JBUON 2016; 21(1): 252-260
ISSN: 1107-0625, online ISSN: 2241-6293 • www.jbuon.com
E-mail: editorial_office@jbuon.com

ORIGINAL ARTICLE

Monte Carlo modeling of linear accelerator using distributed computing

Sotirios Stathakis[1], Federico Balbi[2], Anthony T. Chronopoulos[2], Niko Papanikolaou[1]

[1]University of Texas Health Science Center San Antonio, San Antonio, TX 78229, USA; [2]CS Department, University of Texas, San Antonio, Texas 78249, USA

Summary

Purpose: *The distributed computing implementation of the EGSnrc Monte Carlo system using a computer cluster is investigated and tested.*

Methods: *The computational performance was tested for various scenarios with different number of computers used in order to assess the efficiency of the cluster. The presented computation times and efficiencies include the linac head modeling with full simulation of the multi leaf collimator (MLC) geometry (including tongue and groove) and stereotactic radiosurgery cones as well as the radiation transport simulation and dose computation within water phantom and patient geometry.*

Results: *The simulations performed in the cluster environment had the same total number of histories recorded and simulated as the simulations in a single computer. The statistical uncertainty achieved was the same for all scenarios.*

Conclusions: *The investigated approach shows almost linear performance scaling vs number of computers involved.*

Key words: *distributed computing implementation, EGSnrc, Linac, Monte Carlo*

Introduction

The Monte Carlo (MC) method is based on statistical simulation method using random samplings [1-3]. When MC is applied to radiation transport problems, one simulates the paths of individual particles, using machine-generated random numbers and sampling, following the probability distributions of the physical processes being studied. Using the MC approach in radiotherapy treatment planning dose calculation, it is possible to compute accurately the dose in most cases [3] (and refs there-in). A lot of works has been published on improving the accuracy of the MC dose calculation methods for treatment planning systems in order to reproduce all beam geometries and beam modification devices and taking into consideration all heterogeneities in the full 3-dimensional (3D) patient geometry.

There exist general purpose MC codes such as EGSnrc, MCNPX, GEANT4 and PENELOPE [4-7]. However, these codes are not very efficient. Thus a lot of efforts are devoted to speedup MC codes in particular for dose calculation in radiation therapy [3,8,9]. All aspects of the electron and photon transport must be taken into account in order to obtain accurate results in a heterogeneous phantom. There are many MC codes currently available for radiation transport calculation for applications in medicine (e.g. EGSnrc [10], MCNP [11], GEANT4 [12]). Also, there exist MC codes for simulation of linear accelerators (linac) and dose calculation for patient treatment (e.g. BEAMnrc [13] and DOSXYZnrc [14] which are based on

Correspondence to: Anthony T. Chronopoulos, PhD. Computer Science Department, University of Texas, San Antonio, Texas 78249, USA..
Tel: 210-458-7214, Fax: 210-458-4437, E-mail: antony.tc@gmail.com
Received: 15/08/2015; Accepted: 20/10/2015

EGS4/EGSnrc). There are also general-purpose MC codes (e.g. PENELOPE, MCNP [15,16]).

Two assumptions need to be satisfied for the MC codes to run: a) particles from a radiation source interact with matter but not with each other; and b) radiation histories do not perturb each other. Both these assumptions provide very good approximations for most purposes.

A major obstacle in using MC calculations in radiotherapy treatment planning, which needs high accuracy, is that it is computationally intensive. However, the MC method lends itself well to implementation on parallel and distributed systems. Parallel systems are clusters of homogeneous processors connected by a fast network (with shared or distributed memory). Distributed systems are ensembles of (heterogeneous or homogeneous) computers (with private memory) connected by a slow network (e.g. Ethernet for local clusters, or the Internet for Grid or Cloud systems). Some widely-used programming environments for traditional parallel systems are OpenMP (for shared memory), PVM, and MPI [17-19]. MapReduce is yet another programming environment used in Cloud systems [20].

The use of parallel or distributed computing for MC simulation offers an efficient approach towards improving the overall computational time. As mentioned above, there are efficient and accurate codes that have been optimized for radiotherapy. These codes when implemented on parallel and distributed systems can make it possible to use MC for clinical treatment planning.

The following works are on existing MC particle transport simulation packages, which have been parallelized. Implementations using MPI are: MC4 [21], Dose Planning Method [22], PENELOPE [23], Geant4 [24], and MCNP [25]. MCNP also includes support for PVM for heterogeneous platforms and OpenMP for exploitation of shared-memory parallelism. Instead of parallelizing the source code, it is possible to run in parallel several instances of the MC application and combine (using Linux file system functions) the outputs to obtain the final result. This approach has been demonstrated in both local clusters and Grids or general distributed systems [26-30]. Also, recent implementations using MapReduce and GPU programming have been published (see e.g. [31,32]).

This paper describes the distributed implementation of the EGSnrc MC code for the simulation of linac models and radiation transport in patients using a local cluster of computers, and investigates the sources of degradation in efficiency and speedup in the parallel version of the code. Firstly, the simulation of radiation transport through the linac head was studied. The results of the simulation were then used to calculate the dose in water for the commissioning of the photon beam model. Secondly, a clinical example of dose calculation in a patient was simulated using the computer cluster.

Methods

MC model of the treatment head of the Varian 23Ex linear accelerator

In order to model the linac the EGSnrc/BEAMnrc code was used. According to manufacturer's specifications about the geometry and the materials, the MC model of the treatment head of the Varian 23EX linac (Varian Medical Systems, Palo Alto, CA) was built using the following component modules : target, primary collimator, flattening filter, monitor chamber, secondary collimator, MLF and reticle. The energy of the incident electron beam was tuned to match the measured percent depth dose (PDD) curves and the spot size was tuned according to the measured profiles. A Gaussian distribution of 6.2 MeV energy and 0.125 cm FWHM was chosen as the one resulting in the best agreement with the measurements.

Simulation of the measured percent depth dose curves and profiles using BEAMnrc/DOSXYZnrc code

The PDD curves and the field profiles measured were to be reproduced using the MC model of the linac. In all cases, phase space files for the different field sizes of the 6MV photon beam were created using the EGSnrc/BEAMnrc system. A phase space file contains data relating to particle position, direction, charge, etc. for every particle crossing a scoring plane. Phase space files can be output for each scoring plane in an accelerator. The phase space files were scored below MLC. The cutoff energies used for the simulations were ECUT = 700 keV for electrons and PCUT = 10 keV for photons. The energy thresholds for δ-ray production (AE) and for bremsstrahlung production (AP) were 700 keV and 10 keV, respectively. The maximum fractional energy loss per electron step (ESTEPE) was set to 0.04 and the default parameters were chosen for the PRESTA algorithm [33]. The dose per incident particle deposited in water was computed in each case using the EGSnrc/DOSXYZnrc code.

The simulation setup was created so that it would reproduce the conditions of the measurement in each case. The phase space files were incident on a water phantom keeping the source to surface distance the same as the measurements (100 cm). The resolution of the water phantom was chosen as 0.1 cm for the x and y axis (plane vertical to the beam axis). For the z axis

(parallel to the beam) a resolution of 0.1 cm was chosen for the built up region, while for depths between 2 cm and 25 cm a resolution of 0.5 cm was considered sufficient for our purposes.

The profile measurements were normalized to the depth of maximum dose calculated for each field, while the percent depth dose curves were converted to absolute dose calculations using as a reference the calculated dose per particle at 1.5 cm depth of water, for a 10×10 cm^2 field at 100 cm SSD.

In order to achieve less than 1% statistical uncertainty, a number of about 2×10^8 per cm^2 initial histories were simulated with the EGSnrc/BEAMnrc code and 5×10^8 with the EGSnrc/DOSXYZnrc code for the dose calculations.

Further validation of the beam model was carried out for stereotactic cones dosimetric characteristics. The stereotactic cones were simulated as a separate linac using the phase space file of 8 cm x 8 cm produced after the validation of the main linac model. The phase space was used as input to create a new phase space file at the end of each stereotactic cone. The beam model was validated against measurements with microdiamond ionization chamber (PTW, Freiburg, Germany) and Kodak EDR2 film. The resolution of the dose grid for our simulations was set to 0.1 mm x 0.1 mm x 1 mm and the size was set to 10 cm x 10 cm x 30 cm. A total of $3x10^8$ particles were scored at the plane of the phase space file and $10x10^9$ particles were simulated using EGSnrc/DOSXYZnrc to calculate the deposited dose.

Distributed simulation using single-server-multiple-client

We used a master-worker distributed model to implement the method in the distributed system. The master (server) assigns the work to the workers (clients) and gathers (and combines) their results. We consider the master-worker model mapped to an actual cluster of processors (nodes) connected via a network.

The EGSnrc code and the BEAMnrc/DOSXYZnrc codes were installed in a linux environment and parallelized using a cluster of p (p=1,..,32) workers (processors). The master is also one of the workers. Each node (processor) is a core of an Intel duo core Pentium4 CPU running at 3.20 GHz (actual speed: 3.1933GHZ) with 1 MB cache and 2 GB physical memory. The CPUs of the cluster are connected via a 1 GB Ethernet network. The hard disk capacity of the server is total of 4 TB and each node has an 80 GB hard disk.

The standard metrics for a distributed simulation are the parallel execution time, speedup and efficiency defined as follows.

The parallel execution time is: $T_p = T_O + max\{t_j\}$, $j=1,..,p$, where t_j is the time taken by each processor (or worker) to complete its tasks in parallel and T_O is the ("overhead") time. The overhead is the time spent by the processors in communication and in synchronization in order to complete the simulation (in parallel) and for the partial results to be communicated/combined into the final problem solution.

Speedup, S_p, is the ratio of execution time (for complete problem solution) on a single processor over the execution time using p processors: $S_p=T_1/T_p$. Efficiency is defined as: $E_p = S_p/p$. The upper bound for the speedup is p and that for the efficiency is 1. The best possible speedup $S_p=p$ and efficiency E_p =1 can be attained if the problem solution can be perfectly parallelized and computed in parallel by the p processors. In this case, the overhead T_O =0. In practice, the overhead is T_O > 0,which leads to degradation in speedup and efficiency.

A BEAMnrc simulation can be split into smaller jobs which can then be run on different processors in parallel to reduce the elapsed time required for a simulation. Traditionally, parallel and distributed computing uses Unix/Linux and that one must have a network queuing system such as PBS or NQS [34,35]. In previous versions of BEAMnrc [35], a Unix script process was used to automatically create the individual input files for parallel jobs and to submit them. It also automatically sets the random number seeds to a different values in each input file and the phase space source inputs, IPARALLEL and PARNUM.

In order to run the MC simulations in parallel we developed a set of Linux codes to wrap and concurrently run the simulation code (BEAMnrc\EGSnrc and DOSXYZnrc\EGSnrc) on different nodes and manage the input and output files of each node. Input and output files reside on a directory common to all nodes by using a Network File System (NFS). Our development has been performed on the Intel/Linux platform. The main code to run the parallel simulation, which resides on the server also, uses secure shell (ssh) in order to remotely (and securely) invoke a shell command.

It is important that each parallel job starts with a different state of the random number generator, otherwise one will end up combining identical results at the end of a parallel run, compromising both the results and the uncertainty estimate on them. In our implementation the random number seed JXXIN for each job created is determined as:

$$JXXIN=JXXINinput+i\_parallel,$$

where *JXXINinput* is the value of *JXXIN* in the input value and *i_parallel* is the sequence number of the job to run. Before running the simulation, an input file for each node is produced. The only difference between these files is the randomization seed in order to have the client produce different output. When the input files are ready, the main code (on the server) reads a file that contains the address or name of each node. For each node, it remotely runs EGSnrc using the node specific input file name. At this point, each node runs its own independent simulation and produces an output file. When all nodes terminate, then the output files are merged. The main code on the server performs the following: (1) Produces input for each node; (2) Runs EGSnrc on each node and provides specific input pa-

rameters; (3) Merges output files. The phase space files created concurrently are combined after the completion of all the runs using a program called *addphsp* provided during the installation of BEAMnrc which adds phase space files from different runs into one, and in the case of DOSXYZnrc, a separate Mortran code is written to recombine all the 3D dose files (3ddose) created. Note that *addphsp* requires twice the disk space of the total size of all phase space files being combined, since files are not automatically deleted after they have been added. We present the program scripts that achieve the concurrent execution in the Appendix.

Results

A MC model of a Varian linac 23EX machine with 120 Millennium MLF has been tested and benchmarked to produce phase-space files to be used in future research in order to test the feasibility of clinical application of MC simulation for calculation of dose in phantom and patient CT. Comparison of data from a Varian 23EX machine equipped with 120 Millennium MLCs, with MC simulations of regular fields has been shown, with a very good agreement between the MC simulation and data from the commissioning of the linac and film dosimetry.

The results from the linac simulation using 32 nodes for the computations are shown in Figure 1. The calculated profiles and percent depth dose curves were in very good agreement with the respective measured ones. The agreement in most cases was within 1% dose difference or 1 mm. The statistical uncertainty for the MC calculations was less than 2% at the location of maximum dose (dmax) and in most cases approached 1%.

In Figure 1(a), the percent depth dose curves are shown for field sizes ranging from 5x5 cm^2 to 20x20 cm^2. The MC simulations were able to reproduce the measurements at the buildup part of the curve as well as after the dmax for all field sizes.

Figure 1(b) shows the comparison of profiles at 10 cm depth in water between Monte Carlo simulation and measurements for fields sizes 5x5 cm^2 to 20x20 cm^2 defined by jaws. The agreement was well within 1 mm at the penumbra region and within 1% in the plateau region of the fields. The discrepancy of 1.0 mm was probably due to inaccurate model of the MLC leaf (precisely determine the radius of the round leaf ends). This can be improved if a more detailed modeling of the MLC is implemented. Also, such deviations could be attributed to inaccuracies in the measurements due to setup uncertainties of the ionization chamber, leveling of the ionization chamber, and water tank and simplifications of the simulation components for increased efficiency. Overall, the simulations carried out during this study were very well within acceptable criteria, especially since the behavior of the MLC after dmax was in good agreement with measurements. The current model was within acceptable clinical criteria and it can be used to accurately determine the dose.

A comparison of the stereotactic cone dose calculations is shown in Figure 2. The MC calculated profile of the 0.75 cm cone at 10 cm depth is shown in comparison with the measurements using a microdiamond detector and EDR2 film. The agreement was within 0.2 mm from either the film or the microdiamond detector. Overall, the agreement between measurements and calculations was very good and within the measurement error.

Figure 3 shows the results for a clinical application. The dose was calculated for a lung cancer patient using EGSnrc/DOSXYZnrc. The dose was calculated for 3 volume modulated arcs with 43, 100, and 45 control points, respectively. The dose grid resolution was set to 0.3x0.3x0.3 cm^3. The dose distribution was calculated twice, once using a single processor and once using 32 processors on our parallel processing cluster. The same number of histories was simulated in both cases to achieve statistical uncertainty of 1% at the maximum dose region in both cases. The time for the single processor calculation of the dose was 182 min while the time for the 32 CPU calculation was 6.5 min.

Discussion

The parallelization of the BEAMnrc allowed us to complete the simulations in shorter times. The speed increase obtained was almost linear in terms of the number of processors used. These results are shown in Figure 4. The linac simulations to obtain the required phase space files for each case defined by collimating jaws or MLC and then to calculate the respective dose distribution in water, were carried out using 1 to 32 nodes of our cluster. For each run, the same number of particles was simulated to achieve the same level of statistical uncertainty.

Our proposed method shows that, by utilizing a large number of idle processors, we can use the MC method to calculate the dose to a phantom

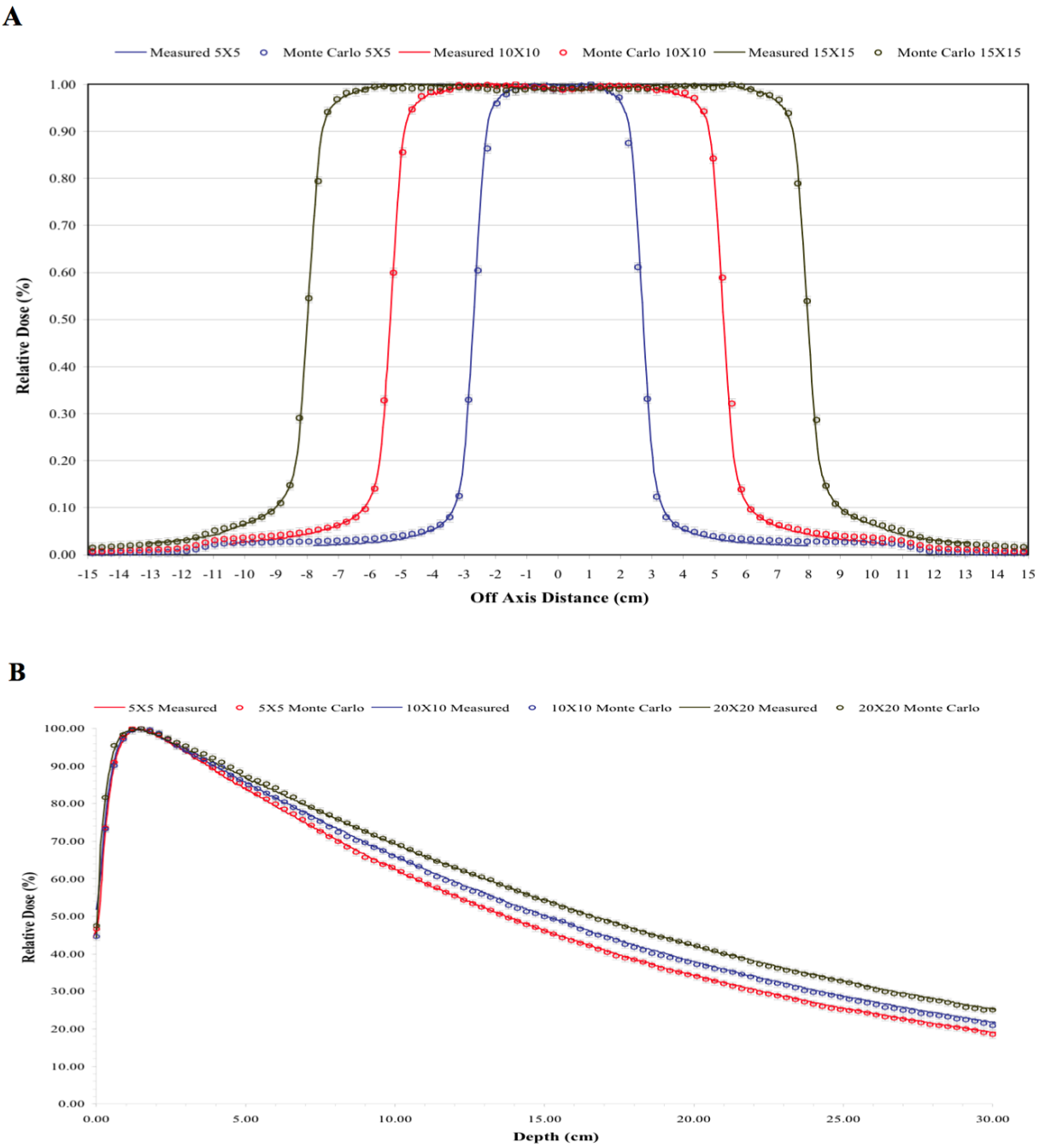

Figure 1. Comparison between measured and calculated percent depth doses (**A**) and profiles at 10cm depth (**B**) for 5x5cm^2, 10x10cm^2 and 20x20cm^2 jaw defined fields.

in a short time, which depends on the number of processors available without losing accuracy. In Figure 4, one can see that the speedup vs number of processors used is almost linear. Using 32 processors would reduce the required time for a particular simulation by approximately 28. The efficiency of the calculations as shown in Figure 4 is about 0.85 when 32 processors are used. This loss of efficiency is mostly attributed to the time required to recombine the phase space files or 3ddose files.

This increase in speed of calculation is very valuable, because accurate MC simulations can be utilized in the clinic to calculate the dose of clinical plans within acceptable time and hence increase the accuracy of the patient dose calculations. To this date full MC calculations as described in this study are not used routinely in the clinic because they are very time-consuming. Most vendors, in order to provide MC calculations engines for treatment planning systems, use variance reductions techniques which on one hand

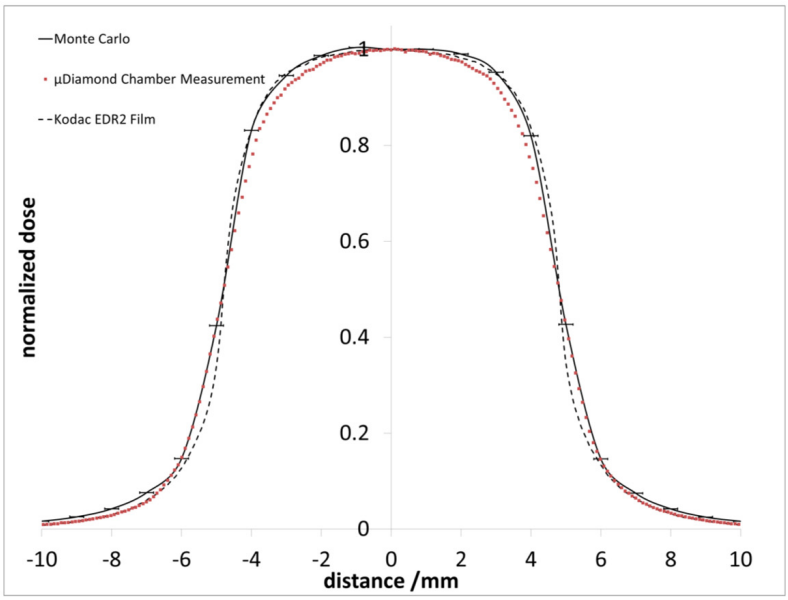

Figure 2. Dose profile comparison at 10 cm depth in water of Monte Carlo simulation of the BrainLab 0.75 cm stereotactic cone vs measurements with micro-diamond detector and EDR2 film.

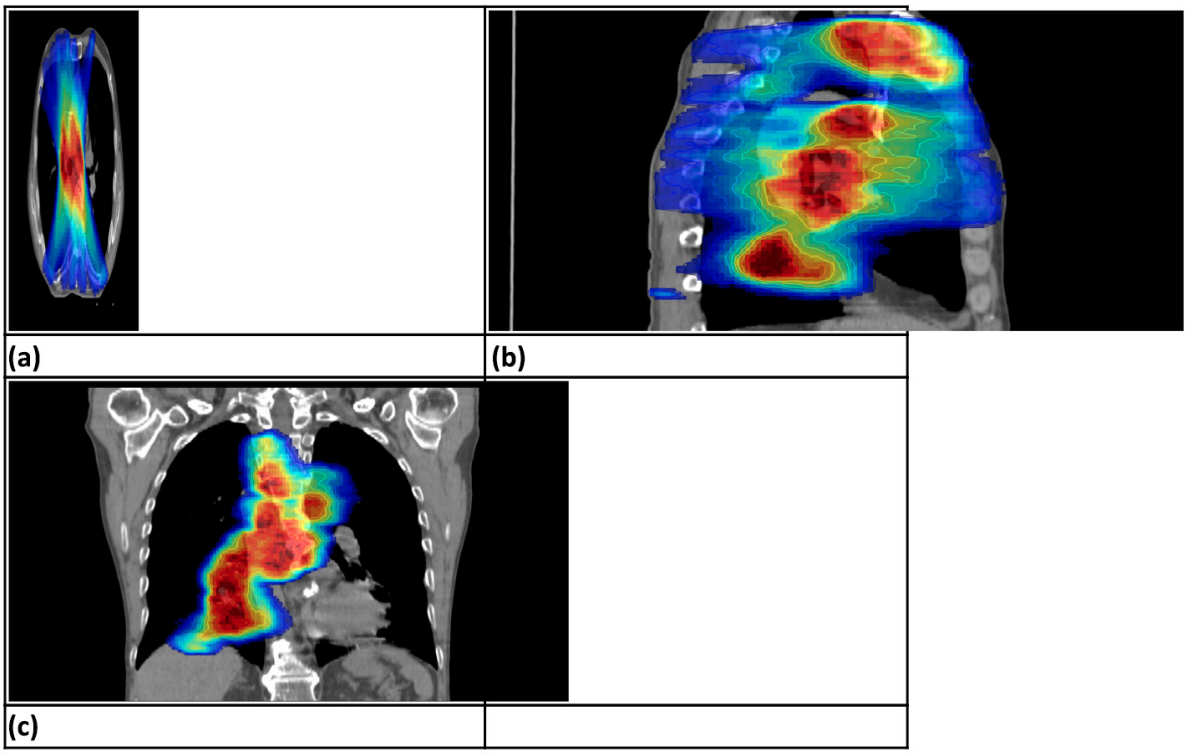

Figure 3. Clinical case results of a lung cancer patient. **a**) axial, **b**) sagittal, and **c**) coronal view through isocenter.

reduce the calculation time but on the other hand are less accurate.

Furthermore, the beam phase space and dose calculations for the stereotactic cones were benefitted by the increase in the number of processors used during the simulations. Such measurements require high precision which can be achieved by increasing the number of histories of the simulation. The fact that the dose grid voxel size is about 100 times smaller in such calculations and the requirement for statistical error is 1% or less for the maximum dose, increases the computational

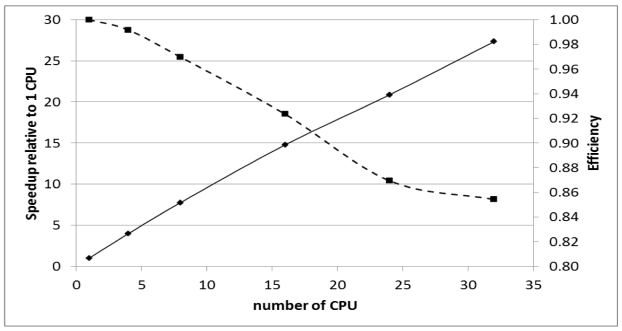

Figure 4. Plot of the Speedup (solid-line curve, measurements on left vertical axis) and Efficiency (dotted-line curve, measurements on right vertical axis) versus the number of workers.

time requirements substantially. The simulations in our case were performed within 1 hr for each stereotactic cone. The same simulations on a single processor required 34.2 hrs. More specifically, a single CPU calculation for the same number of histories to required 34.2 hrs, using 2 CPUs requires 17.4 hrs, 4 CPUs 9.0 hrs, 8 CPUs 4.7 hrs, 16 CPUs 2.6 hrs, 24 CPUs 1.56 hrs and 32 CPUs 1.4 hrs. The processing times for each of the above simulations were 0, 0.09, 0.13, 0.13, 0.13, 0.14, 0.14 hrs, respectively.

Conclusions

A MC dosimetric characterization has been performed for a Varian 23Ex linac equipped with MLC in our department. The commissioning of the linac was performed with clinical data acquired during the installation of the linac. Using the commissioned MC linac head model, we found that the calculated dose from the MC system agreed with the measured data within clinically acceptable criteria from low- to high-dose regions. Thus, this evaluated MC system can be an effective tool for intensity modulated radiation treatment (IMRT) dose calculation and dosimetry quality assurance.

Acknowledgement

The research was partially supported by NSF grant (HRD-0932339) to the University of Texas at San Antonio, USA.

Appendix

In this section, we present the computer programs (in PHP programming language) that achieve the distributed simulation.

Configuration of the Distributed Environment

We configure the distributed environment by using an input file called “Nodes”. This is a text-file that lists the computing-nodes that are used to run a simulation. Every line of the file contains a host name (or DNS entry or IP address) and a unique ID. The unique ID is useful when a computing-node belongs to a multi-core host. In this case, the ID uniquely identifies a worker (either a core of a multi-core CPU host or single core CPU host). Take for example the following “nodes” file: {node 1 1 node2 2 node3 3 node3 4 node4 5 node4 6 node4 7 node4 8}. This “nodes” file is set up to run the simulation a total of 8 workers. The computing-nodes node1 and node2 will have 1 worker each. The computing-node node3 will have 2 workers because it is listed twice and finally the computing-node node4 will have 4 workers (IDs 5, 6, 7, 8).

This structure of the “nodes” file also allows to easily take advantage of new hardware by just adding the new computing-nodes to the “nodes” file. No specific Linux/Unix distribution is required as long as the PHP program (below) and ssh are installed on each computing-node. We next present the distributed program in PHP programming language.

Program/Scripts

```
#!/usr/bin/php
<?php
if ($argc<2)
{
  echo “Error: missing input file name.\n”;
  echo “ex: create_input parallel\n\n”;
  exit();
}
$input_file = $argv[1]; // 1st param
$path = “/home/monte/egsnrc_mp/BEAM_test6”;
$fh = fopen(“/home/monte/nodes”, “r”);
while (!feof($fh))
{
 $line = fgets($fh);
  if ($line==null) break; // skip empty lines
  list($node_name, $node_id) = sscanf($line, “%s %s”);
  $command = “cp $path/$input_file.egsinp $path/” . $input_file . “_w$node_
id.egsinp”;
  echo “Created input file $path/” . $input_file . “_w$node_id.egsinp\n”;
  exec($command);
}fclose($fh);
?>
```

runsim.php: this is the main script that spawns the simulation among workers. For our simulation runsim reads nodes and it remotely invokes the following shell command line on each node:

```
ex beamnrc <input_file> 700icru\ ><input_file>.log &
```

The remote invocation is possible using the secure shell command ssh -x -c

```
#!/usr/bin/php
<?php
// run jobs in parallel using node name and id defined on /home/monte/nodes

if ($argc<2)
{
  echo “Error: missing input file name.\n”;
  echo “ex: runsim parallel\n\n”;
  exit();
}

$input_file = $argv[1]; // 1st param

$fh = fopen(“/home/monte/nodes”, “r”);
while (!feof($fh))
{
  $line = fgets($fh);
  if ($line==null) break; // skip empty lines
  list($node_name, $node_id) = sscanf($line, “%s %s”);
  $command = “ssh -x -c arcfour $node_name \”ex beamnrc “ . $input_file .
“_w$node_id 700icru\” > “ . $input_file . “_w$node_id.log &”;

  echo “Submitted job to $node_name ID $node_id\n”;

  exec($command);
}
fclose($fh);

?>
```

References

1. Rogers DW. Fifty years of Monte Carlo simulations for medical physics. Phys Med Biol 2006;51: R287-301.
2. Chetty I, Curran B, Cygler JE et al. Guidance report on clinical implementation of the Monte Carlo method in external beam radiation therapy treatment planning: Report of the AAPM Task Group No. 105. Med Phys 2007;34:4818-4853.
3. Jabbari K. Review of fast Monte Carlo codes for dose calculation in radiation therapy treatment planning. J Med Signals Sens 2011;1:73-86.
4. Kawrakow I, Rogers DWO. The EGSnrc code system: Monte Carlo simulation of electron and photon transport. NRC, Report PIRS-701, 2000.
5. Briesmeister JE. MCNP-A general Monte Carlo N-particle transport code, Version 4C. Report LA-13709-M, Los Alamos National Laboratory, NM, 2000.
6. Agostinelli S, Allison J, Amako K et al. GEANT4 A simulation toolkit. Nucl Instrum Methods Phys Res 2003;506:250-303.
7. Salvat F, Fernández-Varea JM, Sempau J. PENELOPE A Code System for Monte Carlo Simulation of Electron and Photon Transport. OECD Nuclear Energy Agency, France, 2003.
8. Kawrakow I. VMC++, electron and photon Monte Carlo calculations optimized for Radiation Treatment Planning. Advanced MonteCarlo for Radiation Physics, Particle Transport Simulation and Applications: Proc Monte Carlo 2000 Meet, Lisbon. Kling A, Barao F, Nakagawa M, Távora L, Vaz P (Eds). Springer, Berlin, 2001, pp 229-236.
9. Jabbari K, Keall P, Seuntjens J. Considerations and limitations of fast Monte Carlo electron transport in radiation therapy based on precalculated data. Med Phys 2009;36:530-540.
10. Rogers DWO, Walters B, Kawrakow I. BEAMnrc Users Manual, Report PIRS 509(a) rev H, 2004.
11. Walters BRB, Rogers DWO. DOSXYZnrc Users Manual. NRC Report PIRS 794 (rev B), 2004.
12. Salvat J, Fernández-Varea M, Sempau J. PENELOPE—A Code System for Monte Carlo Simulation of Electron and Photon Transport. Workshop Proc, OECD Nucl Energy Agency, France, 2003.
13. OpenMP: https://computing.llnl.gov/tutorials/openMP/
14. PVM: http://www.epm.ornl.gov/pvm/pvm_home.html
15. MPI: http://www.mcs.anl.gov/research/projects/mpi/
16. Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters, OSDI '04: 6th Symposium on Operating Systems Design and Implementation, pp 137-149, 2004.
17. Hadjidoukas P, Bousis C, Emfietzoglou D. Parallelization of a Monte-Carlo particle transport simulation code. Computer Physics Commun 2010;181:928-936.
18. Tyagi N, Bose A, Chetty IJ. Implementation of the DPM Monte Carlo code on a parallel architecture for treatment planning applications. Med Phys 2004;31:2721-2725.
19. Cruise RB, Sheppard RW, Moskvin, VP. Parallelization of the PENELOPE Monte Carlo particle transport simulation package. Proc Nucl Math Computat Sciences: A Century in Review. A Century Anew, Gatlinburg, Tennessee, 2003.
20. Sutherland K, Miyajima S, Date H. A simple parallelization of GEANT4 on a PC cluster with static scheduling for dose calculation. J Phys Conf Ser 2007;74:12020, 1-8.
21. Forster RA, Cox LJ, Barrett RF et al. MCNP Version 5. Nucl Instrum Methods. 2004;B 213:82-86.
22. Badal A Sempau J. A package of Linux scripts for the parallelization of Monte Carlo simulations. Comput Phys Comm 2006;175:440-450.
23. Chin PW, Lewis DG, Giddy JP. Implementation of BEAMnrcMonte Carlo simulations on theGrid". Proc 14th Int Conf on the Use of Computers in Radiation Therapy, Seoul, Korea, 2004.
24. Chin PW, Giddy JP, Lewis DG, Walker DW. An Embarrassingly Parallel Framework for Running EGSnrc/BEAMnrc/DOSXYZnrc, FLUKA, MCNP/MCNPX, GEANT4, and PENELOPE on Grid and Cluster Computers", Proc 15th Int Conf on the Use of Computers in Radiotherapy, Toronto, Canada, 2007.
25. Chauvie S, Scielzo G. Radiotherapy treatment planning with Monte Carlo on a distributed system. IEEE Nuclear Science Symposium Conference Record, Vol 1, 2003;1765-1769.
26. Caccia B, Attia MM, Amati G et al. Monte Carlo in radiotherapy: experience in a distributed computational environment. J Physics: Conference Series 2007;74:012001.
27. Pratx G, Xing L. Monte Carlo simulation of photon migration in a cloud computing environment with MapReduce. J Biomed Optics 2011;16:125003, 1-9.
28. Hissoiny S, Ozell B, Bouchard H, Despres P. GPUMCD: A new GPU-oriented Monte Carlo dose calculation platform. Med Phys 2011;38:754-764.
29. Bielajew AF, Rogers DW. PRESTA: the parameter reduced electron-step transport algorithm for electron Monte Carlo transport. Nucl Instrum Methods 1987;B18:165-181.
30. Kawrakow I, Rogers DW, Mainegra-Hing E, Tessier F, Walters B. The EGSnrc Code System: Monte Carlo Simulation of Electron and Photon Transport, NRCC Report, 2010.
31. Pratx G, Xing L. Monte Carlo simulation of photon migration in a cloud computing environment with MapReduce. J Biomed Optics 2001;16:125003.
32. Hissoiny S, Ozell B, Bouchard H, Despres P. GPUMCD: A new GPU-oriented Monte Carlo dose calculation platform. Med Phys 2011;38:754-764.
33. Bielajew AF, Rogers DW. PRESTA: the parameter reduced electron-step transport algorithm for electron Monte Carlo transport. Nucl Instrum Methods 1987;B18:165-181.
34. Kawrakow I, Rogers DW, Mainegra-Hing E, Tessier F, Walters B. The EGSnrc Code System: Monte Carlo Simulation of Electron and Photon Transport. NRCC Report, 2010.
35. Rogers DW, Walters B, Kawrakow I. "BEAMnrc users manual," NRCC Report, 2009.