Performance Analysis of Parallel FDTD Algorithm on IBM BlueGene Supercomputer Series

A. P. Smirnov[1], **A. N. Semenov**[1], **and A. V. Pozdneev**[2]

[1]Lomonosov Moscow State University, Moscow, Russia
[2]IBM East Europe/Asia Ltd., Moscow, Russia

Abstract— The advances in high performance computing nowadays have increased the need of the computational resources to solve the large-scale problems. The large-scale electromagnetic and optical problems with the size of the order of 400 wavelengths in every dimension for a problem with arbitrary complex geometry structure can be solved. One of the most widespread numerical methods for Maxwell equations solving is Finite Difference Time Domain (FDTD) method. FDTD is based upon Yee lattice and Maxwell equations in integral form and differentional relations from which finite difference approximation is obtained. In this paper, IBM BlueGene/P and BlueGene/Q performance is compared to calculate the large electromagnetic problem using parallel Finite Difference Time Domain method. Introduced early EMWSolver3D was implemented on Lomonosov Moscow State University IBM BlueGene/P supercomputer. Using 1D data decomposition for the highly elongated spatial domains together with the asynchronous transfer operations between nodes of supercomputer it is allows to solve problems up to 10^{10} computational cells on more than a thousand nodes. Parallel implementation of FDTD algorithm based on hybrid MPI/OpenMP approach is used in this paper. The scalability of the algorithm on both BlueGene/P and BlueGene/Q systems that runs in different parallel modes is considered. Parallel processing performance comparison of both systems shows that the IBM BlueGene/Q supercomputer gives near linear speedup and the great efficiency on FDTD algorithm.

1. INTRODUCTION

Nowadays the numerical modeling of electromagnetic wave propagation in microelectronics, nanooptics and biodiagnostics require full-wave solution of system of Maxwell equations on large spatial grids. The large-scale electromagnetic problems with the size of the order of 400 wavelengths in every dimension for a problem with arbitrary complex geometry structure require more than 1 terabyte of memory. Such large spatial grids are used to describe the complex geometry of the problem. One of the most widespread numerical methods for Maxwell equations solving is Finite Difference Time Domain (FDTD) method. FDTD is based upon Yee lattice and Maxwell equations in integral form and differential relations from which finite difference approximation is obtained. Nowadays the advances in the high performance computing have increased the need of the effective parallel implementations of the well known numerical methods. In this work the scalability of the parallel FDTD algorithm on both BlueGene/P and BlueGene/Q systems is considered.

2. PARALLEL FDTD ALGORITHM

Let's consider convolutional FDTD algorithm [1] based upon Yee algorithm [2] that uses material spatial averaging technique [3]. The Yee algorithm solves for both electric and magnetic fields in time and space using the coupled Maxwell's curl equations. With the systems of finite-difference expressions considered in [1], Chapter 3, the new value of an electromagnetic field vector component at any lattice point depends only on its previous value, the previous values of the components of the other field vector at adjacent points, and the known electric and magnetic current sources. Therefore, at any given time step, the computation of a field vector can proceed either one point at a time, or, if p parallel processors are employed concurrently, p points at a time [1]. The proper balance of performed computations and memory consumption is very significant for the efficient implementation of the algorithm. Because of BlueGene distributed memory architecture and strong computational power per gigabyte of memory, we chose to store in memory only data that is required for the next time step.

The parallel FDTD algorithm is based upon domain decomposition technique. For the sake of simplicity, we assume that the subdomains are partitioned along one z-direction and we use 1-D decomposition. The most important factor that influences the parallel efficiency is the load balancing. The amount of data to be exchanged is directly proportional to the cross-sectional area of the subdomain interface. Hence, the optimal domain decomposition scheme goes hand in hand with the appropriate choice of the processor distribution [4].

3. HYBRID MPI/OPENMP PARALLEL PROGRAMMING TECHNIQUE

Hybrid MPI/OpenMP approach is an effective parallel programming technique [5], that can be applied to the parallel implementation of the FDTD algorithm. In case of the FDTD algorithm, inner node OpenMP parallelization is taken upon main loop of electric and magnetic field calculation and does not require complex atomic operations. There is no critical sections between MPI operations, so the algorithm can be effectively executed on high performance computational systems with large number of processor cores per computational node.

4. TARGET BLUEGENE SUPERCOMPUTER SPECIFICATIONS

In this work the computations are performed on the Lomonosov Moscow State University's IBM Blue Gene/P supercomputer [8] and on IBM Blue Gene/Q, installed in IBM Thomas J. Watson Research Center [7].

The BG/Q is the third generation of the IBM Blue Gene line of supercomputers targeted primarily at large-scale scientific applications. Each BG/Q node is based on a system-on-a-chip processor. Functionally, the chip contains 16 compute cores and 1 supplemental core to handle operating system tasks. Each core supports four-way simultaneous multithreading (SMT) and two-way concurrent instruction issue: one integer, branch, or load/store instruction and one floating-point instruction per clock cycle. Within a thread, dispatch, execution, and completion are in order. Four threads per core are sufficient to fully utilize the execution pipelines, prevent memory latency from adversely affecting performance, and avoid the necessity of out-of-order execution. Each core contains a quad SIMD double precision floating point unit (FPU) capable of 8 floating point operations (fused multiply add) per cycle. The 16 user processor cores at 1.6 GHz gives a BG/Q node a peak performance of 204.8 GFlop/s. The node main memory is 16 GByte of directly attached SDRAM-DDR3 providing 42.7 GB/s total bandwidth. A single BG/Q rack contains 1024 BG/Q nodes like its predecessors.

The BG/Q network has a 5-D torus topology; each compute node has 10 communication links with a peak total bandwidth of 40 GB/s. The internal BQC interconnect has a bisection bandwidth of 563 GB/s. An integrated five-dimensional torus serves as the principal node-to-node communication network and also handles collectives and fast interrupts. Compared to the 3D torus architecture of its predecessor BG/P, the 5D torus in BG/Q markedly decreases the number of hops required to reach the farthest node in the machine from any given node, and thereby both increases the bandwidth and reduces the latency of node-to-node communication.

5. ALGORITHM SCALABILITY

The strong and weak scalability are defined in work [6]. The strong scalability is defined as:

$$S_{strong} = \frac{T\left(N_{proc}^{ref}, M\right)}{T\left(N_{proc}, M\right)}, \tag{1}$$

where $T(N_{proc}^{ref}, M)$ — fixed size problem computation time with computational costs with N_{proc}^{ref} number of processors, which taken as reference time, and $T(N_{proc}, M)$ same size problem computation time with N_{proc} processors. For the weak scalability: $T(N_{proc}^{ref}, M^{ref})$ reference problem computation time with computation costs M^{ref} with reference number N_{proc}^{ref} of processors, than $T(N_{proc}, \frac{N_{proc}}{N_{proc}^{ref}} M^{ref})$ — problem computation time with $\frac{N_{proc}}{N_{proc}^{ref}}$ times more with N_{proc} processors. Measure of the weak scalability is defined as:

$$S_{weak} = \frac{T(N_{proc}^{ref}, M^{ref})}{T\left(N_{proc}, \frac{N_{proc}}{N_{proc}^{ref}} M^{ref}\right)}. \tag{2}$$

In case of BG/P, FDTD solver is compiled by IBM XL compiler with keys "-o3-qarch=450d-qtune=450". All tasks are run in SMP mode. For the first strong scalability measurement on BG/P, the spatial grid size is taken $128 \times 128 \times 1024$, so the problem can be fully computed on the one computational node. The second strong scalability measurement on BG/P is performed with spatial grid size $1024 \times 1024 \times 4096$. Bigger problem can load every computational node more effectively. The result scalability plots are presented at Figure 1. As we can see from the plot, in the case of $128 \times 128 \times 1024$ grid, from 2 to 32 nodes the strong scalability is linear. From 64 nodes scalability factor starts to decrease, but still holds near linear. Strong scalability drop is caused by load lowness of each node because of small size of the reference problem. In fact, for the bigger task with spatial grid size $1024 \times 1024 \times 4096$, we have strong scalability drop from 128 to 256 computational nodes.

Let's consider the weak scalability of FDTD algorithm implementation on BG/P. The computation is performed for $1024 \times 1024 \times 16N_{proc}$ and $1024 \times 1024 \times 64N_{proc}$ spatial grids. Figure 2 illustrates the weak scalability plot for both sizes of the problem. In both cases it's fair enough. The weak scalability drop at 512 nodes is presumably caused by unbalancing of computational domain on each node and the size of the sent and received data to the nearby nodes. Also, in the case of the bigger task, the average weak scalability is better.

In case of BG/Q, FDTD solver is compiled by IBM XL compiler with keys "-o5-qhot". All tasks are run in 16, 32 and 64 threads modes.

The strong scalability measurement on BG/Q with spatial grid size: $1024 \times 1024 \times 16384$ is illustrated in Figure 3. As we can see, strong scalability is linear, except 4096 computational nodes. The drop is probably caused by distinctive features of interconnection between two racks of the supercomputer.

The weak scalability measurement of FDTD algorithm on BG/Q is performed for $1024 \times 1024 \times 64N_{proc}$ spatial grid. Figure 4 illustrates the weak scalability plot for three multi-thread modes of

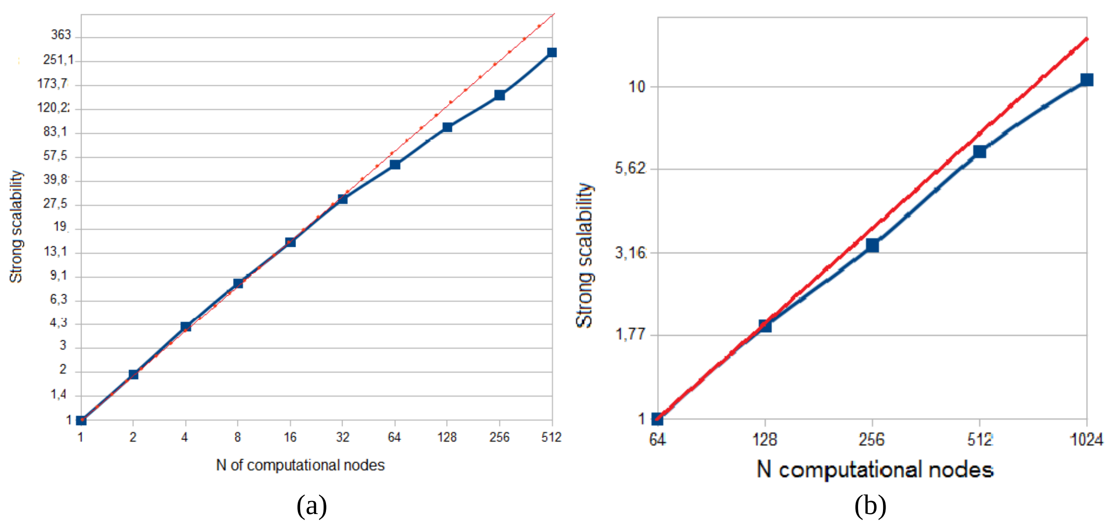

Figure 1: Strong scalability plot on BlueGene/Q with spatial grid size: (a) $128 \times 128 \times 1024$, (b) $1024 \times 1024 \times 4096$.

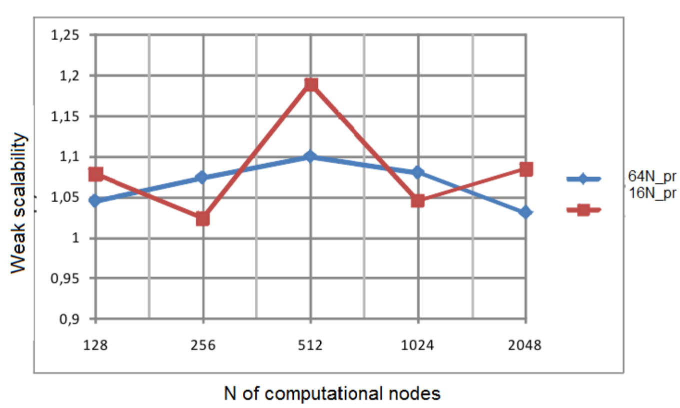

Figure 2: Weak scalability plot on BlueGene/P with spatial grid size: $1024 \times 1024 \times 16N_{proc}$ and $1024 \times 1024 \times 64N_{proc}$.

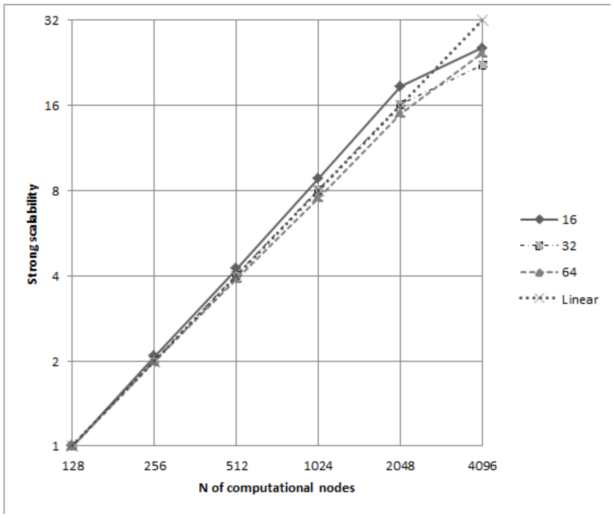

Figure 3: Strong scalability plot on BlueGene/Q with spatial grid size: $1024 \times 1024 \times 16384$.

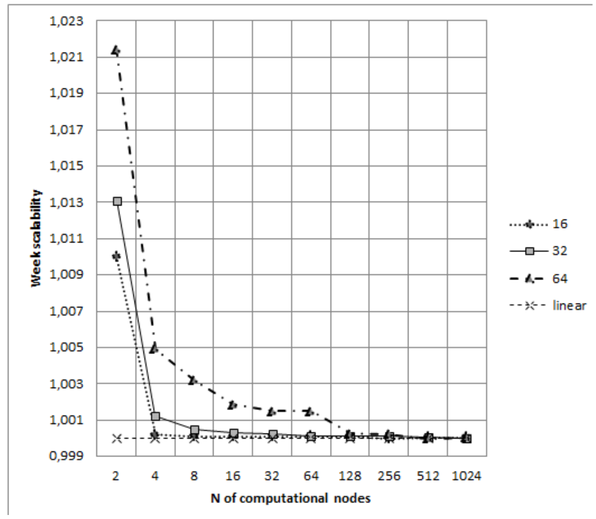

Figure 4: Weak scalability plot for three multi-thread modes of BlueGene/Q with spatial grid size: $1024 \times 1024 \times 64N_{proc}$.

supercomputer. In all three cases, the weak scalability is almost perfect.

6. CONCLUSION

Parallel implementation of FDTD algorithm based on hybrid MPI/OpenMP approach is used in this paper. The scalability of the algorithm on both BlueGene/P and BlueGene/Q systems that runs in different parallel modes is considered. Parallel processing performance comparison of both systems shows that the IBM BlueGene/Q supercomputer gives near linear speedup and the great efficiency on FDTD algorithm.

ACKNOWLEDGMENT

This work is supported by Russian Foundation for Basic Research (RFFI grant 14-01-00337).

REFERENCES

1. Taflove, A. and S. C. Hagness, *Computational Electrodynamics: The Finite-difference Time-domain Method*, Norwood, 2000.
2. Yee, K., "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, Vol. 14, No. 3, 302–307, May 1966.
3. Semenov, A. N. and A. P. Smirnov, "Numerical modeling of Maxwell's equations with dispersive materials," *Matem. Mod.*, Vol. 25, No. 12, 1932, 2013.
4. Yu, W., R. Mittra, T. Su, Y. Liu, and X. Yang, *Parallel Finite-difference Time-domain Method*, 1st Edition, Artech House, 2006.
5. Rabenseifner, R., G. Hager, and G. Jost, "Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes," *2009 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, 427–436, Feb. 18–20, 2009.
6. Chatelain, P., et al., "Billion vortex particle direct numerical simulations of aircraft wakes," *Comput. Methods Appl. Mech. Engrg.*, Vol. 197, 1296–1304, 2008.
7. Haring, R. A., M. Ohmacht, T. W. Fox, M. K. Gschwind, D. L. Satterfield, K. Sugavanam, P. W. Coteus, P. Heidelberger, M. A. Blumrich, R. W. Wisniewski, A. Gara, G. L.-T. Chiu, P. A. Boyle, N. H. Chist, and C. Kim, "The IBM BlueGene/Q compute chip," *IEEE Micro.*, Vol. 32, No. 2, 48–60, Mar.–Apr. 2012.
8. "Overview of the IBM BlueGene/P project," *IBM Journal of Research and Development*, Vol. 52, No. 12, 199–220, Jan. 2008.
9. Smirnov, A. P. and A. N. Semenov, "Full wave Maxwell's equations solver EMWSolver3D," *PIERS Proceedings*, 252–255, Moscow, Russia, Aug. 19-23, 2012.