

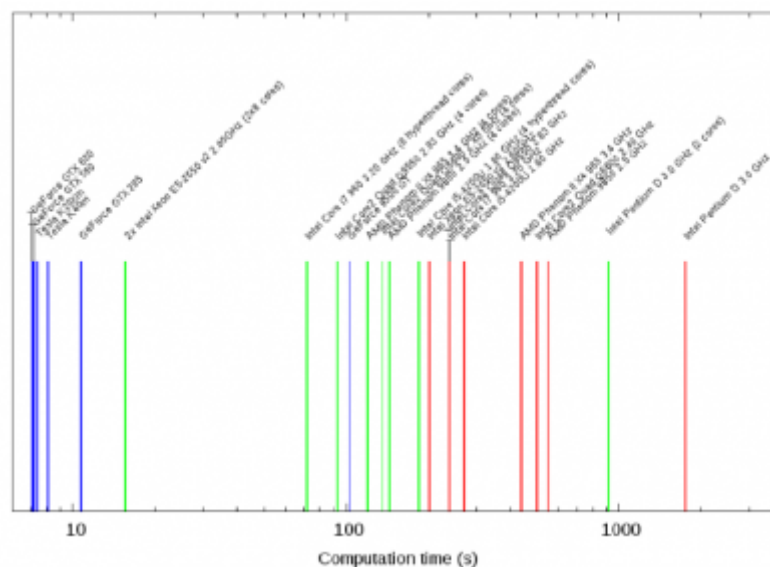
Table of Contents

Graphics cards use 1

Graphics cards use

FDTD method is computationally limited by dense discretization that is necessary for its proper operation. For the standard Yee's algorithm (that is a core of FDTD) the stable discretization requires a spatial minimum of $\Delta x \leq \lambda/10$, where λ is propagating electromagnetic wave wavelength. There are some approaches in the literature to overcome these limitations, which however lead to much more complicated algorithms that cannot be simply adapted to all of the large spectrum of FDTD calculations.

As a result, both the capacity of computer memory and the processing speed is a limiting factor for large and long calculations in FDTD. This fact decreases the applicability of FDTD for many purposes. An alternative approach, that is used in Gsvit, is based on using a graphics processing unit (graphics card) to move the calculation from the computer processor (CPU) to the graphics processing unit (GPU) and to speed it up significantly, as shown in the following graph where the time necessary for the same calculation on different cards and processors is listed (note that the x axis is logarithmic).



accessible by single processor. All the memories are hardware limited (for each type of GPU differently). We refer to Nvidia CUDA developer zone for further details.

To check if the graphics card installed on your computer is suitable for GSvit calculations and if GSvit was installed with graphics card support at all, you can run the solver with parameter "test 0", e.g. on Linux system:

```
klapetek@pejsek:~/rungsvit/bin> ./gsvit test 0
Running GSvit tests at level 0
System has 32 cores
Gsvit is installed at /home/klapetek/rungsvit/bin
Program is compiled with GPU support
Searching for available GPUs...
Found 4 GPUs
The Properties of the Device with ID 0 are
Device Name           : Tesla K40m
Device Memory Size    : 3489202176
Block Shared memory size: 49152
Max grid size         : 2147483647x65535x65535
Max threads dim       : 1024x1024x64
The Properties of the Device with ID 1 are
Device Name           : Tesla K20Xm
Device Memory Size    : 1744371712
Block Shared memory size: 49152
Max grid size         : 2147483647x65535x65535
Max threads dim       : 1024x1024x64
The Properties of the Device with ID 2 are
Device Name           : Tesla K20Xm
Device Memory Size    : 1744371712
Block Shared memory size: 49152
Max grid size         : 2147483647x65535x65535
Max threads dim       : 1024x1024x64
The Properties of the Device with ID 3 are
Device Name           : Tesla K20Xm
Device Memory Size    : 1744371712
Block Shared memory size: 49152
Max grid size         : 2147483647x65535x65535
Max threads dim       : 1024x1024x64
```

This means that you can use four different GPUs on this system (which does not usually happen on a standard computer). GSvit does not support use of all of them at once, but you can still run up to four different instances on different cards. The basic settings for using the GPU are therefore whether to use it at all (GPU directive in the parameter file) and eventually which one to use (UGPU directive in the parameter file). GPUs are numbered from 0, exactly the same way as what `<tt>gsvit test 0</tt>` outputs, so these settings in the parameter file:

`<html> GPU 1`

`UGPU 0 <html>`

will run the calculation on the Tesla K40m card on our system.

From:

<http://www.gsvit.net/wiki/> - **GSvit documentation**

Permanent link:

http://www.gsvit.net/wiki/doku.php/opt:graphics_cards?rev=1535698580



Last update: **2018/08/31 08:56**