PERFORMANCES OF VORPAL-GPU SLAB-SYMMETRIC DLW*

F. Lemery[1], P. Mullowney[2], P. Stoltz[2], K. Duffin[3], N. Karonis[3], D. Mihalcea[1],P. Piot[1,4], J. Winans[3]
[1] Department of Physics and Northern Illinois Center for Accelerator &
Detector Development, Northern Illinois University DeKalb, IL 60115, USA
[2] Tech-X Corporation, Boulder, CO 80303, USA
[3] Department of Computer Science, Northern Illinois University DeKalb, IL, 60115, USA
[4] Accelerator Physics Center, Fermi National Accelerator Laboratory, Batavia, IL 60510, USA

Abstract

GPU-based computing has gained popularity in recent years due to its growing software support and greater processing capabilities than its CPU counterpart. GPU computing was recently added in the finite-difference time-domain program VORPAL. In this paper we carry electromagnetic simulations through a slab-symmetric dielectric-lined waveguide (DLW). We use this simulation model to explore the scaling of the GPU version of VORPAL on a new TOP1000-grade hybrid GPU/CPU computer cluster available at Northern Illinois University (NIU).

INTRODUCTION

The development of computational power in the past several decades has led to our capability to solve complicated problems numerically, which would otherwise be very difficult or impossible.

The rapid pace at which processor architecture evolves must be matched with improvements in program/software design to take advantage of the full capability of a processing unit. The most recent development include the deployment of hardware with large numbers of Graphical Processing Units (GPU). GPU's and their highly parallel structure makes them more effective than general-purpose Central Processing Units (CPUs) for algorithms where processing of large blocks of data is done in parallel.

Although a CPU is faster than a GPU on a per-core basis, each GPU may contain several thounsand cores [1] while each CPU is presently limited to 16 cores [2]. Moreover, GPUs are capable of processing many parallel streams of data simultaneously which makes them highly efficient for parallel computation. Finally GPUs are capable of performing vector operations and double-precision floating-point numbers, which makes them useful in electromagnetic simulations such as VORPAL [3].

VORPAL is a three-dimensional electromagnetic and electrostatic PIC code. VORPAL uses a conformal finite difference-time domain (FDTD) method to solve Maxwell's equations and that includes an advanced technique known as cut-cell boundaries to allow accurate representation of curved geometries within a rectangular grid.

To test the scalability of VORPAL-GPU on a recently acquired TOP1000-grade hybrid GPU/CPU, "GAEA," we use a familiar problem [4] on collinear beam driven wakefield acceleration. The GAEA computing cluster at NIU includes 60 compute nodes with 3 additional nodes to handle disk and terminal services. Each compute node contains 2 hex core CPUs (Intel Xeon X5650 opearting at 2.67 GHz), 72 GB of RAM, 2 TB of local storage and 2 GPU cards. Each GPU card is an NVIDIA TESLA M2070 capable of 515 Gigaflops of double precision floating point arithmetic, 1030 Gigaflops of single precision floating point arithemetic, each GPU card has 448 CUDA cores, with access to 6 GB of GDDR5 memory, and 150 GB/s memory bandwidth. The cluster also possesses approximately 192 TB of disk storage accessible to all compute nodes. Lastly, the compute nodes and storage nodes are interconnected with a 40 Gb/s QDR INFINIBAND network.

PROBLEM SETUP

We explored the scaling of a dielectric wakefield accelerator (DWFA) simulation using VORPAL-GPU. In DWFA's a drive bunch is used to excite an electromagnetic wake in a dielectric-lined waveguide (DLW). The trailing wake can be used to accelerate a properly timed witness bunch. We focus on the slab-symmetric DLW based on [5]. The two types of modes in this structure are longitudinal section electric (LSE) and longitudinal section magnetic (LSM). The latter, fundamental is the accelerating mode.

Currently VORPAL-GPU does not support particle-in-cell functionalities, instead the electron bunch is modeled by the time-dependent current density

$$\vec{J}(x,y,z,t) = \frac{cQ\,(2\pi)^{-\frac{3}{2}}}{\sigma_x\sigma_y\sigma_z} e^{\left(-\frac{z^2}{2\sigma_z^2} - \frac{y^2}{2\sigma_y^2} - \frac{(x-ct)^2}{2\sigma_x}\right)}\hat{x}, \quad (1)$$

where the charge $Q = 1$ nC, the transverse sizes are $\sigma_y = \sigma_z = 30\ \mu m$, the longitudinal size is $\sigma_x = 100\ \mu m$ and c is the speed of light.

The computational domain associated to the problem appears in Fig. 1. The current distribution enters the computation domain from $(x,t) = (0,0)$ and propagates through

* This work was supported through DOE award # DE-SC0005135 and DTRA award # HDTRA1-10-1-0051 to Northern Illinois University

the structure until it exits on the $x > 0$ side. We implement perfectly conducting boundaries (PCB) on the y and z sides to mimic the conducting plates surrounding the DLW. In addition, perfectly matched layers (PMLs) are used on both ends (x) of the structure to mimic an open boundary and avoid spurious reflection of the electromagnetic field generated by the current distribution.

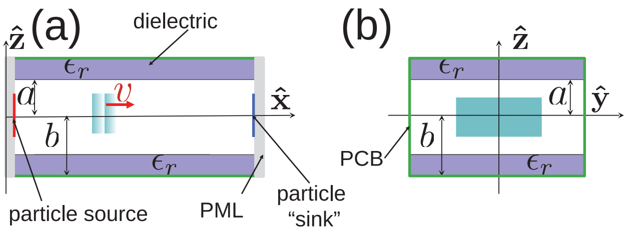

Figure 1: A diagram of the slab seen from the y-transverse direction (a) and the from the x-longitdinal direction (b). The slab is composed of a dielectric coating with dielectric permitivity ϵ_r surrounded by a PCB. PMLs are implemented at $x = 0$ and $x = 2\ cm$ prevent any reflections in the dlw.

In this problem, Maxwell's equation were solved using VORPAL's `FieldCombo` algorithm and because of the dielectric medium the updated included the construction and update of the ϵ permitivity matrix necessary to define and update the $\vec{D}$ electric-displacement field components.

SCALING STUDIES

In high-performance computing, there are generally two figure of merits used to describe scaling performance: the strong and weak scalings. Strong scaling refers to how the solution time changes for a parallel computation with a fixed computational volume. Weak scaling refers to how the solution time changes for a fixed parallel computation with different computational volumes. With respect to VORPAL-GPU, we are interested in knowing the cost of the solution time per processor(s) for an increase in resolution (finer grid).

A larger number of processors will not necessarily decrease the solution time; instead this relationship depends on many factors and is problem specific. Examples of such factors include, e.g., (i) increase in time to split the jobs to more cores (ii) increase in time to send information to more cores (iii) increase in time to allocate less memory to more cores (iv) and the finite bandwidth available to communicate between the cores.

In VORPAL-GPU we use a cartesian grid (N_X, N_Y, N_Z) to describe number of cells in $\hat{x}, \hat{y}, \hat{z}$ directions (see Tab. 1). To improve the scaling capability, it was important to choose the decomposition of the grid wisely.

The simulation was designed such that the electron bunch propagation direction was in the long, longitudinal direction of the simulation ($\hat{x}$). This was a convenient choice which enabled several clear benefits. First it allows for a simple one dimensional decomposition of the gridded field quantities. This leads to a simple messaging pattern of the quantities in the halo–that is each GPU sends and receives data only from the other GPUs on each side of the decomposition This choice leads to a minimization of the messaging costs leading to optimal efficiency. Moreover, VORPAL-GPU is designed such that the x-dimension represents the slowest dimension in the memory layout of the field quantities while the z-dimension is the fastest. This is a very important choice which enables contiguous chunks of memory in the planes perpendicular to the x-axis (i.e. the halo guard cell planes) to be transferred back to the host in a single memory transfer across the PCI Express bus. This yields the optimal performance as it enables one to effectively overlap communication costs with the update of the main "body" region on each GPU.

However, one must also be careful to choose the dimensions of the simulation domain carefully. Even a simple one-dimensional domain decomposition can perform poorly if the ratio of the cells needing to be messaged to the cells needing to be updated in the body region is too large.

Because we decompose in the $\hat{x}$ direction, our message volume will be $2N_Y N_Z$. For a problem involving N GPUs, our update volume will be $\frac{N_X N_Y N_Z}{N}$. When $\frac{2N_Y N_Z}{(N_X N_Y N_Z)/N}$ becomes large (i.e. greater than .02 we expect the performance to degrade substantially. In these situations, the amount of work in the body update region will be too small to effectively hide the communication costs. In particular, the body update region kernels will simply run too fast to hide these costs.

Table 1: Cartesian Grid Dimensions and Volumes used in this Scaling Study

Size	NX	NY	NZ	Volume
S	602	128	128	9863168
M	1552	256	256	101711872
L	7752	256	256	508035072
XL	3852	512	512	1009778688
XXL	7702	512	512	2019033088
XXXL	5002	768	768	2950299648

We investigated the scaling over $1, 2, 4, 8, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120$ GPUs. However, because of the limited memory on each GPU, larger problems cannot run on a smaller number of GPUs. The results are presented in Fig. 2. In each case, several hundred steps were taken and we use the minimum time per step; an alternative method would be to take the average time per step over the run sample, however the loading and dumping time of the larger volumes skews this data significantly. In this figure, the volume curves represent the strong scaling, while the weak scaling is presented by the different volume sizes for a fixed number of GPUs (i.e. vertical points). We see

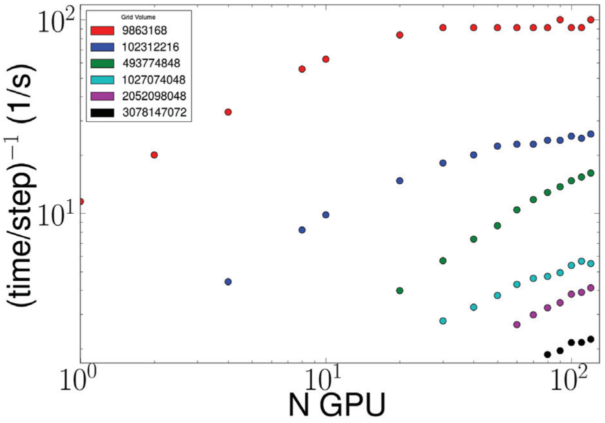

Figure 2: Strong and weak scaling of VORPAL-GPU on GAEA. The horizontal axis shows the number of GPUs used for a given simulation (see Tab. 1). The vertical axis shows the inverse of the time per step.

linear scaling up to certain numbers of GPUs for each volume, afterwhich it falls off. The fall off comes from the aforementioned changing update and communication volume ratios.

COMPARISON AND CONCLUSION

For comparison we benchmarked VORPAL-GPU with an analytical model based on [6]. We see very good agreement and notice convergence of better resolution toward the analytical model (see Fig. 3. We also confirmed good consistency between both gpu, and cpu versions of VORPAL.

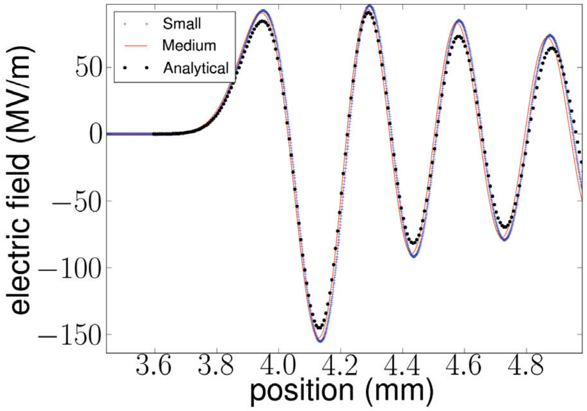

Figure 3: Comparison plot between VORPAL-GPU and an analytical code. The small and medium volume sizes correspond to the S and M volumes. The structure has dimensions $a = 100\mu\ m$ with $b = 120\ \mu m$ with $\epsilon = 5.7$.

Some of the transverse and longitudinal characteristics of the wakefields produced in these simulations are nearly impossible to see at low resolution, therefore the powerful capability of VORPAL on GAEA, allows us to potentially do optimization studies due to the small runtimes needed for large scale problems (see Fig. 4).

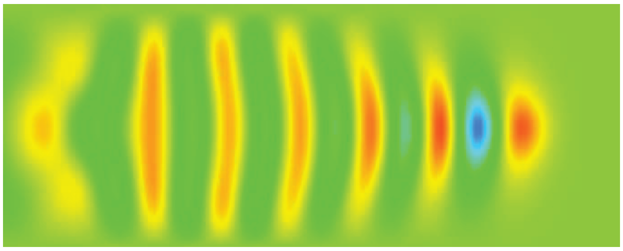

Figure 4: A contour plot of the simulation carried out with VORPAL-GPU shown from a slice in z=0 plane on a L grid volume for the longitudinal E_x field. A gaussian bunch (1 nC, $\sigma_x = 100\ \mu m$ passes through a dielectric structure exciting a wake. The transverse-extended shape corresponds to the combination of the LSE and LSM modes. The peak accelerating field (blue) corresponds to 150 MV/m. The structure has dimensions $a = 100\ \mu m$ with $b = 120\ \mu m$ with $\epsilon = 5.7$.

The next decades in computational development, both in hardware and software will lead to faster, more powerful computation capabilities; which, will make possible the optimization of large scale problems and large data analysis.

We are thankful to Dr. Clyde Kimbal for his support and Dave Ulrick for his help with the GAEA cluster.

REFERENCES

[1] http://www.nvidia.com/object/tesla-supercomputing-solutions.html

[2] http://www.amd.com/us/products/server/processors/6000-series-platform/Pages/6000-series-platform.aspx

[3] C. Nieter, J. R. Cary, J. Comp. Phys. **169**, 448 (2008).

[4] F. Lemery, *et al.*, Proc IPAC12, 3012 (2012).

[5] A. Tremaine, *et al.*, PRE **56**, 7204 (1997).

[6] D. Mihalcea, *et al.*, PRSTAB **15**, 081304 (2012).