SOLVING MAXWELL'S EQUATIONS WITH A FINITE DIFFERENCE TIME DOMAIN METHOD ON A CLUSTER OF WORKSTATIONS IN AN INDUSTRIAL ENVIRONMENT

Anders Ålund

Swedish Institute of Applied Mathematics
Chalmers Teknikpark
S-412 63 Göteborg, Sweden
email: anders@itm.se

and

Ulf Thibblin

Saab Military Aircraft
S-581 88 Linköping, Sweden
email: ulfth@weald.air.saab.se

The ITM Parallel Project

- Three year project aimed at porting industrial applications to run on parallel environment.
- Sponsored by NUTEK, Swedish National Board for Industrial and Technical Development
- Participating companies and problem areas
  - Saab Military Aircraft
    - CFD
    - Electromagnetics
  - Volvo Aero Corporation
    - CFD
  - ABB Corporate Research
    - Electromagnetics
    - Structural mechanics
    - Linear Algebra

Overview

- Introduction
- Electromagnetic problems in the aircraft industry
- Computational Electromagnetics (CEM)
- Maxwell's Equations
- FDTD code
- PVM implementation
- Test case
- Results
- Theoretical speedup analysis
- Conclusions
- Future

Typical EM problems in aircraft industry

- Aircraft should withstand electromagnetic threats such as
  - Lightning
  - Radio and radar waves
  - Nuclear electromagnetic pulses (EMP)
- Antenna design
- Radar Cross Section (Stealth technology)

Computational methods

- Finite Difference Time Domain (FDTD)
- Finite Difference Frequency Domain (FDFD)
- Finite Volume Time Domain (FVTD)
- Finite Element Method (FEM)

FDTD method

- Interested in both pulse exitation and frequency responses
- Typical problem size: $10^6 - 10^7$ grid cells
  6 variables / cell
- Typical solution time: several hours on Cray Y-MP or days on a SGI Indigo 4400
- Typical memory req.: 50-1000 MByte

Maxwell's equations

Maxwell's equations for linear media read

$$\mu \frac{\partial H}{\partial t} + \nabla \times E = 0$$

$$\epsilon \frac{\partial E}{\partial t} - \nabla \times H = -J$$

where

E is the electric field,

H the magnetic field,

J the source current density,

μ the magnetic permeability and

ϵ the electric permittivity.

PVM at Saab Military Aircraft

Objectives

- Find cheaper solutions to replace the costly and heavily loaded Cray
- Utilize unused capacity on workstations
- Reduce turn-around time for computations
- Solve larger problems

Computer resources for CEM

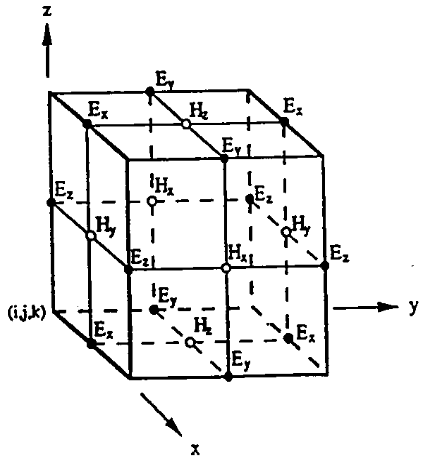
- 5 SGI Indigo/R4000 (10-15 MFLOPS)
- Share in CRAY Y-MP/464

Computer resources at other departments

- Approx 100 SGI Indigo/R4000 (10-15 MFLOPS)
- Approx 50 HP9000/712-715 (10-15 MFLOPS)

FDTD Discretization

The spatial and time derivatives are approximated with central differences on a uniform Cartesian grid. The algorithm uses a "leapfrog" scheme in time and space applied on a staggered grid, both in time and space.



Implementation

- Cartesian grid presents a natural way of splitting the flow problem by domain decomposition and a host-node programming model
- Host process
  - Handles Input/Output
  - Distributes sub-problems to node processes
  - Synchronizes nodes
  - Controls convergence criteria
  - Saves solutions
- Node process
  - Integrates equations
  - Communicates interface information between nodes
  - Sends its solution to the host

Implementation took approx. 3 man weeks

Output from FDTD-code

- Surface currents on the inside and outside surface of the whole model for a given number of frequences.
- Surface currents on given surfaces at given times.
- Surface currents in given grid points for all timesteps.
- Full solutions.

FDTD main loop

The main iteration loop of the FDTD code looks like:

Repeat

Update radiating boundary conditions

Update all inner magnetic field variables using local electric field variables plus local electric ghost-variables.

Apply magnetic source terms corresponding to the exitation.

Communicate all updated magnetic field variables to the corresponding magnetic field ghost-variables on the neighboring processors.

Update all inner electric field variables using local magnetic field variables plus local magnetic ghost-variables.

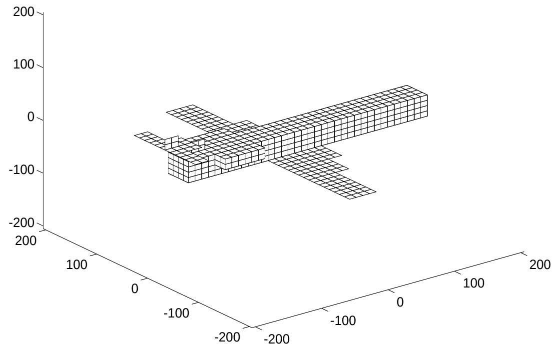
Apply electric source terms corresponding to the exitation.

Communicate all updated electric field variables to the corresponding electric field ghost-variables on the neighboring processors.

Update scattering object boundary conditions

Until(done)

Generic aircraft model

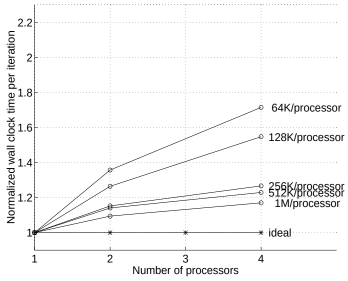


FDTD - Results

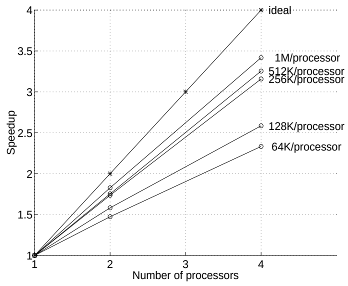
Problem-size (cells)	Number of procs	Machines SGI	HP715	Cray Y-MP
64K	1	0.25	0.28	
128K	1	0.48	0.53	
	2	0.34	0.38	
256K	1	0.99	1.05	
	2	0.62	0.67	
	4	0.56	0.48	
512K	1	1.66	1.92	0.13
	2	1.15	1.21	
	4	1.01	0.82	
1M	1	3.26	3.82	0.24
	2	1.84	2.19	
	4	1.71	1.33	
2M	1	6.78	7.64	0.45
	2	3.74	4.18	
	4	2.65	2.36	
4M	2	7.12		0.88
	4	5.03	4.47	
8M	4	9.63		

Elapsed time in seconds/iteration

FDTD - Fix problem size per processor

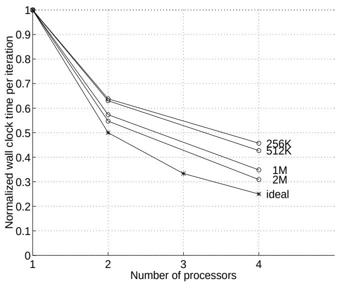


Normalized time / iteration

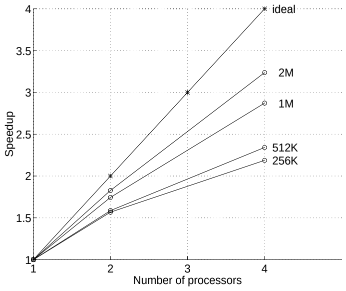


Speedup

FDTD - Fix total problem size



Normalized time / iteration



Speedup

Analysis of the work and data transmission

With

t_{EH}	Time to update E- and H-fields
t_{RBC}	Time to update radiating BC
t_{OBC}	Time to update scattering object BC
t_{SRC}	Time to update source terms
t_{COM}	Time to communicate interface variables

we get

$$t_{EH} \sim n_x n_y n_z$$
$$t_{RBC} \sim n_x n_y + n_y n_z + n_z n_x$$
$$t_{OBC} \approx 0$$
$$t_{SRC} \sim n_x n_y + n_y n_z + n_z n_x$$

Time to solve the problem on one processor is

$$T(1) = K_3 n_x n_y n_z + 2K_2(n_x n_y + n_y n_z + n_z n_x)$$

and on p processors

$$T(p) = \frac{T(1)}{p} + t_{COM}$$

Analysis (cont'd)

Time to solve the problem on one processor is

$$T(1) \;=\; K_3 n_x n_y n_z + 2K_2(n_x n_y + n_y n_z + n_z n_x)$$

and on p processors

$$T(p) \;=\; \frac{T(1)}{p} + t_{COM}$$

Assuming $n_x = n_y = n_z = n$ gives

$$T(p) = \frac{K_3 n^3}{p} + 48\beta(p^{1/3} - 1)n^2.$$

The speedup is

$$S(p) = T(1)/T(p) = \frac{p}{1 + \frac{48\beta p(p^{1/3}-1)}{k_3 n}}.$$

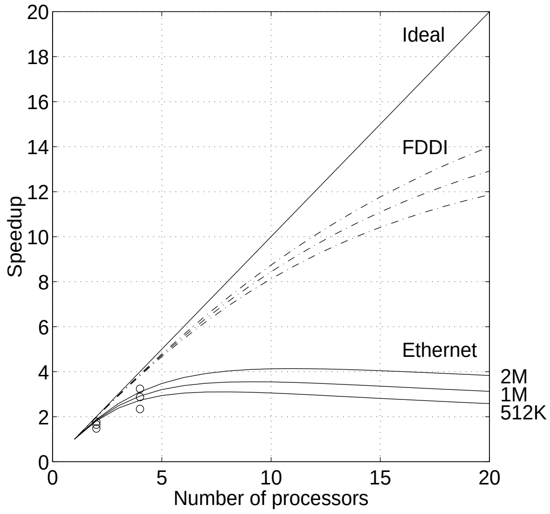
and the optimal number of processors, given by minimizing the time $T(p)$ is

$$p_{opt} \;=\; \left(\frac{K_3 n}{16\beta}\right)^{3/4}$$

Conclusions

- Implementation of a parallel FDTD code using PVM is straight forward
- Poor balance between computations and communication for small problems
- On shared ethernet: Optimum 4 WS
- Would scale better on switched communication networks

FDTD - Theoretical speedup limits



Theoretical speedup of the parallel FDTD code on HP715/50 for different problem sizes and communication networks

Future

Implement

- Checkpoint restart mechanism
- Submodels (wires etc)
- Non-perfect conductors

Run on parallel, distributed memory, computers

- Convex SPP1
- IBM SP2
- Parsytec CG/POWER PLUS

Will look at

- Higher order methods
- Scheduler / Queuing system