

A Parallel Algorithmic Approach for Microwave Tomography in Breast Cancer Detection

Meilian Xu¹, Abas Sabouni², Parimala Thulasiraman¹, Sima Noghianian²,
and Stephen Pistorius³

¹University of Manitoba
Dept. of Computer Science
Winnipeg, MB R3T 2N2 Canada
{maryx, thulasir}@cs.umanitoba.ca

²University of Manitoba
Dept. of Electrical and Computer Engineering
Winnipeg, MB R3T 2N2 Canada
{sabouni, sima}@ee.umanitoba.ca

³CancerCare Manitoba
674 McDermot Avenue
Winnipeg, MB R3E 0V9 Canada
Stephen.Pistorius@cancercare.mb.ca

Abstract

Different technologies have been used for breast cancer detections clinically. But they have weaknesses in terms of sensitivity and specificity. Microwave imaging technique, on the contrary, uses the apparent dielectric property contrasts between different breast tissues at microwave frequencies and is a prospective direction to find small tumor at their early stage. Microwave tomography falls in one category of microwave imaging technique. There are two main components in microwave tomography to detect abnormalities in breasts: Genetic Algorithm (GA) and Finite-Difference Time-Domain (FDTD). Both GA and FDTD are time-consuming, but, they are data-parallel in nature. In this paper, we have designed a parallel framework for microwave tomography: parallel GA combined with parallel FDTD. The algorithms are implemented on distributed memory machines running MPI. The execution time of the sequential algorithm (GA and FDTD combined) is 10,131 seconds. The total execution time obtained on 16 processors which is approximately 2000 seconds surpasses the sequential algorithm.

1. Introduction and Motivation

Breast cancer is the second leading cause of cancer deaths in women today and is the most commonly diagnosed cancer in women. In North America, an estimated 202,044 cases were diagnosed and 51,184 patients died from the disease in 2000 [4]. A promising way to lower the mortality rate is to detect the tumor at its early stage, followed by effective treatments.

Currently, the most common method for breast cancer detection is X-ray mammogram screening [9]. But the process is uncomfortable for patients because of the compression on the breast. It also poses the patients to possibly harmful ionizing radiation, which is recognized as a cause of cancer. Besides, the resolution of this technique is limited because the absorption of X-rays is similar for a large number of tissues. Thus, the contrast of different tissue imaging is low, resulting in approximately 20% of missed breast cancer detection and leading to low sensitivity rate. Sensitivity is defined as the rate at which tumors are detected [12].

There are other technologies for breast cancer detection as a second aid to X-ray screening: ultrasound imaging and Magnetic Resonance Imaging (MRI). Ultrasound imaging removes the ionizing radiation in X-ray screening, but it does not solve the problem of low resolution [9]. For this reason, MRI is a suitable choice because MRI has high sensitivity at detecting tissue abnormalities. But, MRI cannot distinguish malignant tumors from benign tumors, which

may increase false positive rate and lead to unnecessary biopsies. This drawback makes MRI a technique with low specificity rate, which is related to false positive error. Also, an MRI system is too expensive and not too many systems can be installed in clinics.

A comparatively more recent technology is active microwave imaging (MWI) for breast cancer detection [3, 7]. The feasibility of MWI technique relies on the high contrasts between the dielectric properties of tumors and those of normal breast tissues at microwave frequencies [11]. Typically, tumors have a permittivity 10-20% higher than that of normal tissues. It reconstructs the material properties of the breast by measuring the scattering of the electromagnetic signals posed on the breast. It is an application of the inverse scattering problem. The inverse scattering problem determines the characteristics of an unknown object (its shape, internal material profile, etc.) from measurement data of radiation from the object [10]. Two approaches in active imaging technique have been developed: the microwave tomography (MT) [8] and the radar microwave imaging [3].

Confocal microwave imaging [3](falls under radar imaging technique) reconstructs the breast by synthetically focusing reflections from the breast. Although it can find small tumors, it does not attempt to reconstruct the exact permittivity profile of the breast. Its emphasis is more on detecting the strong scattering center which may be tumors. This assumption leads to its limitations on breast imaging because a breast is an object with inhomogeneous materials, consisting of fatty tissues, glandular tissues, fibrous tissues, and possible malignant tumors. On the contrary, microwave tomography is a process during which the image reconstruction process involves iteratively matching measured and forward computed data [8]. The process continues until the calculated data converge with the measured data. The output of the forward computing process is the solution to the inverse scattering problem incurred in microwave scattering and represents the dielectric property profile of the breast. Computed data are based on numerical techniques and a model of the object with estimated material properties. In this paper, we consider microwave tomography.

A suitable numerical technique is important to microwave tomography and should be efficient as to calculation time and be accurate in terms of the restored unknown object. There are two components in MT: Genetic Algorithm (GA) and Finite-Difference Time-Domain (FDTD). FDTD is normally selected in solving inverse electromagnetic scattering problem as in this application because it can efficiently model an inhomogeneous object of arbitrary shape [13]. GA is used to find the globally optimized solution to the inverse scattering problem in reasonable amount of time. The paper develops a parallel framework for microwave tomography involving GA and FDTD to breast

cancer detection on a network of computers. To our knowledge, GA combined FDTD for microwave tomography in breast cancer detection has not been parallelized.

The paper is organized as follows. Section 2 focuses on the framework of MT, illustrating how GA and FDTD interact with each other. Section 3 extends GA and parallel GA for the application. Section 4 introduces FDTD in general, a sequential FDTD and a parallel FDTD for our application. In section 5, the implementation environment of our parallel framework is given, including the runtime and speedup comparison. Conclusions are presented in section 6.

2. A Two-Level Parallel Framework for Microwave Tomography

GA and FDTD are the two key components of MT. The efficiency of GA and FDTD is vital to its applicability clinically. FDTD algorithm is essentially computation-intensive, but shows intrinsic characteristic of data parallelism. Thus, a parallel FDTD (PFDTD) is an important approach to improve the efficiency. MT is an application of the inverse scattering problem, and the profile of the breast characteristics is unknown. Different presumed profiles need to be tried as the input of FDTD. In breast cancer detection, combinations of the different tumor types (also including non-tumor scenario) at different positions must be calculated by FDTD to determine which combination leads to the closest FDTD calculation result with the measurement. Therefore, the detection problem involves a global optimization problem by searching all profiles. The optimization problem can be defined in equation 1, where i ranges from 1 to 4, indicating that we impinge plane wave from four directions: east, south, west, and north, and θ represents different angles of the observation points. $E_{i\theta}^{measurement}$ is the measurement at angle θ using i plane wave. $E_{i\theta}^{FDTD}$ is the calculated data at angle θ using i plane wave.

$$f = \max \left[1 - \sqrt{\frac{\sum_{i\theta} (E_{i\theta}^{measurement} - E_{i\theta}^{FDTD})^2}{\sum_{i\theta} (E_{i\theta}^{measurement})^2}} \right] \quad (1)$$

GA is an efficient search algorithm based on principle of natural selection and genetics [5]. It is generally able to find good solutions in reasonable amount of time. The profile combinations in MT increase dramatically in order to improve the image resolution, the sensitivity, and the specificity. Hence, a parallel GA (PGA) is also vital to an efficient MT. In MT, GA is first executed followed by FDTD. The output of the GA is required as an input to compute FDTD. Therefore, the algorithms in MT work in a synchronous manner. However, GA and FDTD by themselves can be parallelized. Therefore, we design a two-level paral-

1el framework for MT in breast cancer detection as shown in Figure 1.

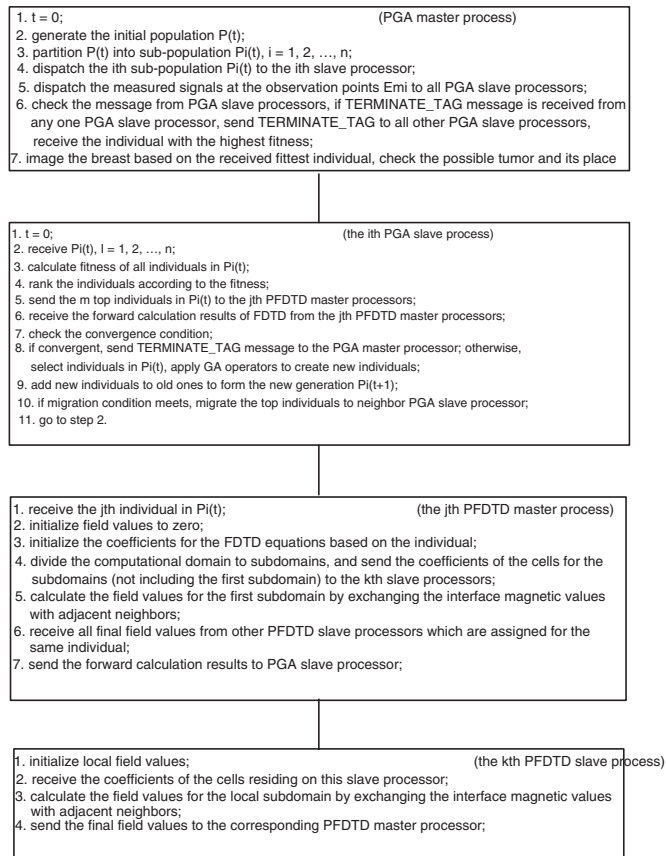


Figure 1. A Parallel Framework for Microwave Tomography. PGA stands for Parallel Genetic Algorithm, FDTD for Finite-Difference Time-Domain, and PFDTD for Parallel FDTD.

In Figure 1, the top two parts are related to PGA and the bottom two parts to parallel FDTD (PFDTD). A master-slave approach is used in the algorithm. One master process (PGA master process) is used to generate the initial population which consists of different combinations. The initial population is divided into subpopulations and dispatched to a number of slave processes (PGA slave process). All PGA slave processes evolve their own subpopulations simultaneously. Each PGA slave process, sends the profiles with highest fitness to a number of processes which act as the master process (PFDTD master process). They work in parallel on different profiles. The PFDTD master processes in turn dispatch work of the calculation to other processes which are called PFDTD slave processes. The PFDTD master processes are responsible to collect the final results and communicate with PGA slave processes. It is the PGA slave

processes that determine whether the FDTD calculation result is close enough with the measurement, thus obtaining the profile of the breast and further determining whether a tumor is present.

3. GA and Parallel GA for MT

3.1. Related Issues in GA

The first issue is operators. Operators are the important part in GA to evolve the population. In GA, each individual is represented as a string. All operators are applied to the strings. The first operator is crossover. Crossover operator takes bits from each parent and combines them to create a child. There are several implementation variants of crossover operator: one-point crossover, two-point crossover, n -point crossover and uniform crossover. They differ in the number of positions to exchange bits in the parent string. For example, two-point crossover can randomly choose two positions in the parent strings and create two children strings by exchanging the blocks of each parent between these two positions. Another operator is mutation. Mutation operator is used to change some bits of the single parent string. Crossover and mutation operators are used both in sequential GA and parallel GA. There is another special operator that is used only in parallel GA as mentioned before. It is migration operator. By using migration operator, each subpopulation is not isolated from each other and can exchange their fittest individuals. According to the time of migration, there exist synchronous migration and asynchronous migration. Synchronous migration means each subpopulation evolves at the same rate and performs all exchanges at the same time; while in asynchronous migration, each subpopulation independently decides the time to migrate. Normally, non-blocking communication is used to deal with the migration because GA is computation-intensive and it is more efficient to overlap communications with computations. Furthermore, non-blocking communication can avoid deadlock in a ring migration topology.

Another issue is selection algorithms. There are different selection algorithms in GA to select the parent individuals to apply the operators to. Roulette Wheel Selection(RWS) and Tournament Selection(TS) are two of them. RWS is similar to spinning a roulette wheel in which the size of the roulette wheel slot for each individual in the population is allocated in proportion to its fitness. Thus, the individuals with higher fitness have better chance of being selected relative to the less fit individuals. This selection scheme is used to select the relatively best individual. It can also be used to select the relatively worst individuals, using the reciprocal of individual's fitness instead of the fitness itself. But this scheme can lower the selection strength in GA. Tour-

nament selection is used to generate mating pool such that individuals can be chosen from the pool to be applied with crossover or mutation operator. There are some other selection algorithms such as elitist selection, rank selection, etc. TS is used in this application.

The final issue is termination condition. Two conditions are always used to terminate GA. One is to decide whether the iteration number of generating populations reaches the pre-defined maximum iteration number. Another one is when there is no improvement in fitness of the population for the last several(pre-defined number of) iterations.

3.2. Sequential GA

Sequential GA can be categorized into two kinds of algorithms according to how the population is replaced for the next generation [6]. They are steady-state GA (SSGA) and generational replacement GA (GRGA). For SSGA, one individual of the population is changed at a time. The child individual can be generated by applying crossover on two parent individuals selected from the population, or applying mutation on one selected parent individual. The newly-generated child individual replaces an individual of the population using different replacement strategies to form the new generation. Replacement strategies include: replace the worst and replace a randomly chosen individual.

On the contrary, GRGA replaces the whole population at each generation. According to their definition, SSGA appears faster than GRGA although the result of SSGA will not be as satisfactory as that of GRGA because SSGA does not explore the whole generation as well as GRGA does. A sequential SSGA is shown in Algorithm 1.

3.3. Parallel GA

Parallel GA can be classified into three categories [2]: global single-population master-slave GA, single-population fine-grained GA, multiple-population coarse-grained GA.

The three parallel GAs differ in how the initial population is distributed and how the task of GA is distributed. The most important difference between sequential GA and parallel GAs is the migration operator that is used only in parallel GA. For example, in a multiple-population coarse-grained GA, initial population is partitioned into subpopulations and are distributed to different processors. Each subpopulation evolves as a sequential GA, except that some fittest individuals in one subpopulation are exchanged with those in other subpopulation. It is called migration operator. The migration operator has to decide on the following parameters: when to migrate between subpopulations, whom to be migrated to other subpopulation(s), how many individuals(migrants) in the subpopulation should be mi-

Algorithm 1 Sequential SSGA (Steady-State GA)

```

t = 0;
Initialize P(t);
Evaluate P(t);
FOR each generation
BEGIN
local_search(Xrandom from P(t));
FOR each pair in Xrandom
BEGIN
Select (x1, x2) from Xrandom;
if( $r < P_c$ )
Xnew = crossover(x1, x2);
else
Xnew = mutate(x1, x2);
Delete Xworst from P(t);
while (Xnew exists in P(t))
mutate(Xnew);
Add Xnew to P(t) to form P(t + 1);
END //end for each pair
Evaluate P(t + 1);
t = t + 1;
END //end for each generation
Output the results;

```

grated, where to migrate the selected migrants(migration topology).

Based on the manipulation of the migrants between subpopulations, parallel GA can be classified as Island Model and Shared Pool Model [2]. In the Island Model, all subpopulations evolve independently of each other, except that each subpopulation occasionally migrates individuals to other subpopulation(s) based on the migration topology. This model is suitable for distributed memory parallel environment. On the other hand, in Shared Pool Model, all subpopulations store their best individuals to a shared pool, and replace some of their own subpopulation with individuals stored in the pool that come from other subpopulations. This model is more suitable for shared memory parallel environment.

4. FDTD and Parallel FDTD for MT

4.1. Sequential FDTD

FDTD is a popular numerical simulation method to solve problems in electromagnetics. It was first proposed by Yee in 1966 [14]. The basic idea is to discretize the electromagnetic computational domain into a collection of Yee cells and calculate the electric fields(E-fields) and magnetic fields(H-fields) of all cells in Cartesian system. There are 6 components in 3D domain: E_x , E_y , and E_z ; H_x , H_y , and

Hz. An illustration of a 3D Yee cell in FDTD is given in Figure 2.

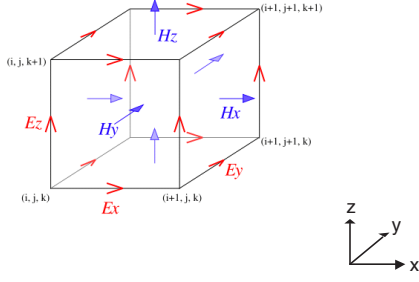


Figure 2. Illustration of a standard Yee cell for FDTD. The E-field components correspond to the edges of the cube, and the H-field components to the faces.

Because FDTD uses central difference, the vector components of E-fields and H-fields of Yee cells are spatially staggering in the Cartesian computational domain, meaning that each E-field vector is located midway between a pair of H-field vector, and vice versa. Furthermore, as H-fields are sampled at a half sampling interval difference than that of E-fields sampling, E-fields and H-fields are updated in a leapfrog scheme for marching forward in time. This implies that E-fields are updated midway during each time-step between successive H-field updates, and vice versa. The relationship can be manifested in the following equations, which are the solutions to our application.

$$E_{zx}|_{i,j}^{n+1} = \left[\frac{1 - \frac{\sigma \Delta t}{2\varepsilon_0 \varepsilon_r}}{1 + \frac{\sigma \Delta t}{2\varepsilon_0 \varepsilon_r}} \right] E_{zx}|_{i,j}^n + \left[\frac{\frac{\Delta t}{\varepsilon_0 \varepsilon_r}}{\left(1 + \frac{\sigma \Delta t}{2\varepsilon_0 \varepsilon_r}\right) \Delta x} \right] \times [H_y|_{i,j}^{n+1/2} - H_y|_{i-1,j}^{n+1/2}] \quad (2)$$

$$E_{zy}|_{i,j}^{n+1} = \left[\frac{1 - \frac{\sigma \Delta t}{2\varepsilon_0 \varepsilon_r}}{1 + \frac{\sigma \Delta t}{2\varepsilon_0 \varepsilon_r}} \right] E_{zy}|_{i,j}^n - \left[\frac{\frac{\Delta t}{\varepsilon_0 \varepsilon_r}}{\left(1 + \frac{\sigma \Delta t}{2\varepsilon_0 \varepsilon_r}\right) \Delta y} \right] \times [H_x|_{i,j}^{n+1/2} - H_x|_{i-1,j}^{n+1/2}] \quad (3)$$

$$H_x|_{i,j}^{n+1/2} = H_x|_{i,j}^{n-1/2} - \frac{\Delta t}{\mu \Delta y} [E_{zx}|_{i,j+1}^n + E_{zy}|_{i,j+1}^n - E_{zx}|_{i,j}^n - E_{zy}|_{i,j}^n] \quad (4)$$

$$H_y|_{i,j}^{n+1/2} = H_y|_{i,j}^{n-1/2} + \frac{\Delta t}{\mu \Delta y} [E_{zx}|_{i+1,j}^n + E_{zy}|_{i+1,j}^n - E_{zx}|_{i,j}^n - E_{zy}|_{i,j}^n] \quad (5)$$

Several preconditions must be met before using FDTD method [13]. Firstly, a computational domain must be established on which the Yee cells are based. Normally, the computational domain is the physical region over which the simulation will be conducted, such as the breast in our application. Secondly, the material of each cell within the computational domain must be specified with their permittivity, permeability and conductivity. Since FDTD allows the material at each cell to be specified, an inhomogeneous object of any shape can be easily modeled. This is the reason that FDTD is used in our application since a breast includes fatty tissues, glandular tissues, fibrous tissue, and possible malignant tumor tissues. Thirdly, a source must be specified. As in active microwave imaging, the property of the pulse sent by the transmitters must be considered. Gaussian pulse is used in our application.

The availability of massive computer resources makes FDTD simulation feasible for different applications. But the computational domains must be finite for finite computer memory. Therefore, for open region problems such as our application, absorbing boundary conditions (ABCs) must be considered. We use Perfectly Matched layer (PML), which employs a material absorber around interested object [1].

Another critical issue in FDTD is its stability because FDTD is a time-marching computational simulation. The time step must satisfy certain preconditions to ensure that the simulation result is stable and correct. In our application, the plane waves are propagating across discrete cells. The time step must be less than the time for the waves to travel adjacent grid points. Otherwise, a nonzero field value of a cell is introduced before the wave can reach the cell, violating causality of the simulation system and resulting in an unstable and inaccurate output [15]. This precondition is called Courant condition.

In Figure 3, a 2D computational domain shows a cross section of a dielectric cylinder, which can be simulated as the cross section of a breast. Yee cells are represented by squares. We assume that a transverse magnetic plane wave along Z axis (TMz) is used and the 2D object is placed in X-Y plane [13]. Therefore, there are only 3 components in the application: Ez, Hx, and Hy. A sequential FDTD with the above issues considered is shown in Algorithm 2.

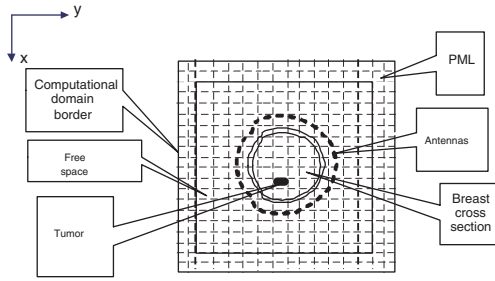


Figure 3. 2D Computational Domain

Algorithm 2 Sequential FDTD using TMz mode for breast cancer detection

```

Initialize E-fields Ez, H-fields Hx and Hy to zero;
Initialize the material of the computational domain,
including the cross section of the breast, the free space,
and PML layers using the coefficients in equation 2
through equation 5;
FOR n = 1 TO MAX_TIMESTEP
BEGIN
Compute Ez of all cells based on their previous values
and neighboring Hx, Hy at the previous timestep, using
equation 2 and equation 3;
Compute Hx of all cells based on their previous values
and neighboring Ez at the previous timestep, using
equation 4;
Compute Hy of all cells based on their previous values
and neighboring Ez at the previous timestep, using
equation 5;
Apply Fourier transform;
END
Apply post processing;
Output the results at the observation points;

```

4.2. Parallel FDTD

FDTD algorithm is computationally intensive due to several reasons. The calculation is complex as shown in equation(2) to equation(5). The size of Yee cells should be small enough such that each cell can be treated as a homogeneous material. The more fine grained the cells are, the more accurate are the results for the inverse scattering problem. This factor is critical for early breast cancer detection to find a tumor at its early stage which is less than several millimeters. However, this granularity implies an increase in the number of cells in the computational domain. The Courant condition posed on a stable FDTD indicates that the largest time-step depends on the number of cells. The complexity of a 2D FDTD algorithm is $O(N^3)$ where N is the number of cells in the computational domain. The

sequential FDTD algorithm takes about 200 seconds for a 600×600 computational domain. For finer granularity, this timing will increase together with memory.

FDTD is data-parallel in nature and exhibits apparent nearest-neighbor communication pattern [15]. Yu et al. [15] introduce three communication schemes in parallel FDTD. The three schemes differ in which components of E-fields and H-fields should be exchanged and which process should update the E-fields on the interface. The division of the computational domain is on the E-field along the Cartesian axis. Note in all these work, FDTD is considered as a stand alone algorithm and parallelized. In our work, FDTD is a subpart of MT application. The FDTD algorithm is the most time-consuming component of MT, as shown in section 2.

We adopt the communication scheme with one cell overlapping region to get a robust implementation. The scheme is shown in Figure 4. The division interface is along E-fields. We divide the computational domain along x axis. Each process attaches one more row of cells to the interface. The extra row of cells are the exact copy of the same row transferred from the neighboring process. Therefore, for process with rank 0 and $(size - 1)$ ($size$ is the total number of processes), only one row of cells are added since they only have one neighbor. For other processes, two rows of cells are added as shown in Figure 4. The E-fields on the interface of adjacent processes are calculated on both processes. The purpose of the scheme is to reduce the necessary communication of E-fields, trying to improve the computation/communication efficiency. The parallel FDTD is given in Algorithm 3.

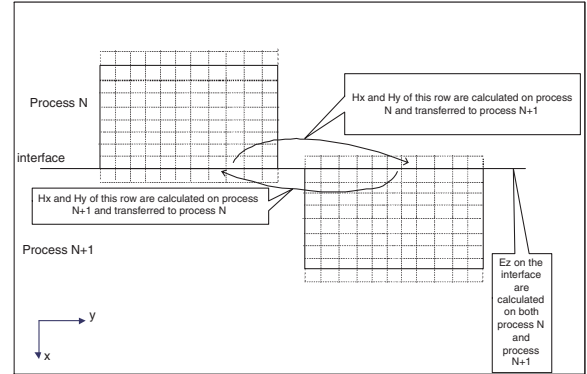


Figure 4. Illustration of the Communication Scheme in Parallel FDTD

5. Implementations and Results

We have parallelized the GA and FDTD on a network of computers. Both the sequential FDTD and the parallel

Algorithm 3 Parallel FDTD using TMz mode for breast cancer detection

```
Initialize E-fields Ez, H-fields Hx and Hy on each process to zero;
IF (process is master process) THEN
BEGIN
Initialize the material of the computational domain, including the cross section of the breast, the free space, and PML layers using the coefficients in equation( 2) through equation( 5);
Decide the sub-domain for each process;
Send material parameters of sub-domain to the corresponding process;
END
ELSE
BEGIN
Receive the material parameters for the cells residing on the process;
END
FOR n = 1 TO MAX_TIMESTEP DO (for all processes)
BEGIN
Compute Ez of all cells residing on the process based on their previous values and neighboring Hx, Hy at the previous time-step, using equation 2 and equation 3;
Compute Hx of all cells residing on the process based on their previous values and neighboring Ez at the previous time-step, using equation 4;
Compute Hy of all cells based on their previous values and neighboring Ez at the previous timestep, using equation 5;
Apply Fourier transform;
Exchange Hx and Hy fields with the neighboring processes;
Synchronize among all processes;
END
IF (process is not master process) THEN
BEGIN
Send the final results to master process;
END
ELSE
BEGIN
Receive results from all other processes;
Apply post processing;
Output the results at the observation points;
END
```

FDTD are implemented in C++ on Fedora Core 5. The parallel version uses C bindings of LAM/MPI 7.1.2. The hardware configuration is as follows: AMD Athlon(tm)64 X2 Dual Core Processor 3800+, with 512KB cache, and 2GHz (variable-speed) clock; 1GB of main memory, and 2GB of swap space, for a total of 3GB of virtual memory; 120GB Seagate ST3120813AS SATA (Serial ATA) disk drive; 1Gb/s Ethernet network interface. The machines are connected to the network via a 100Mb/s Ethernet switch.

Figure 5 shows the total execution time of the algorithms involved in microwave tomography: parallel GA and parallel FDTD. The algorithms are run on five generations with ten individuals in each generation. As the number of processors increase, the execution time decreases. Figure 6 shows the speedup of the microwave tomography algorithms. The speedup on 8 processors is little above 4 while on 16 processors it is close to 6. Though there is an increase in speedup from 8 processors to 16 processors, the speedup increase is gradual. This is due to the synchronization between the genetic algorithm and FDTD. Also, the FDTD algorithm by itself is synchronous. Every i^{th} iteration depends on the $(i - 1)^{th}$ iteration.

Figure 7 and Figure 8 show the execution time of parallel FDTD. These figures show clearly that the FDTD algorithm dominates the total execution time of the microwave tomography algorithm.

The execution time of the sequential algorithm (GA and FDTD combined) is 10,131 seconds. The total execution time obtained on 16 processors which is approximately 2000 seconds surpasses the sequential algorithm.

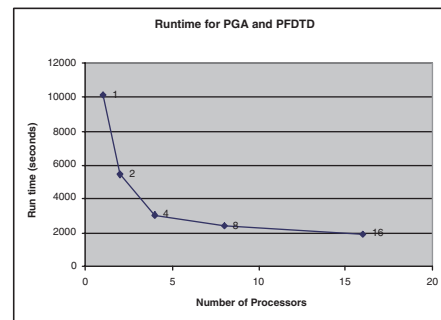


Figure 5. Runtime for PGA and PFDTD

6. Conclusions

This paper focuses on one particularly important problem in the medical field - breast cancer detection. In particular, the paper focuses on microwave tomography technique to detect abnormalities in breasts. There are two main algorithms in this technique: Genetic Algorithm and Finite Difference Time Domain algorithm. We have parallelized both

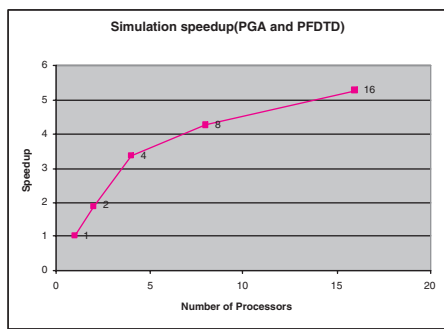


Figure 6. Speedup for PGA and PFDTD

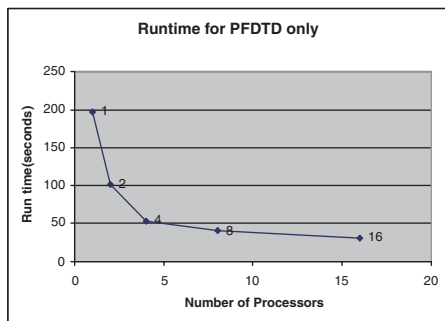


Figure 7. Runtime for PFDTD only

these algorithms and implemented on distributed memory machines. The execution time of the sequential algorithm (GA and FDTD combined) is 10,131 seconds. The total execution time obtained on 16 processors which is approximately 2000 seconds surpasses the sequential algorithm. The results are encouraging and has illustrated the need for parallel computers for such an important application.

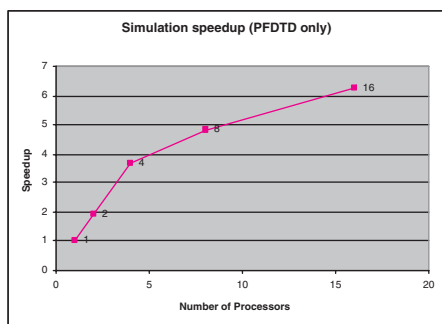


Figure 8. Speedup for PFDTD only

7. Acknowledgement

The first and third authors are thankful to the University of Manitoba Research Grants Program for their support in this research. The third author acknowledges the partial support from Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

- [1] J. Berenger. A Perfectly Matched Layer for the Absorption of Electromagnetic Waves. *Journal of Computational Physics*, 114(2):pp. 185–200, October 1994.
- [2] E. Cantu-Paz. A Survey of Parallel Genetic Algorithms. *Calculateurs Paralleles, Reseaux et Systems Repartitis*, 10(2):141–171, 1998.
- [3] E. C. Fear, P. M. Meaney, and M. A. Stuchly. Microwaves for Breast Cancer Detection. *IEEE Potentials*, vol. 22(1):pp. 12–18, Feb/Mar 2003.
- [4] J. Ferlay, F. Bray, P. Pisani, and D. Parkin. Globocan 2000: Cancer Incidence, Mortality and Prevalence Worldwide. *IARC CancerBase*, Version 1.0(No. 5), 2001.
- [5] J. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor, University of Michigan Press, 1975.
- [6] D. Levine. *A Parallel Genetic Algorithm for the Set Partitioning Problem*. PhD thesis, Illinois Institute of Technology, Argonne, IL, May 1994.
- [7] X. Li and S. C. Hagness. A Confocal Microwave Imaging Algorithm for Breast Cancer Detection. *IEEE Microwave Wireless Components Lett.*, vol. 11:pp. 130–132, 2001.
- [8] P. Meaney, M. Fanning, D. Li, S. Poplack, and K. Paulsen. A Clinical Prototype for Active Microwave Imaging of the Breast. *IEEE Transactions on Microwave Theory and Techniques*, Vol. 48(No. 11):pp. 1841–1853, November 2000.
- [9] M. Patlak, S. J. Nass, I. C. Henderson, and J. C. Lashof. *Mammography and Beyond: Developing Technologies for the Early Detection of Breast Cancer: A Non-Technical Summary*. National Academy Press, 2001. Available from <http://www.nap.edu/catalog/10107.html>.
- [10] I. Rekanos and T. Tsiboukis. An Iterative Numerical Method for Inverse Scattering Problems. *Radio Science*, 34:1401–1412, 1999.
- [11] L. Sha, E. R. Ward, and B. Stroy. A Review of Dielectric Properties of Normal and Malignant Breast Tissue. In *Proceedings IEEE SoutheastCon*, 2002.
- [12] R. N. Strickland. *Image-Processing Techniques for Tumor Detection*. Marcel Dekker, Inc., 2002.
- [13] A. Taflove and S. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method, Second Edition*. Artech House, 2000.
- [14] K. Yee. Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media. *IEEE Transactions on Antennas and Propagation*, Vol. AP-14(No. 8):pp. 302–307, May 1996.
- [15] W. Yu, R. Mittra, T. Su, Y. Liu, and X. Yang. *Parallel Finite-Difference Time-Domain Method*. Artech House publishers, July 2006.