Parallelizing NEC's Equation Solver Algorithm with OpenMP

Mario Trangoni[1] **Victor Rosales**[2]

Argentina Software Design Center (Intel)

[1]mario.trangoni@intel.com

[2]victor.h.rosales@intel.com

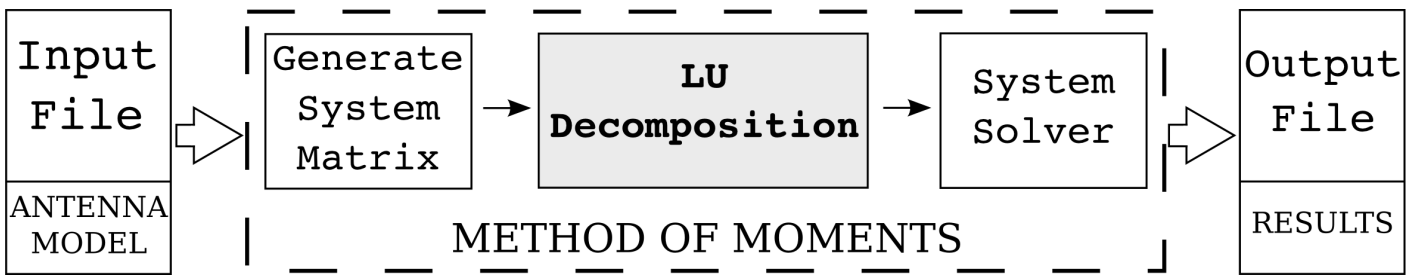
HPCLatAm 2012
Buenos Aires, Argentina
July 24th, 2012

Agenda

Agenda

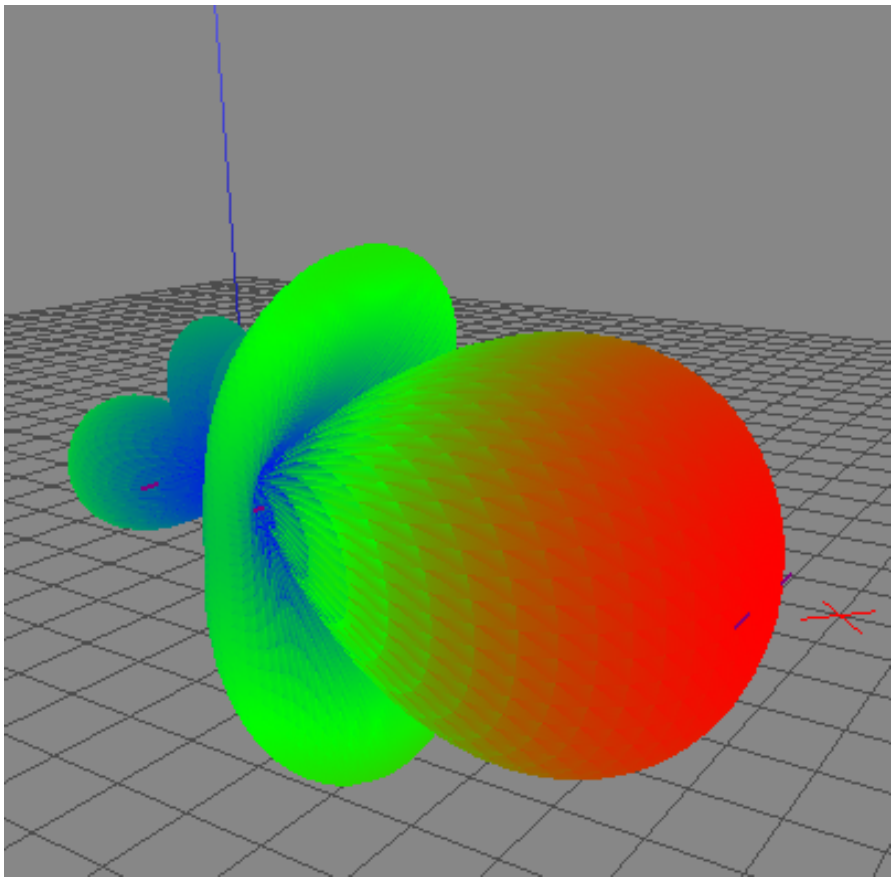
Introduction I

- NEC was written in FORTRAN in the 1970s and translated from FORTRAN to C in 2003 renamed as *nec2c*.
- It uses the method of moments solution of the electric field integral equation for thin wires and the magnetic field integral equation for closed, conducting surfaces.
- Models are defined as elements of wire or similar as an input text file. These models are then input into the NEC application.

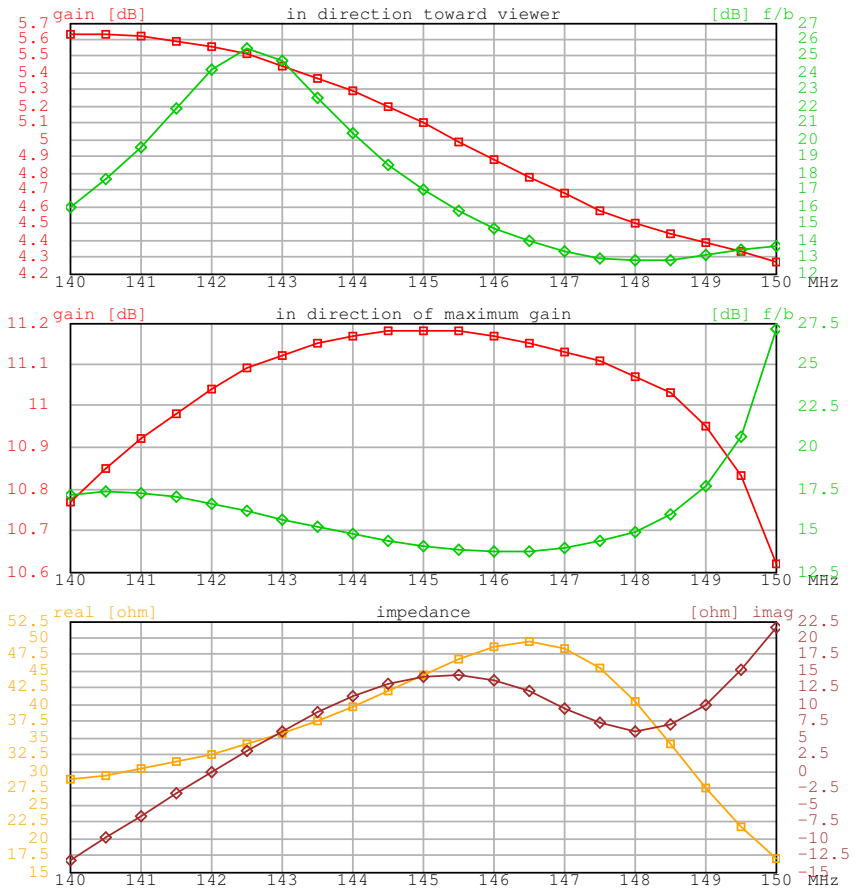


Introduction II

- Results can then be input into subsequent 'helper' applications for visual viewing and the generation of other graphical representations.



qantenna 0.2

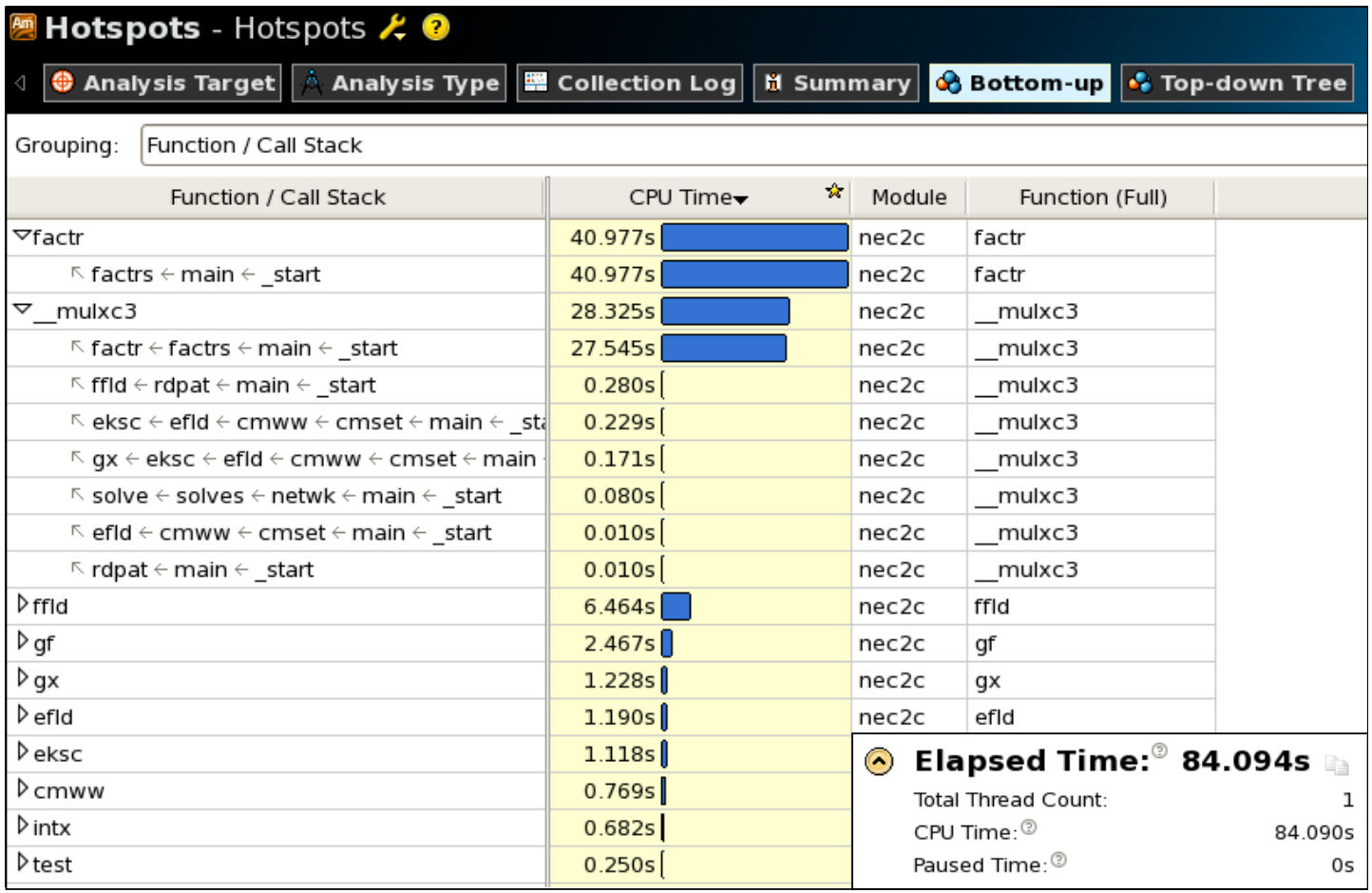


xnecview 1.35

Agenda

Discovering Bottlenecks

- Intel® VTune Amplifier was used to detect the hot spots.
- nec2c spends most of the execution time in *factr* (LU factorization)
- *mulxc3* does complex numbers multiplication and is invoked mostly by factr.



Agenda

Original Implementation

- Works one row at a time from top to bottom

Row 0

$$\begin{pmatrix} \boxed{a_{11}} & \boxed{a_{12}} & \boxed{a_{13}} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Division: $a_{12} = \frac{a_{12}}{a_{11}}$; $a_{13} = \frac{a_{13}}{a_{11}}$

Row 1

$$\begin{pmatrix} a_{11} & \boxed{a_{12}} & \boxed{a_{13}} \\ \boxed{a_{21}} & \boxed{a_{22}} & \boxed{a_{23}} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Subtraction: $a_{22} = a_{22} - a_{12} \cdot a_{21}$; $a_{23} = a_{23} - a_{13} \cdot a_{21}$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & \boxed{a_{22}} & \boxed{a_{23}} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Division: $a_{23} = \frac{a_{23}}{a_{22}}$

Row 2

$$\begin{pmatrix} a_{11} & \boxed{a_{12}} & \boxed{a_{13}} \\ a_{21} & \boxed{a_{22}} & \boxed{a_{23}} \\ \boxed{a_{31}} & \boxed{a_{32}} & \boxed{a_{33}} \end{pmatrix}$$

Subtraction: $a_{32} = a_{32} - a_{12} \cdot a_{31}$; $a_{33} = a_{33} - a_{13} \cdot a_{31}$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & \boxed{a_{23}} \\ a_{31} & \boxed{a_{32}} & \boxed{a_{33}} \end{pmatrix}$$

Subtraction: $a_{33} = a_{33} - a_{23} \cdot a_{32}$

Agenda

Parallelization Proposal

Matrix (3×3):

$$\begin{pmatrix} \boxed{a_{11}} & \boxed{a_{12}} & \boxed{a_{13}} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Division: $a_{12} = \frac{a_{12}}{a_{11}}$; $a_{13} = \frac{a_{13}}{a_{11}}$

$$\begin{pmatrix} a_{11} & \boxed{a_{12}} & \boxed{a_{13}} \\ \boxed{a_{21}} & \boxed{a_{22}} & \boxed{a_{23}} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Subtraction: $a_{22} = a_{22} - a_{12} \cdot a_{21}$;
$a_{23} = a_{23} - a_{13} \cdot a_{21}$

$$\begin{pmatrix} a_{11} & \boxed{a_{12}} & \boxed{a_{13}} \\ a_{21} & a_{22} & a_{23} \\ \boxed{a_{31}} & \boxed{a_{32}} & \boxed{a_{33}} \end{pmatrix}$$

Subtraction: $a_{32} = a_{32} - a_{12} \cdot a_{31}$;

$a_{33} = a_{33} - a_{13} \cdot a_{31}$

Matrix (2×2):

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & \boxed{a_{22}} & \boxed{a_{23}} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Division: $a_{23} = \frac{a_{23}}{a_{22}}$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & \boxed{a_{23}} \\ a_{31} & \boxed{a_{32}} & \boxed{a_{33}} \end{pmatrix}$$

Subtraction: $a_{33} = a_{33} - a_{23} \cdot a_{32}$

- First step operates on the full matrix and the subsequent steps will successively reduce the size of the operated matrix.

Agenda

Implementation I

Maximum Search

- An implementation with *Maximum Search*, *Row Exchange and Division by Row's Diagonal Element* and *Linear Combinations* subsections was executed and compared against an implementation including only the last two subsections.

Number of Threads	Implementation 1 (*sec.*)	Implementation 2 (*sec.*)	Difference (Imp.1 – Imp.2)
2	906.71	870.15	4.20 %
4	531.62	516.49	2.93 %
8	344.77	343.93	0,24 %

Comparison of performance over a matrix of 5000 x 5000 elements.

- *Maximum Search* parallelization not implemented.

Implementation II

Row Exchange and Division by Rows Diagonal Element

- Exchange of rows only if the maximum is not located on the diagonal.
- Avoid division by zero and reduce the precision error.
- Division by the diagonal element or first element of the row.

```
# pragma omp parallel shared (a,...) private(i,...)
   { /*Begin parallel zone*/
if ((ip[r]!=r+1) && (r!=n-1)) {
    i=ip[r]-1; /*if max not at diagonal, do interchange */
# pragma omp for
    for (j=r; j<n; j++) {   /*Row's interchange */
       arj=a[i+j*ndim];
       a[i+j*ndim]=a[r+j*ndim];
       a[r+j*ndim]=arj;
    }
   }

# pragma omp for
    for (i=r+1; i<n; i++) {  /*n-1 iterations */
       a[i+rndim]=a[i+rndim]/a[r+rndim];
   }
```

Implementation III

Linear Combinations

- Simply parallelized with an OpenMP pragma.
- Parallelization's maximum gain zone.

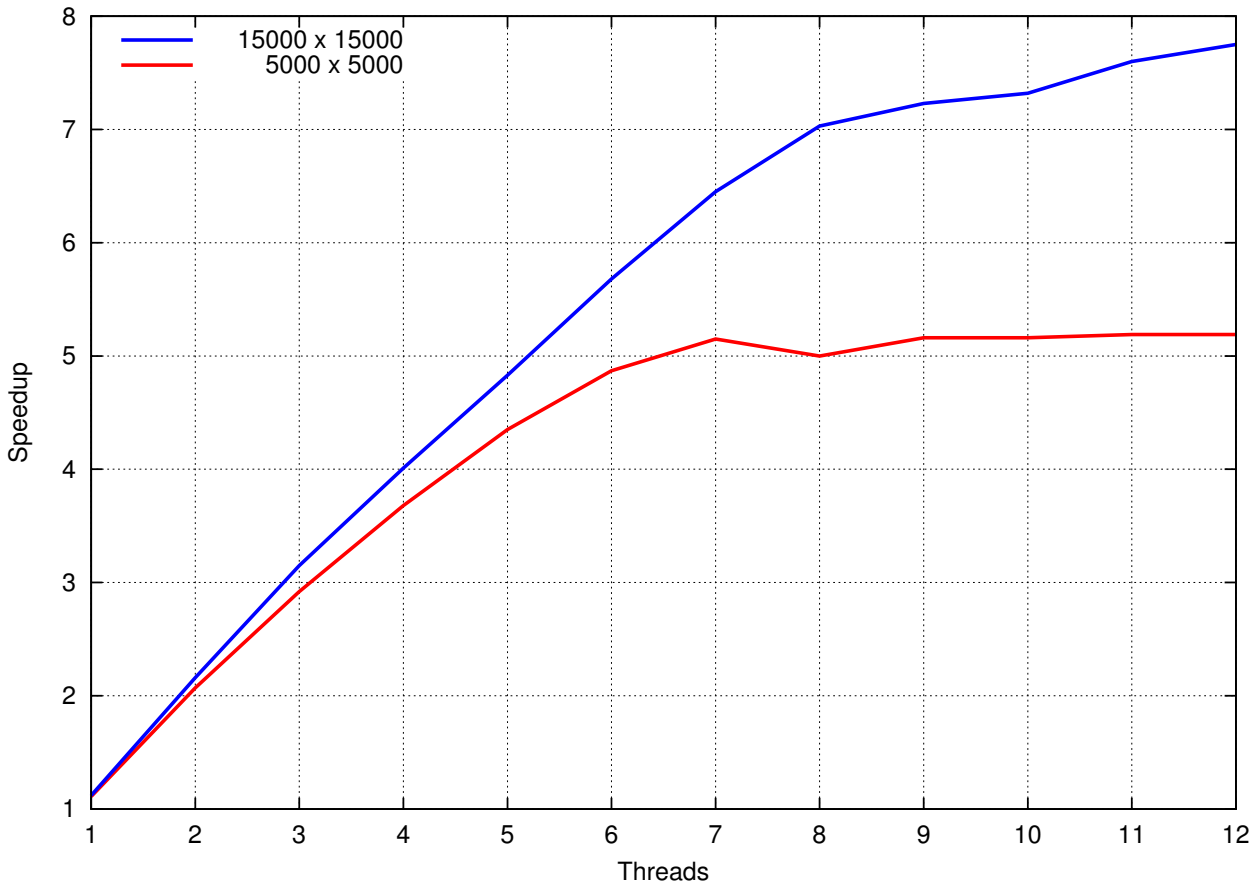
```
# pragma omp for
   for (j=r+1; j<n; j++) {
      /*n-1 iterations - rows */
      for (i=r+1; i<n; i++) {
         /*n-1 iterations - elements */
         a[i+j*ndim]-=(a[i+rndim]*a[r+j*ndim]);
      }
   }
  } /*Finish Parallel zone*/
```

Agenda

Results

Scalability over the number of OpenMP threads

- GCC 4.4.4 - Flags: `-O2 -march=core2 -mtune=generic -fopenmp`
- Ran on: Intel Westmere-EP (2 x Intel Xeon X5670 (6 cores) @ 2.93 Ghz / 12 MB shared Last-Level Cache / 12Gb RAM DDR3-1066 (8533,33 MB/seg))
- OS: CentOS release 5.4 (Final)



Agenda

References

Roger F. Harrington, *Field Computation by Moment Methods*, IEEE Press Series on Electromagnetic Wave Theory, Wiley-IEEE, ISBN 0-78-031014-4, 1993.

F. Gisin, *Using the method of moment NEC code to solve EMC problems*, IEEE International Symposium on Electromagnetic Compatibility,Dallas, TX, USA, 9-13 August 1993.

H. Sormann, *Numerische Methoden in der Physik*, Institut für Theoretische Physik - Computational Physics, TU Graz, Graz, Austria, 2011.

G. J. Burke, A. J. Poggio, *Numerical Electromagnetics Code* (NEC-2), Public Domain Code, 1980.

Thanks!