3D Full Wave EM Simulations
Accelerated by GPU Computing

Fabrizio Zanella
Systems Manager
CST of America
Abstract

3D Full Wave Electromagnetic simulations of RF components, antennas, printed circuit boards, can be quite time consuming. The CST2010 tool suite includes the capability to activate GPU Computing. Examples will be shown of using the Nvidia Tesla C1060, M1060 and S1070 configurations to provide significant performance improvements for complex simulations.
CST 3D Full Wave Complex Simulations
CST Hardware Based Acceleration Techniques Improve Performance

- Multithreading
- GPU Computing
- Distributed Computing
- MPI Computing

GTC 2010 | Sept 20-23, 2010
San Jose Convention Center, San Jose, California
CST GPU Computing Hardware
CST MWS Transient Solver

Available for OS: 🍀 🦉 Licensing: Token Scheme

Nvidia GPU hardware supported by CST21010; NVIDIA Tesla 10 series

- Quadro FX 5800/Tesla C1060 (1 GPU)
- Quadro Plex 2200 D2 (2 GPUs)
- Tesla S1070 (4 GPUs)

- 240 Cores per GPU
- 4 GB GDDR3 memory per GPU
- Memory Bandwidth 102 GB/s (normal DDR3 RAM: 25.6 GB/s)

As the Transient Solver performance is heavily influenced by the memory bandwidth it benefits a lot from the high bandwidth to the GPU memory.
Supported Configurations

Windows XP x64
Windows Vista x64
Windows 7 x64
Windows Server 2003 R2
Windows Server 2008 R2
RHEL 4, 5
CentOS 4,5

Workstations:  HPZ800, Dell T7500, etc
Servers:  1U, 2U SuperMicro GP-GPU servers
NVIDIA Tesla 20 series, or Fermi supported in CST2011

- 512 Cores per GPU
- 3 GB GDDR5 (C2050), 6 GB GDDR5 (C2070) memory per GPU
- First GPU with Error Protecting Code (ECC)
- Floating Point Performance strongly improved by factor of about 8
- C2050, C2070 supported in CST2011 release, Jan 2011
Hardware Configurations used for benchmarks

2U SuperMicro, Windows Server 2008 R2
(8) E5530, 72GB DDR3 RAM
Connected to NVIDIA Tesla S1070 (4xM1060)

1U SuperMicro server, CentOS v5.5
X8DAH+-F motherboard, 2xM1060 GPU cards
(8) E5630, 48GB DDR3 RAM
DC+GPU Computing used
GPU Benchmark: Disk Drive Interconnect

Model = 25 M HEX cells; lossy metal and dielectrics
GPU=2xM1060
10X speedup over 8 Xeon X5550 cores
GPU Benchmark: Multilayer Package Model

8xE5630 cores, 4 ports total time: 5.9 hours
1xM1060 4 ports total time: 1 hour
5.9x performance improvement
GPU Benchmark: PCB Model

8xE5630 cores; 4 ports total time: 23h,15m
1xM1060; 4 ports total time: 3h,45m
6.2x performance improvement
GPU Benchmark: PCB-Cable-PCB

8xE5630 cores total time: 17 hours
1xM1060 total time: 2 hours
8.5x performance improvement
GPU Benchmark: Package on PCB

100M Mesh Cells
CPU: 2x Intel Xeon E5530, 72 GB RAM
GPU: NVIDIA Tesla S1070 (4xC1060)

<table>
<thead>
<tr>
<th>Test Case</th>
<th># Nodes</th>
<th># Cores per Node</th>
<th># GPUs per Node</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure CPU</td>
<td>1</td>
<td>2x4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>GPU</td>
<td>1</td>
<td>2x4</td>
<td>4</td>
<td>16x</td>
</tr>
</tbody>
</table>
CST MPI Cluster Computing

Matrix calculation and transient solution distributed
Memory & CPU balancing

Complex model split into sub-structures (domains) for solution on a cluster
As the capacity of the GPU hardware is limited. Combined MPI and GPU Computing provides a possibility to combine GPU accelerated nodes to form an MPI cluster. This helps to increase the maximum model size.

Each MPI node can be accelerated by GPU hardware.
### PCB and Package- 100M cells

<table>
<thead>
<tr>
<th>Test Case</th>
<th># Nodes</th>
<th># Cores per Node</th>
<th># GPUs per Node</th>
<th>Total # GPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure CPU</td>
<td>1</td>
<td>2x4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MPI</td>
<td>8</td>
<td>2x4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GPU</td>
<td>1</td>
<td>2x4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>MPI+GPU</td>
<td>4</td>
<td>2x4</td>
<td>2</td>
<td>8</td>
</tr>
</tbody>
</table>

**Speedup**

![Speedup Chart](chart.png)

**MPI+GPU Computing Performance**

- PCB and Package- 100M cells
- Test Case: Pure CPU, MPI, GPU, MPI+GPU
- # Nodes: 1, 8, 1, 4
- # Cores per Node: 2x4, 2x4, 2x4, 2x4
- # GPUs per Node: 0, 0, 4, 2
- Total # GPUs: 0, 0, 4, 8

---

**Notes:**

- Pure CPU: No GPUs used
- MPI: 0 GPUs used
- GPU: 4 GPUs used
- MPI+GPU: 2 GPUs used

---

**Additional Information:**

- www.cst.com
- 16
- 0
- 5
- 10
- 15
- 20
- 25

---

**Stress Test:**

- Full Load: 100M cells
- 100% Utilization
- Performance Optimization

---

**Conclusion:**

- MPI+GPU offers significant speedup compared to other configurations
- Efficient resource allocation for high-performance computing
CST GPU Computing
- Conclusion -

- CST GPU Computing provides a significant performance improvement for complex simulations.

- The maximum model size (number of mesh cells) which can be handled by GPU Computing is limited by the amount of memory available on the GPU hardware (currently max. 4 GB).

- For larger models multiple GPU cards can be used or combined MPI and GPU Computing