes



“Client-Server” Simulations

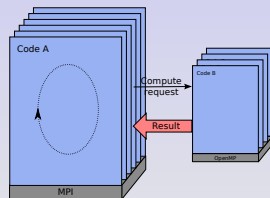
Simbio

- Molecular simulation in fluids
- Code A : molecular dynamic data : atoms (position, velocity)
- Code B : electrostatic computation data : surface mesh, scalar field



Client-Server

- A code (client) which use functionalities of an other code (server)
- Client as a time loop and servers don't have any

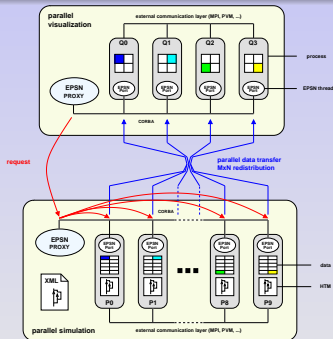


- Problem Solving Environment: SCIRun, COVISE
 - interconnect modules to build new simulations
 - provides visualisation and steering modules
- Steering frameworks: CUMULVS, Visit, EPSN
 - instrumentation of legacy simulations codes
 - dedicated to steer simulations
- Grid and Component frameworks: Cactus, CCA, RealityGrid
 - Some steering possibilities

EPSN Keypoints

Features

- Legacy parallel simulations
- Parallel visualisation capable
- Client/Server relationship
- Communication layer based on CORBA



EPSN Keypoints

Features

- Legacy parallel simulations
- Parallel visualisation capable
- Client/Server relationship
- Communication layer based on CORBA

Abstract model for simulations representation

- EPSN based on an abstract description in 2 parts :
 - Description model:
 - Execution flow description
 - Data description
 - Steering model
 - Steering requests - Interaction availability on tasks

Description Model 1/2

HTM : Hierarchical Task Model to capture the execution flow

- Description based on a task tree
- 4 tasks kinds (grasp main structures):
 - (a) simple, (b) loop, (c) conditional and (d) point
- Date associated to each task
- Task defined by an XML file and by code annotations



Description Model 2/2

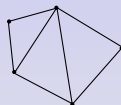
Data we want to steer or visualize

- Geometry description (mesh, particles, grid, etc.)
- Variables
- Data distribution and storage description
- Data access right on tasks
- Description given by an XML file and by code annotations

```
<simulation id="Finite Elements">  
  <data>  
    <object class="mesh" id="FE">  
      <variable id="vertex" />  
      <variable id="connectivity" />  
      <variable id="displacement" />  
      <variable id="velocity" />  
      <variable id="stress" />  
    </object>  
  </data>  
</simulation>
```

geometry

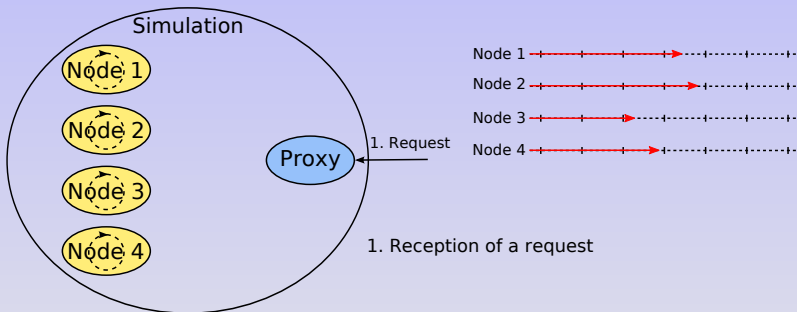
data



displacement
velocity
stress

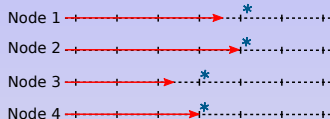
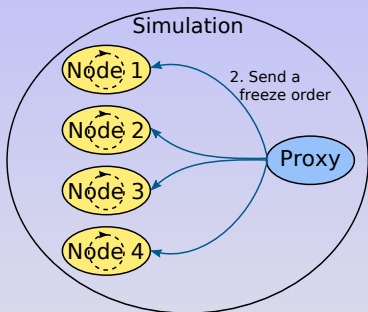
SPMD Coordination Algorithm

Aims : Date of treatment prevision with minimum influence on the code execution



SPMD Coordination Algorithm

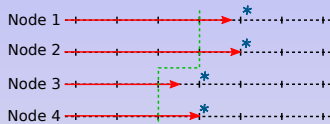
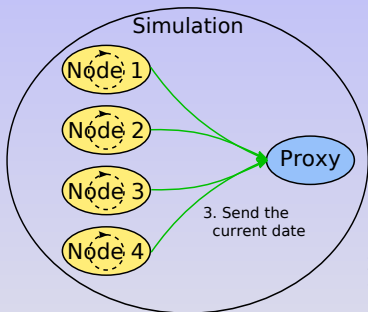
Aims : Date of treatment prevision with minimum influence on the code execution



1. Reception of a request
2. Send a freeze order to the nodes

SPMD Coordination Algorithm

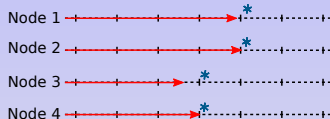
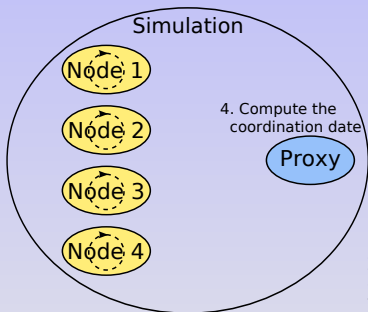
Aims : Date of treatment prevision with minimum influence on the code execution



1. Reception of a request
2. Send a freeze order to the nodes
3. Get the current date from the nodes

SPMD Coordination Algorithm

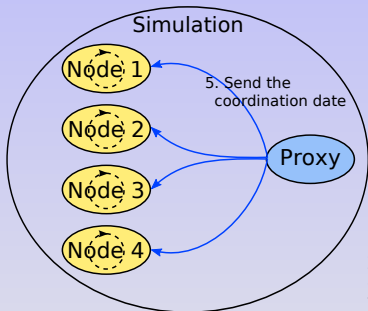
Aims : Date of treatment prevision with minimum influence on the code execution



1. Reception of a request
2. Send a freeze order to the nodes
3. Get the current date from the nodes
4. Computation of the coordination date

SPMD Coordination Algorithm

Aims : Date of treatment prevision with minimum influence on the code execution



1. Reception of a request
2. Send a freeze order to the nodes
3. Get the current date from the nodes
4. Computation of the coordination date
5. Broadcast the coordination date to the nodes

Structure description

- SPMD Simulations : ✓
- M-SPMD Simulations :
 - on codes with the same HTM : ✓
 - on codes with too many differences in HTMs: ✗
- Other simulations kind : ✗

Data description

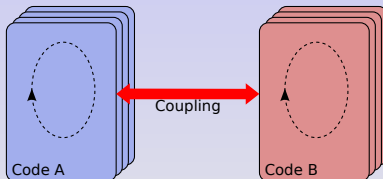
- SPMD Simulations : ✓
- Other simulation kinds : ✗

No way to represent data that are combination of many data on different codes

Problem to Solve 1/2

How to describe M-SPMD and Client-Server simulations to steer them efficiently ?

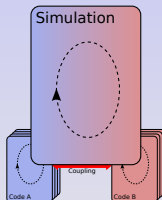
- Need a model to capture the execution flow
 - have a unique date system
 - coordinate precisely treatment
 - ensure data coherency



Problem to Solve 1/2

How to describe M-SPMD and Client-Server simulations to steer them efficiently ?

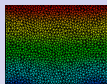
- Need a model to capture the execution flow
 - have a unique date system
 - coordinate precisely treatment
 - ensure data coherency



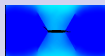
Problem to solve 2/2

How to describe M-SPMD and Client-Server simulations to steer them efficiently ?

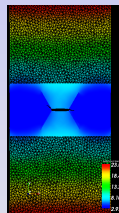
- Data description which afford to have a global view of them
 - ensure data coherency
 - get data as a unique entity to visualize them



Mesh on code A



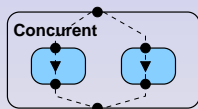
Atoms on code B



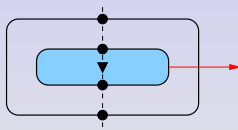
What we want to visualize

Execution flow description

- Build an unique HTM that
 - Describes the main loop of the distributed simulation
 - Describes the common steerable tasks of HTM of both codes
 - Afford to have only one model of description for M-SPMD and Client-Server simulations
- Two new task types
 - Remote : corresponds to remote function call on remote codes
 - Concurrent : represents tasks which are executed at the same time on different codes



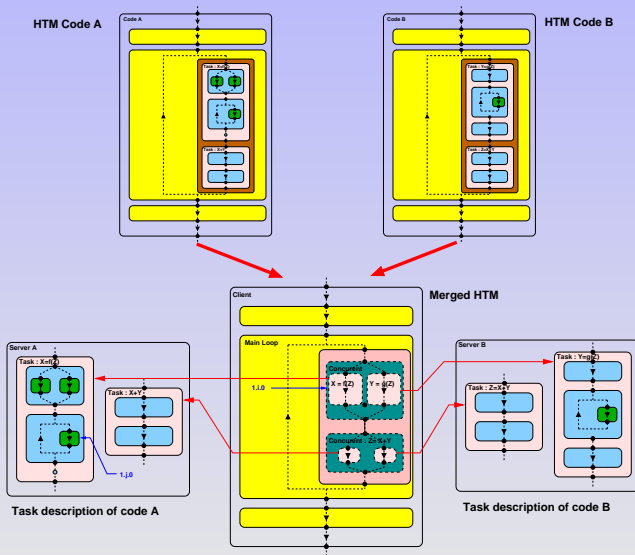
(a) concurrent task



(b) remote task

Sample HTM for a coupled code

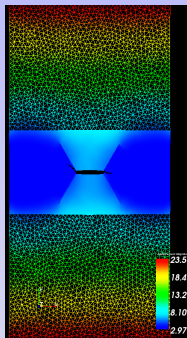
To illustrate :



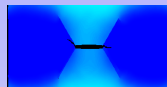
Introduction of a new object type

- Meta-object that grab the description of data present on codes
 - Name of concerned data and variables
 - Task in codes where the data are coherent
- New key word in the XML file: meta-object

Data description for LibMultiScale

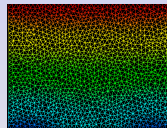


```
<simulation id="LibMultiScale">
<data>
  <meta-object id="displacement">
    <variable id="atoms-displacement">
      object_id="atoms"
      variable_id="X"
      function="Leapfrog"
      task_id="Forces (n+1)"
      location="Atoms"/>
    <variable id="fe-displacement">
      object_id="FE"
      variable_id="displacement"
      function="Leapfrog"
      task_id="Forces (n+1)"
      location="Finite Element"/>
    </meta-object>
  </data>
</simulation>
```

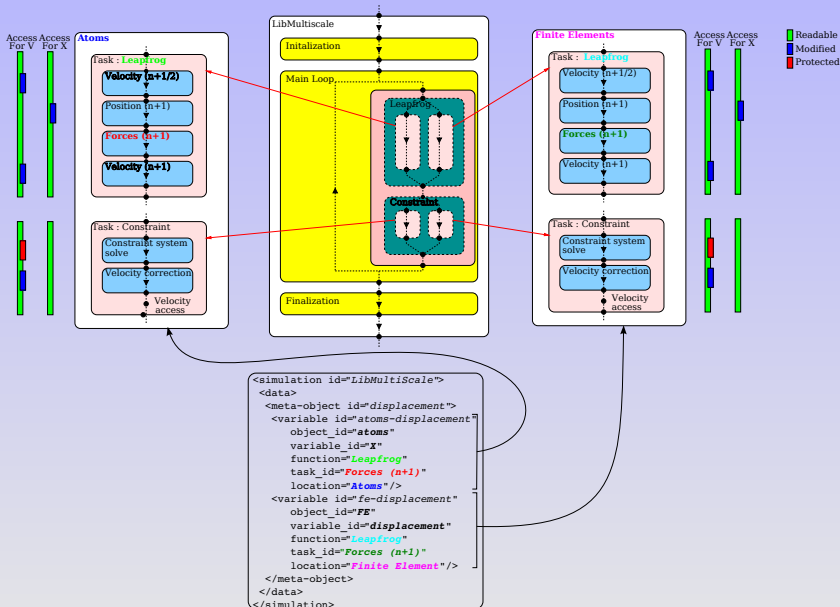


```
<simulation id="Atoms">
<data>
  <object class="points" id="atoms">
    <variable id="X0" />
    <variable id="X" />
    <variable id="velocity" />
  </object>
</data>
</simulation>
```

```
<simulation id="Finite Elements">
<data>
  <object class="mesh" id="FE">
    <variable id="vertex" />
    <variable id="connectivity" />
    <variable id="displacement" />
    <variable id="velocity" />
    <variable id="stress" />
  </object>
</data>
</simulation>
```

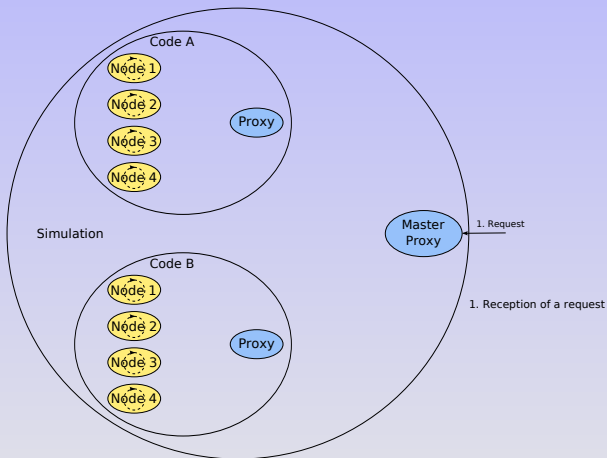


Data description for LibMultiScale



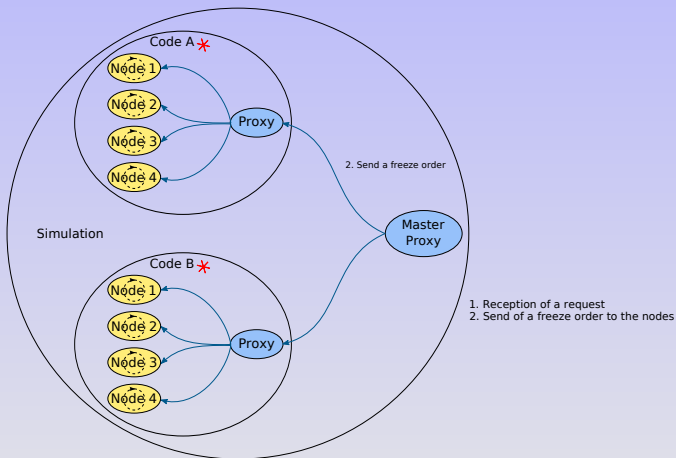
Coordination algorithm for data request

Aim : get coherent data for all codes



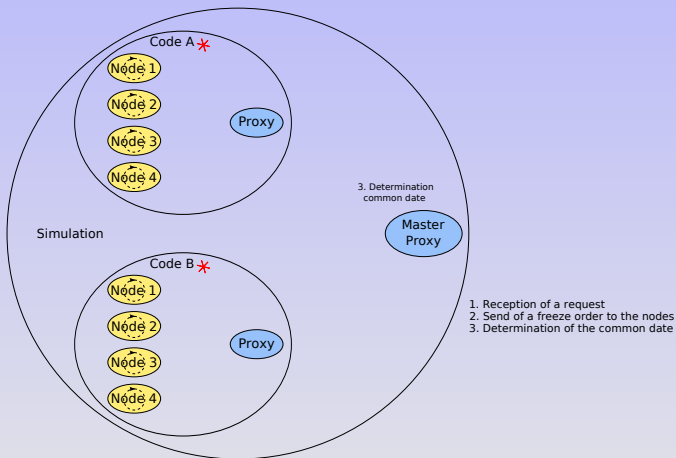
Coordination algorithm for data request

Aim : get coherent data for all codes



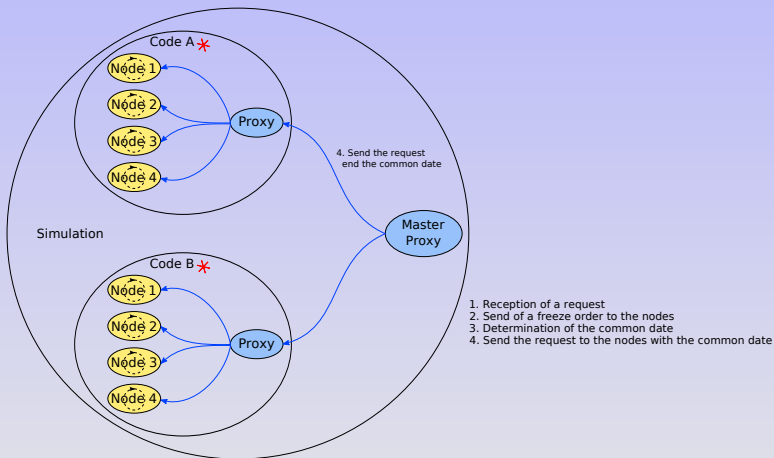
Coordination algorithm for data request

Aim : get coherent data for all codes



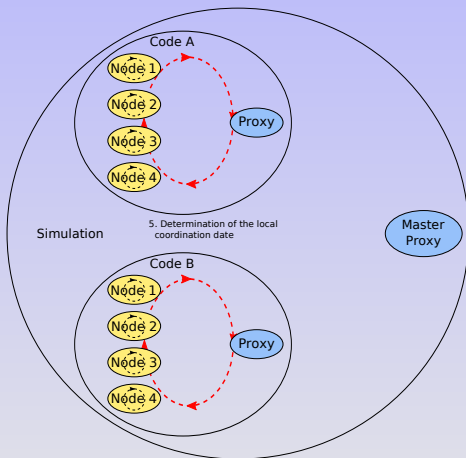
Coordination algorithm for data request

Aim : get coherent data for all codes



Coordination algorithm for data request

Aim : get coherent data for all codes



1. Reception of a request
2. Send of a freeze order to the nodes
3. Determination of the common date
4. Send the request to the nodes with the common date
5. Determination of the local coordination date with the SPMD algorithm

Steering of coupled simulations

- Steering framework for legacy SPMD simulations
Open source project : <http://epsn.gforge.inria.fr/>
- A model to steer coupled simulations
 - Way to describe execution flow of M-SPMD and Client-Server simulations, and to describe distributed data
 - Coordination algorithm to manage coherent treatments in such simulations

Future works

- Test the scalability of this approach
- Test on a real case study