High Performance Computing Electromagnetics Challenge: solving tens of millions of unknowns

J.C. Mourino[1] A. Gomez[1]
L. Landesa[2] J.M. Taboada[2]
F. Obelleiro[3] J.L. Rodriguez[3]

[1]Centro de Supercomputacion de Galicia

[2]Universidad de Extremadura

[3]Universidad de Vigo

Iberian GRID Infraestructure Conference, 2008

Outline

Outline

Method of Moments

Solve the integral expressions from Maxwell equations

$$\vec{E}^{i}_{tan}(\vec{r}) = jk\eta \iint_S \vec{J}_s(\vec{r}')G(\vec{r},\vec{r}')ds' - \frac{\eta}{jk}\nabla_s \iint_S \left[\nabla'_s \cdot \vec{J}_s(\vec{r}')\right] G(\vec{r},\vec{r}')ds'$$

where $G(\vec{r},\vec{r}')$ denotes the free space Green's function and is defined as:

$$G(\vec{r},\vec{r}') = \frac{e^{-jk|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|}$$

Method of moments

Expansion of the unknown on a set of geometrical basis functions:

$$\vec{J}_s(\vec{r}') = \sum_{j=1}^{N} I_j \vec{f}_j(\vec{r}')$$

Method of Moments

Linear system of equations

$$ZI = V$$

Z is a $N \times N$ matrix (Impedance Matrix)
I is a $N \times 1$ vector (unknown)
V is a $N \times 1$ vector (excitation)

Computational complexity

1. Solving $ZI = V$ with matrix factorization or matrix inversion
   - $O(N^2)$ in memory
   - $O(N^3)$ in CPU time
2. Solving $ZI = V$ with iterative methods (e.g. GMRES)
   - $O(N^2)$ in memory
   - $O(N^2)$ in CPU time

Method of Moments

Linear system of equations

$$ZI = V$$

Z is a $N \times N$ matrix (Impedance Matrix)
I is a $N \times 1$ vector (unknown)
V is a $N \times 1$ vector (excitation)

Computational complexity

1. Solving $ZI = V$ with matrix factorization or matrix inversion
   - $O(N^2)$ in memory
   - $O(N^3)$ in CPU time
2. Solving $ZI = V$ with iterative methods (e.g. GMRES)
   - $O(N^2)$ in memory
   - $O(N^2)$ in CPU time

Method of Moments

Linear system of equations

$$ZI = V$$

Z is a $N \times N$ matrix (Impedance Matrix)
I is a $N \times 1$ vector (unknown)
V is a $N \times 1$ vector (excitation)

Computational complexity

1. Solving $ZI = V$ with matrix factorization or matrix inversion
   - $O(N^2)$ in memory
   - $O(N^3)$ in CPU time
2. Solving $ZI = V$ with iterative methods (e.g. GMRES)
   - $O(N^2)$ in memory
   - $O(N^2)$ in CPU time

F-18 Radar Cross Section (RCS) analysis

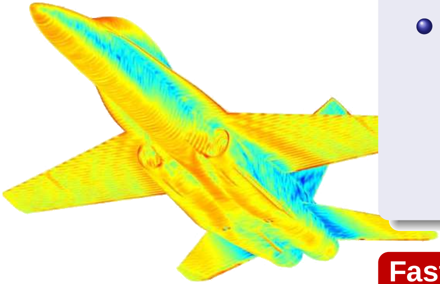


Figure: F18 currents for a nose plane wave incidence

Bistatic RCS at 1GHz with MoM

- Memory: 4TB
- CPU time:
  - SETUP: Several years
  - Solution
    - Factorization: Several years
    - Iterative solution: Several days

Fast Multipole Methods

Setup and solution are obtained in less than two hours requiring a few GB of memory in a conventional PC.

Fast Multipole Methods in Electromagnetics

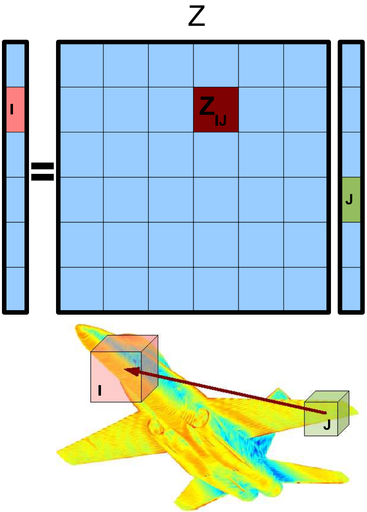
Computational Complexity

- Memory can be reduced to $O(N^{3/2})$ or less
- CPU time can be reduced to $O(N^{3/2})$ for an iterative solver
- SETUP time is from $O(N)$ to $O(N^{3/2})$

Multilevel versions

- Memory order $O(N \log N)$
- CPU time order $O(N \log N)$
- SETUP time order $O(N \log N)$

Grouping of interactions



Grouping of geometry

- Geometry is clustered in a set of separated groups
- Typically, octree partition is applied

Interactions between groups

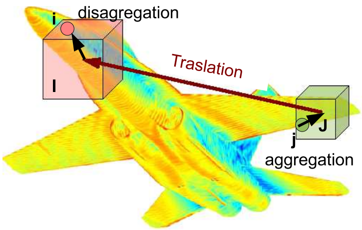
- Matrix Z is divided based on the geometry clustering
- Interactions between groups are represented by blocks of Z

Multipoles in Z_{IJ}

Interaction between element $i \in I$ and $j \in J$

Partial interaction between elements of clusters I and J (elements of Z_{IJ}) is decomposed into::

1. Aggregation
2. Translation
3. Disaggregation



Sequencing of steps

In FMM the previous steps are performed sequentially

1. All the elements i of each group are aggregated
2. The aggregation in each group is translated to all the other groups
3. Finnally, the calculated contribution in each group is disaggregated: Contribution in element j

Reduction in cost

Translation in the spectral domain

Translation in the spectral domain is a DIAGONAL operator. Using also a spectral transform in the groups, matrix $Z_I J$ can be decomposed as:

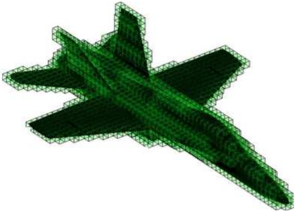
$$Z_{IJ} = A_I^H T_{IJ} A_J$$

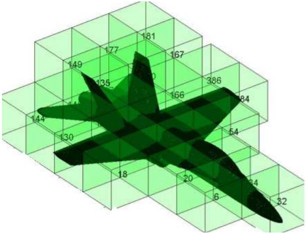
where

1. A_J is a full matrix that makes the aggregation of group J
2. T_{IJ} is a diagonal matrix that makes the translation between groups I and J
3. Disaggregation is the hermitic operator of the aggregation

Minimal Cost – Group size

$$Z_{IJ} = A_I^H T_{IJ} A_J$$





Aggregation: *A* Full matrix

Large Groups Full large matrices: $O(N^2)$

Small Groups Small matrices: $O(N)$

Translation: *T* Diagonal matrix

Large Groups Few diagonal matrices: $O(N)$

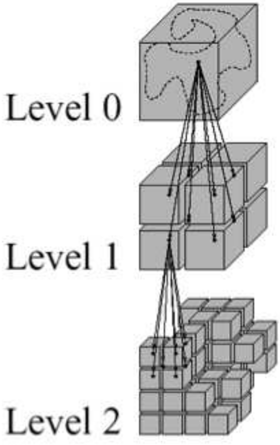
Small Groups A lot of translations: $O(N^2)$

Tradeoff

If number of groups: $O(\sqrt{N})$. Then, Memory and CPU become $O(N^{3/2})$.

The Multilevel Fast Multipole Method

Recursive implementation of Fast Multipole Method



Two new operators: Vertical translation between levels

1. Interpolation
2. Anterpolation

Computational Cost

Memory and CPU costs are $O(N \log N)$

Outline

Several drawbacks of the Multilevel implementations

Scalability

Scalability is limited by

- Heavy load unbalance
- Amdahl's Law

Memory limitations

Several structures are need to be common to all processors

- Memory footprint
  - Translation operators in low levels
  - Interpolation/Anterpolation operators in low levels
  - ...

Solutions to the different drawbacks

Improvements

- Schemes to improve the load balance
- In-core calculation of some structures
  - Increasing serial fraction: Reduction of scalability
  - Load unbalance

Number of processors

In a distributed system Multilevel Fast Multipole Methods in electromagnetics are limited.

- A maximum of 8, 16 or 32 processors
- Low efficiencies are achieved

Challenges in Computational Electromagnetics

RCS of a conducting sphere

W.C. Chew, 2003

Diam 100λ

Unk 10 millions $(10,002,828)$

Gurel, 2007

Diam 192λ

Unk 30 millions $(33,791,232)$

Gurel, 2007 Late

Diam 210λ

Unk 40 millions $(41,883,648)$

Landesa et al, 2008

Diam 200λ

Unk 30 millions $(32,411,106)$

[??], 2008,2009

Diam $> 350\lambda$

Unk $>$ 100 millions

Challenges in Computational Electromagnetics

RCS of a conducting sphere

W.C. Chew, 2003

Diam 100λ

Unk 10 millions $(10,002,828)$

Gurel, 2007

Diam 192λ

Unk 30 millions $(33,791,232)$

Gurel, 2007 Late

Diam 210λ

Unk 40 millions $(41,883,648)$

Landesa et al, 2008

Diam 200λ

Unk 30 millions $(32,411,106)$

[??], 2008,2009

Diam $> 350\lambda$

Unk $>$ 100 millions

Challenges in Computational Electromagnetics

RCS of a conducting sphere

W.C. Chew, 2003

Diam 100λ

Unk 10 millions $(10,002,828)$

Gurel, 2007

Diam 192λ

Unk 30 millions $(33,791,232)$

Gurel, 2007 Late

Diam 210λ

Unk 40 millions $(41,883,648)$

Landesa et al, 2008

Diam 200λ

Unk 30 millions $(32,411,106)$

[??], 2008,2009

Diam $> 350\lambda$

Unk $>$ 100 millions

Challenges in Computational Electromagnetics

RCS of a conducting sphere

W.C. Chew, 2003

Diam 100λ

Unk 10 millions $(10,002,828)$

Gurel, 2007

Diam 192λ

Unk 30 millions $(33,791,232)$

Gurel, 2007 Late

Diam 210λ

Unk 40 millions $(41,883,648)$

Landesa et al, 2008

Diam 200λ

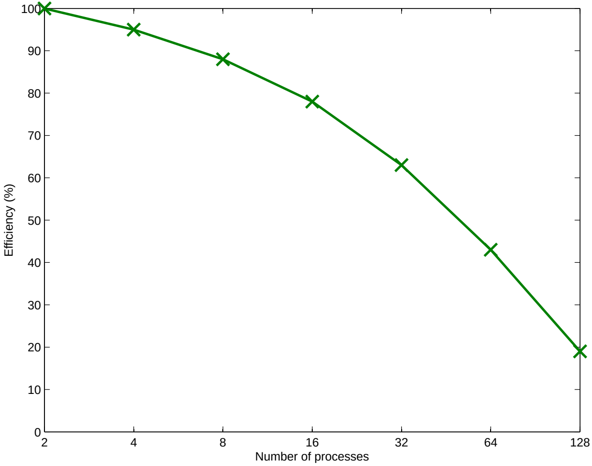
Unk 30 millions $(32,411,106)$

[??], 2008,2009

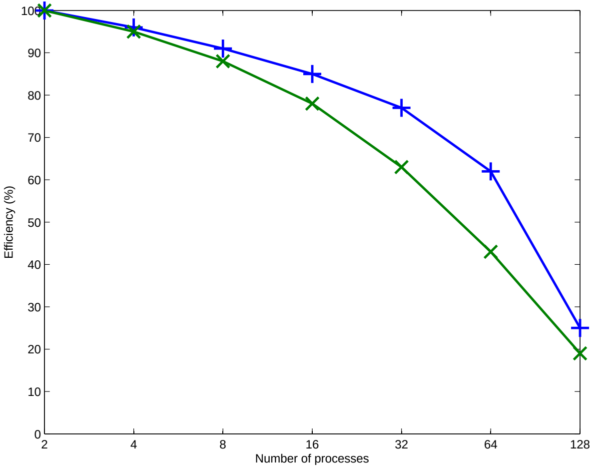
Diam $> 350\lambda$

Unk $>$ 100 millions

Ergul Multilevel FMM performance



Ergul Multilevel FMM performance



Outline

HEMCUVE ++ foundations

Electromagnetic methods implemented

- Single Level Fast Multipole Method
- Multilevel Fast Multipole Method

Parallel implementations

Shared Memory	OpenMP implementation
Distributed Memory	MPI implementation
Mixed Memory	Hybrid MPI/OpenMP implementation

Language

HEMCUVE ++ is implemented in C++

Parallel performance of HEMCUVE ++

Implementations

MPI	Very high efficiency
OpenMP	High efficiency
MPI/OpenMP	High efficiency

Multilevel FMM

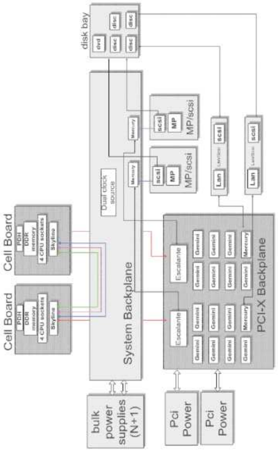
- Parallel efficiency is similar to other implementations
- Maximum scalability: 16 to 32 processes

Single level FMM

- Specific parallel implementation
- Parallel efficiency is very high
- Maximum scalability: 512 to 1024 processes, assured

Outline

Finis Terrae Architecture



142 cc-NUMA Integrity rx7640 nodes

- 8–dual core Intanium-2 Montvale processors
- 128GB memory
- Infiniband network
- Linux SLES 10
- 2 additional Superdome Integrity nodes

memory/CPU ratio

8GB/CPU minimum

Finis Terrae

More than 2500 cores and more than 19TB of memory

Finis Terrae Challenges

Challenges

Before 1 April 2008 (release of Finis Terrae), CESGA planned 4 computational challenges

- High complex problems
- Finis Terrae Time-to-solution

Challenges date

Challenges have been scheduled in one week in one week of:

1. From February 11 to February 17
2. From February 18 to February 24

HEMCUVE Challenge

HEMCUVE Challenge

- Electromagnetics Challenge
- Evaluation of the capabilities of Finis Terrae to beat the WORLD RECORD
- Very high efficiency: For using hundreds of GB and hundreds of processes
- Intensive use of resources: Memory, network and CPU

Objectives

1. Measurement of the performance of HEMCUVE code
2. Analysis of an electromagnetic problem with tens of millions of unknowns

Outline

Selection of the electromagnetics method

Multilevel Fast Multipole Method

- Poor scalability
- Load unbalance
- Great footprint in very large problems and with many processors

Selection of the electromagnetics method

Single Level Fast Multipole Method

- Good scalability
- Medium footprint
- Low dependence of memory footprint with the number of processors
- Specific parallelization of HEMCUVE:
  - Smart management of communications
  - No explicit synchronization between processes
  - Fine tuning of the code: Superscalability with 8 processes

Summarizing

- Single Level FMM is able to take advantage of large amounts of resources
- Multilevel FMM is not

Challenge week (From February 11 to February 15)

Monday

Tuesday

Wednesday

Thursday

Friday

Challenge week (From February 11 to February 15)

Monday

Tuesday

Wednesday

Thursday

Friday

Challenge week (From February 11 to February 15)

Monday

Tuesday

Wednesday

Thursday

Friday

Challenge week (From February 11 to February 15)

Monday

Tuesday

Wednesday

Thursday

Friday

Challenge week (From February 11 to February 15)

Monday

Tuesday

Wednesday

Thursday

Friday

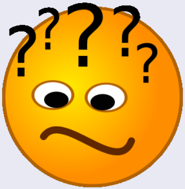
Challenge week (From February 11 to February 15)

Monday

Tuesday

Wednesday

Thursday



Friday

Challenge week (From February 11 to February 15)

Monday

Tuesday

Wednesday

Thursday

Friday

Challenge week (From February 11 to February 15)

Monday

Tuesday

Wednesday

Thursday

Friday

Main problems < 1 >

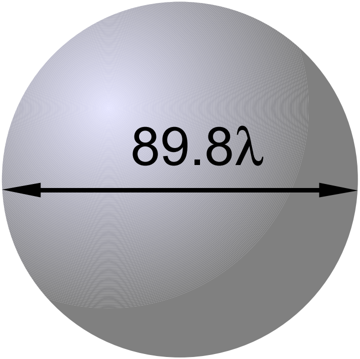
NaN

Main problems $< 2 >$

Load unbalance with large number of processors

- Poor performance of the run time for non main operations
- Bug in the distribution of cells when a large number of processors are involved
- Very easy solution

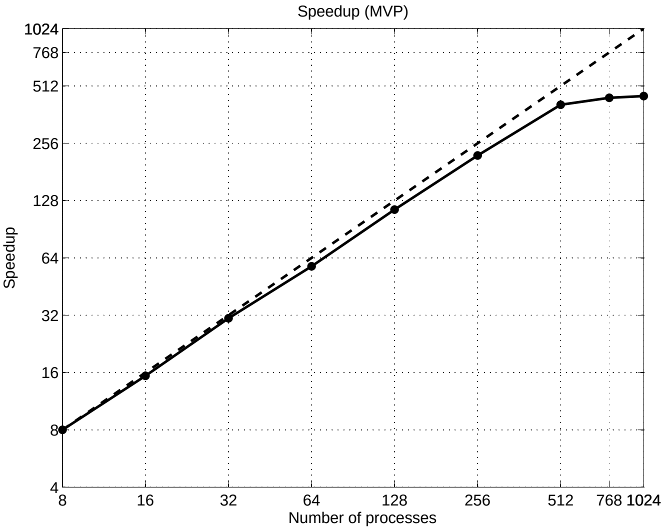
More than 7 millions of unknowns



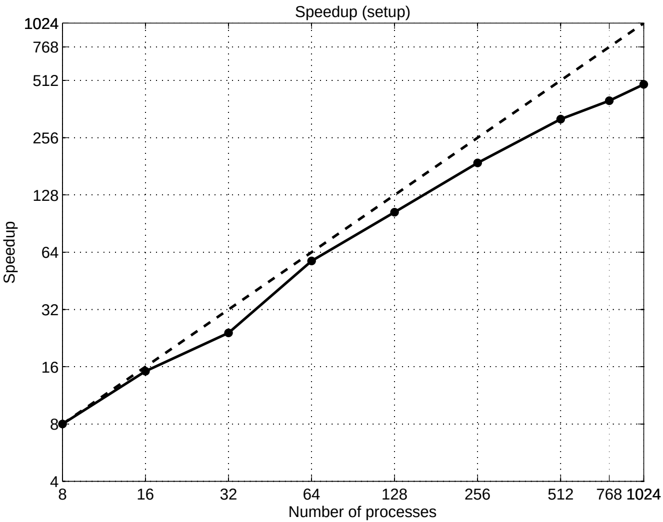
RCS of a Sphere

- 89.8λ diameter
- 7.6 millions of unknowns (7,651,221)
- Multiple runs from 8 to 1024 processes

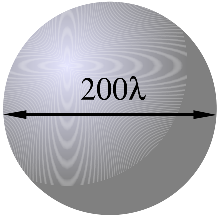
Scalability. Matrix Vector Product time



Scalability. Setup time



More than 30 millions of unknowns



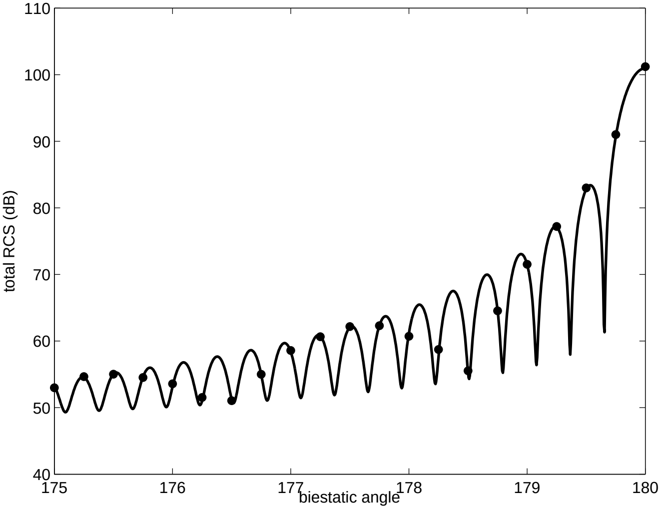
RCS of a Sphere

- 200λ diameter
- 32 millions of unknowns (32,411,106)
- Multiple runs from 8 to 1024 processes

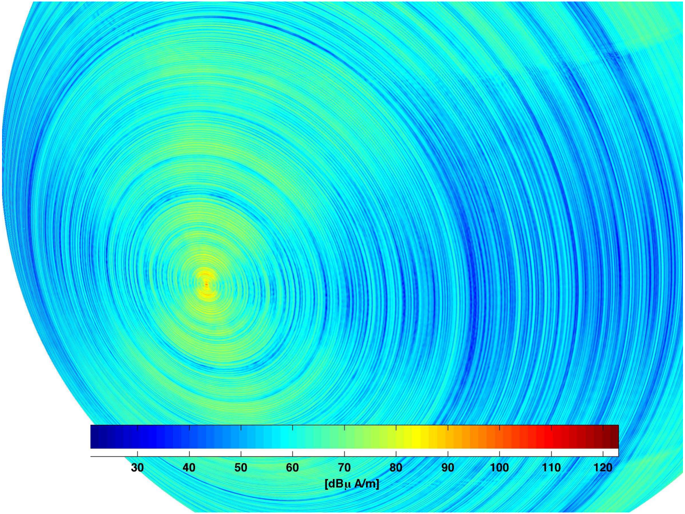
Technical data

- 512 process
- 7TB of total memory
- Setup time: 4h35m
- Time for each MVP: 6m6s
- TOTAL Time: 15h10m

Results: Bistatic RCS of the Sphere



Results: Currents the Sphere



Outline

Conclusions

Near World Record in Electromagnetics

- Only one week in Finis Terrae. Best Time-to-Solution than any other record
- Memory/CPU ratio of Finis Terrae: Solution to problems irresolvable by other supercomputers with more CPU's
- Scalability: Relegated single Level FMM is very attractive for high performance scientific challenges.

Is possible more than a hundred of millions?

- Ergul: Objective for the next years
- Extremadura, Vigo and CESGA: Several improvements to achieve a great record in 2008 or 2009.