Recent Progress of Parallel SAMCEF with MUMPS
MUMPS User Group Meeting 2013

Jean-Pierre Delsemme
Product Development Manager

Clamart, May 29th and 30th 2013
Summary

- SAMCEF, a brief history
- Co-simulation, a good candidate for parallel processing
- MAAXIMUS, The "Gigadof" challenge
- SLEPC/MUMPS, a parallel eigenvalue solver
- Test the limit of your machine
- Conclusions
SAMCEF, a brief history

- SAMCEF: general purpose FE software
- First line of code written in 1967 at University of Liege
- First customer in aeronautic in 1977
- Creation of SAMTECH in 1986 (9 people)
- Implementation of BCSLIB in 1998
- Implementation of MUMPS 2005
- Acquisition of 60% of SAMTECH by LMS in 2011 (9 subsidiaries, 300 people, 30 MEuro per year)
3rd participation...

Recent Progress of Parallel SAMCEF with MUMPS
MUMPS User Group Meeting 2013

Jean-Pierre Delisserme
Product Development Manager

Clamart, May 29th and 30th 2013
### What is parallel computing in SAMCEF?

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
<th>Use of MUMPS for</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASEF</td>
<td>Linear static analysis</td>
<td>linear system resolution</td>
</tr>
<tr>
<td>MECANO</td>
<td>Non-linear static and dynamic analysis</td>
<td>Elements generation / System resolution / Results archive (depending on PARALL)</td>
</tr>
<tr>
<td>DYNAM</td>
<td>Modal analysis</td>
<td>Use of SLEPC and MUMPS for eigenvalues problem resolution</td>
</tr>
<tr>
<td>STABI</td>
<td>Linear buckling analysis</td>
<td>Use of SLEPC and MUMPS for eigenvalues problem resolution</td>
</tr>
<tr>
<td>BACON</td>
<td>Ray Tracing (.MCRT command)</td>
<td>Distribution of ray tracing computation</td>
</tr>
</tbody>
</table>
Summary

- SAMCEF, a brief history
- Co-simulation, a good candidate for parallel processing
- MAAXIMUS, The "Gigadof" challenge
- SLEPC/MUMPS, a parallel eigenvalue solver
- Test the limit of your machine
- Conclusions
Co-simulation principle
- 2 fields of unknown
- 2 governing equations
- Function of time

Staggered solution scheme
- Field are exchanged at time step level
- Approximation due to delay of the information
- Can lead to numerical problem

\[ F(x, y, t) = \begin{pmatrix} f(x, y, t) \\ g(x, y, t) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \]
### MECANO-MECANO Co-simulation

- **Monolithic solution scheme**
  - Coupling at iteration level
  - All equations treated at once

- **Master – slave approach**
- **Internal dof's "11"**
- **Interface dof's "22"**

- At each iteration, exchange of:
  - Interface unknowns $q_2$, $R_2$
  - Schur complement of the interface and linear constrains

---

#### LMS

**LMS**

A Siemens Business

8. MUMPS User Group Meeting 2013

Clamart, May 29th and 30th 2013
Example: 4 wheel car model
- 400 000 dof's per tires
- 2 phase loading
  - Tire inflate without contact
  - Apply gravity on wheel to force contact with ground

Non optimum automatic domain decomposition
MECANO-MECANO Co-simulation

- Example: 8 wheel car model
  - 400,000 dof's per tires
  - Impressive computation time reduction

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Time (min)</th>
<th>Speed up</th>
</tr>
</thead>
<tbody>
<tr>
<td>monolithic 1 node, 4 procs</td>
<td>182</td>
<td>1.0</td>
</tr>
<tr>
<td>monolithic 2 nodes, 8 procs</td>
<td>45</td>
<td>4.0</td>
</tr>
<tr>
<td>monolithic 4 nodes, 16 procs</td>
<td>22</td>
<td>8.3</td>
</tr>
<tr>
<td>monolithic 8 nodes, 32 procs</td>
<td>28</td>
<td>6.5</td>
</tr>
<tr>
<td>New Method 5 nodes, 18 procs</td>
<td>12</td>
<td>15.1</td>
</tr>
<tr>
<td>New Method 9 nodes, 34 procs</td>
<td>7</td>
<td>26.0</td>
</tr>
</tbody>
</table>
Summary

- SAMCEF, a brief history
- Co-simulation, a good candidate for parallel processing
- MAAXIMUS, The "Gigadof" challenge
- SLEPC/MUMPS, a parallel eigenvalue solver
- Test the limit of your machine
- Conclusions
Maaximus, the "Gigadof" challenge

- Calculate a fuselage made of, as much as possible, identical single barrels
- ~4,500,000 dof's per barrel
- Identification of new limitations
- "standardization" of long long integer version
Maaximus, the "Gigadof" challenge

3 sections of fuselage
- 13,956,620 dof's
- 1,891,176 shell elem.
- 23,220,111 elem. In total
- 7 h 24' on a cluster of 8 nodes, up to 100% of prescribed load
- 5 time steps, 1 rejected, 19 iterations
- Intel(R) Core(TM) 2.67 GHz
- 12 Gb per node
Maaximus, the "Gigadof" challenge

4 sections of fuselage

- 18,580,217 dof's
- 2,521,568 shell elem.
- 3,092,810 elem. In total
- 28 h 30' on a cluster of 10 nodes.
- 7 time steps, 5 rejected,
- 57 iterations up to 91% of load
- Intel(R) Core(TM) 2.67 GHz
- 12 Gb per node
Maaximus, the "Gigadof" challenge

6 sections of fuselage

- 27,823,805 dof's
- 3782746 shell elem.
- 4,635,044 elem. In total
- 37h 30’ on a cluster of 8 nodes.
- 7 time steps, 3 rejected,
  41 iterations up to 100% of load
- Intel(R) Core(TM) 2.67 GHz
- 12 Gb per node
Maaximus, the "Gigadof" challenge

7 sections of fuselage
- 37,127,423 dof's
- 5044328 shell elem.
- 6,177,110 elem. In total
- 0h 0’ on a cluster of 8 nodes.
- 0 time steps, 0 rejected,
  0 iterations up to 0% of load
- Intel(R) Core(TM) 2.67 GHz
- 12 Gb per node

**Segmentation fault in METIS !**
Maybe long long integer needed
Can anyone help ?
Summary

- SAMCEF, a brief history
- Co-simulation, a good candidate for parallel processing
- MAAIMUS, The "Gigadof" challenge
- SLEPC/MUMPS, a parallel eigenvalue solver
- Test the limit of your machine
- Conclusions
Parallel Eigenvalue Solver

- Since version 15 (partially available in 14) an interface with SLEPC solver has been implemented.
- SLEPC library is a public domain software developed by Universidad Politecnica de Valencia, Spain.
- SLEPC perform mathematical operations in parallel based on:
  - PETSC for matrix – vector product.
  - MUMPS for linear system solve.
Parallel Eigenvalue Solver

Benefit for users

- Faster computation
- Possibility to use several computers and than more memory

Possible disadvantage

- MUMPS is able to work out-of-core but PETSC isn’t.
- On a given computer (fixed memory) there is maximum problem size that is lower than BCSLIB maximum size.
Parallel Eigenvalue Solver

- Scalable test case: parametric model

Comparison of a sequential BCSLIB run with parallel 4 proc. SLEPC run.

Intel(R) Xeon(R) @ 2.66GHz, 48 GBytes of memory, 8 cores

![Graph showing speedup and elapsed time for different problem sizes.](image)
Parallel Eigenvalue Solver

- Scalable test case: parametric model
- Allows to use several nodes in a cluster and than more memory.
  - Intel(R) Core(TM) i7-2600 @ 3.40GHz, 16 Gbytes of memory per node, 4 cores
  - 4 threads per node; 100 eigenvalues

![Graph showing elapsed solver time against degrees of freedom for 4 and 8 nodes.](image-url)
Summary

- SAMCEF, a brief history
- Co-simulation, a good candidate for parallel processing
- MAAXIMUS, The "Gigadof" challenge
- SLEPC/MUMPS, a parallel eigenvalue solver
- Test the limit of your machine
- Conclusions
Customer question...

- On my machine, what is the maximum model size I can analyze?
  or
- To analyze my model, what is the minimum machine configuration (cheapest) I should acquire?

Answer: "test it yourself"
Parametric model

- Realistic "fuselage" model
- Made of shell elements
- With skin, frames and stringers

- 2 types of analysis
  - A non-linear static loading
    2 iterations (2 factors, 2solves)
  - An eigenvalue analysis
    2 eigenvalues, 1 factor, n solves (18 or 27)
Design of experiment

**Hardware**

- Cluster of 16 (identical) nodes
- Processor: Intel(R) Core(TM) i7-3930K CPU @ 3.20GHz
- # Cores: 6
- MemTotal: 32855900 kB
- SwapTotal: 33559776 kB
Design of experiment

Software

- 10 model sizes [1-10] million of dof's
- 2 solvers BCSLIB and MUMPS or SLEPC/MUMPS
  - Sequential BCSLIB with 1, 2, 4 or 6 MKL Threads
  - Sequential MUMPS with 1, 2, 4 or 6 MKL Threads
  - Parallel MUMPS on 2 procs. with 1 or 2 MKL Threads
  - Parallel MUMPS on 3 procs. with 2 MKL Threads
  - Parallel MUMPS on 4 procs. with 1 MKL Threads
  - Parallel MUMPS on 6 procs. with 1 MKL Threads
    9 among 14 possible cases
- Total of $10 \times 2 \times (4 + 4 + 2 + 1 + 1 + 1 + 1) = 260$ analyses
- Total of 19 days 14h 03min 26.05sec…
Model properties

Minimum core processing

27. MUMPS User Group Meeting 2013

Clamart, May 29th and 30th 2013
Non-linear static analysis

![Graph showing solver elapsed time for different MUMPS and BCSLIB versions with varying processors.](image)

- MUMPS 1th 1Proc
- MUMPS 1th 2Proc
- MUMPS 1th 4Proc
- MUMPS 1th 6Proc
- MUMPS 2th 1Proc
- MUMPS 2th 2Proc
- MUMPS 2th 3Proc
- MUMPS 4th 1Proc
- MUMPS 6th 1Proc
- BCSLIB 1th
- BCSLIB 2th
- BCSLIB 4th
- BCSLIB 6th

**Axes:**
- Solver elapsed time (sec.)
- Solver elapsed time (sec.)

**Legend:**
- MUMPS 1th 1Proc
- MUMPS 1th 2Proc
- MUMPS 1th 4Proc
- MUMPS 1th 6Proc
- MUMPS 2th 1Proc
- MUMPS 2th 2Proc
- MUMPS 2th 3Proc
- MUMPS 4th 1Proc
- MUMPS 6th 1Proc
- BCSLIB 1th
- BCSLIB 2th
- BCSLIB 4th
- BCSLIB 6th
Non-linear static analysis

Total Elapsed time, 8 Mdof

Total Elapsed Time (sec.)

Speedup

BCSLIB 1th
BCSLIB 2th
MUMPS 1th 4Proc
MUMPS 1th 1Proc
BCSLIB 4th
BCSLIB 6th
MUMPS 1th 2Proc
MUMPS 2th 3Proc
MUMPS 2th 1Proc
MUMPS 2th 2Proc
MUMPS 4th 1Proc
MUMPS 6th 1Proc
MUMPS 1th 6Proc

29. MUMPS User Group Meeting 2013

Clamart, May 29th and 30th 2013
Non-linear static analysis

Total Elapsed time, 4 M dof
Eigenvalue analysis
Eigenvalue analysis

Total Elapsed time, ~8 M dof

![Graph showing total elapsed time and speedup for different processors and libraries.](image_url)
Eigenvalue analysis

Total Elapsed time, 4 Mdof

- Speedup
- Total Elapsed Time (sec.)
- MUMPS 2th 1Proc
- BCSLIB 1th
- BCSLIB 4th
- BCSLIB 2th
- MUMPS 1th 2Proc
- BCSLIB 1th
- BCSLIB 6th
- MUMPS 1th 1Proc
- BCSLIB 2th 1Proc
- MUMPS 2th 1Proc
- MUMPS 4th 1Proc
- MUMPS 6th 1Proc
- MUMPS 2th 3Proc
- MUMPS 1th 6Proc
- MUMPS 2th 3Proc
- MUMPS 1th 4Proc
Conclusions

- **SAMCEF still improves parallel computing**
  - Better parallel processing of non-linear analysis
  - Co-simulation helps parallel processing
  - Parallel eigenvalue solver SLEPC/MUMPS
  - Need help for METIS `long long integer`

- **Test the limits of your machine**
  - Parametric model available
  - No need to try too far out of memory
  - Optimum balance MKL thread / MUMPS proc to find
  - SLEPC shows good performance if it can run in core
  - Real need for efficient eigenvalue solver for large problem
Thank You
If you come to Liège