Multi-domain approach to parallel computations in structure dynamics using RADIOSS

Maciek Wronski
Bertrand Maurel
Summary

- Method targets
- Development background and history
- Basic concepts
- Method details
- Application and performance tests
- Future plans and conclusion
Multi-domain approach to parallel computations

- *time step dispersion* - principal bottleneck in explicit computations
- *the smallest time step determines the performance of the whole model*

Target:
- Reduce CPU time by optimising domain decomposition based on time step

Problems:
- Definition of domain frontiers – usually needs user action
- Efficient multi-time step scaling algorithms and resolution methods for frontier nodes
- Efficient communication schemes for data transfert between domains
Multi-domain approach to parallel computations

- Development history and background
  - First implementation – M.Wronski (Mecalog) < 2002
    - Proof of concept - working environment with mono-time step scaling
  - PSA/Mecalog cooperation
    - Phd Thesis : Bertrand Herry (ENS Cachan – 2002)
    - Phd Thesis : Benjamin Bourel (INSA Lyon - 2006)
  - Radioss development and industrialisation
    - Altair Development France (M.Wronski, B.Maurel)
Multi-domain approach to parallel computations

• Basic concepts
Multi-domain approach to parallel computations

- Basic concepts – first ideas:
  - Domain subdivision based on time step (mesh size)
  - Attribute fixed number of processors for each model, optimized for equilibrated CPU charge
  - Communication between master program and radioss models using simple system pipe system (restrictions)

- Main objective:
  - each domain advances with using its own time step
  - only synchronisation points are imposed to enforce precision
Multi-domain approach to parallel computations

- Basic concepts

Interprocess communication principle

\[ t_a > t_b \]
Multi-domain approach to parallel computations

• Basic Concepts
  • Synchronization frequency – imposed by user
  • Free and imposed time steps for each domain:

\[
\begin{align*}
& t^n + 2 \Delta t^n < t_{syn} \Rightarrow \Delta t = \Delta t^n \\
& t^n + \Delta t^n > t_{syn} \Rightarrow \Delta t = t_{syn} - t^n \\
& t^n + 2 \Delta t^n > t_{syn} \& t^n + \Delta t^n < t_{syn} \Rightarrow \Delta t = \frac{1}{2} (t_{syn} - t^n)
\end{align*}
\]
Multi-domain approach to parallel computations

• Fixed CPU number for each domain = Problem:
  • The domain which is in advance with the others must wait for interpolated data sent by the master program
  • Consequence:
    • Limited speed up possibilities using large number of processors
    • Fixed, “optimized” CPU numbers per model is a false good idea

• Solution :
  • Attribute the total number of processors to each domain
  • Optimize interprocess communication : TCP/IP sockets instead of pipes

• Result :
  • Idle processes no longer take CPU time => all power is distributed for active processes only
  • Very good speed up, except for master program

⇒ Idle CPU time
Multi-domain approach to parallel computations

• Solution:
  • Attribute the total number of processors to each domain
  • Optimize interprocess communication: TCP/IP sockets instead of system pipes
  • Reduce MPI communications for idle processes (SPMD)

• Result:
  • Idle processes no longer take CPU time => all computation power is only distributed for active processes
  • Very good speed up, except for master program
  • No additional development needed to optimize / equilibrate CPU load between sub-domains
Multi-domain approach to parallel computations

Radioss 1 — Rad2rad — Radioss 2

**Activation**

- $t_1^n < t_2^n \Rightarrow$ domain 1 active
- $u_2^n_{\text{free}} = \alpha u_2^{n-1}_{\text{free}} + (1-\alpha) u_2^n_{\text{free}}$
- $u_1^n_{\text{link}} = f(u_1^n_{\text{free}}, u_2^n_{\text{free}}, \lambda)$

**Desactivation**

- $t_1^n > t_2^n \Rightarrow$ domain 2 active
- $u_1^n_{\text{free}} = \alpha u_1^{n-1}_{\text{free}} + (1-\alpha) u_1^n_{\text{free}}$
- $u_2^n_{\text{link}} = f(u_1^n_{\text{free}}, u_2^n_{\text{free}}, \lambda)$

**Time Integration**

- $t_1^n < t_2^n \Rightarrow$ domain 1 active

**Time Integration**

- $t_1^n > t_2^n \Rightarrow$ domain 2 active
Multi-domain approach to parallel computations

• Method details
  • Actual status:
    • All starter and engine radioss models are run separately using individual input files
    • Separate input file for master rad2rad program
    • Rad2rad master program must be launched manually in parallel with radioss engine processes
    • Output files contain separate results of each model

• Current development
  • Common input file containing all sub domains and their interface links
  • Automated launch of starter and engine
  • Common output file with composed data of all sub domains
Multi-domain approach to parallel computations

• Method details

• Inter-domain interface possibilities
  • Common (coincident) nodes
  • Non-conform mesh (tied condition)
  • Beam to shell tied interface
  • Shell to shell non-symmetric contact interface (slave / master formulation)

• Domain decomposition must be done manually by the user
• Link interfaces must be clearly specified by node groups on domain frontiers
Multi-domain approach to parallel computations

- Application test

Simple crash box:

- 1 model with 2 parts
- 1 RADIOSS run

Imposed displacement

- 2 models with 1 part each
- 2 RADIOSS runs + Rad2Rad

Common section between both models with the same number of nodes and the same node coordinates
Multi-domain approach to parallel computations

- Application test

**Simple crash box:**

Same deformation between both calculation methods: the classical one and the multi-scaled one.

- 1 model with 2 parts
  - 1 RADIOSS run
- 2 models with 1 part each
  - 2 RADIOSS runs + Rad2Rad
Multi-domain approach to parallel computations

- Performance test case: Head Impact

PART1: 13803 shell elements
  time step: 2.97e-7 sec

PART2: 303648 shell elements
  time step: 8.00e-7 sec
Multi-domain approach to parallel computations

- Performance test case:
  Head Impact
Multi-domain approach to parallel computations

- Head Impact – Quality of results

Head acceleration vs time:

![Graph showing head acceleration vs time]
## Multi-domain approach to parallel computations

- **Head impact**: performances (elapsed time):

<table>
<thead>
<tr>
<th></th>
<th>1 CPU</th>
<th>16 CPUs (2 nodes)</th>
<th>32 CPUs (4 nodes)</th>
<th>64 CPus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monodomain</td>
<td>63500 s</td>
<td>5420 s</td>
<td>3470 s</td>
<td>2790 s</td>
</tr>
<tr>
<td>Multidomain case 1</td>
<td>27260 s</td>
<td>2740 s</td>
<td>2100 s</td>
<td></td>
</tr>
<tr>
<td>Multidomain case 2</td>
<td></td>
<td>2584 s</td>
<td>1670 s</td>
<td>1390 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1 CPU</th>
<th>16 CPUs</th>
<th>32 CPUs</th>
<th>64 CPus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed up case 1</td>
<td>2.33</td>
<td>2.00</td>
<td>1.65</td>
<td></td>
</tr>
<tr>
<td>Speed up case 2</td>
<td>2.30</td>
<td>2.09</td>
<td>2.08</td>
<td>2.01</td>
</tr>
</tbody>
</table>

*Test step factor = (dt domain 1) / (dt domain2) = 2.6
Monodomain = complete model with common time step
Speed UP = Temps elapsed Monodomain / Temps elapsed Multidomains

Case 1: using synchronisation by sockets (total CPU number for all processes)
Case 2: Case 1 + optimisation of internal interprocess MPI communications (SPMD)
Multi-domain approach to parallel computations

• Conclusions
  • Already satisfying performance and robustness
  • A lot of work is still needed to create ergonomic user interface

• Future work
  • common input / output
  • automated launching
  • Parallelization of the master rad2rad program (if needed)

• Potential applications:
  • Car component optimization / rupture analysis with the full car model
  • Structure + SPH / ALE mixed models (tank security)
  • Airbag behaviour analysis