Phys. Med. Biol. **58** (2013) 6525–6540 doi:10.1088/0031-9155/58/18/6525

Toward a web-based real-time radiation treatment planning system in a cloud computing environment

Yong Hum Na[1,2,3], **Tae-Suk Suh**[2], **Daniel S Kapp**[1] **and Lei Xing**[1]

[1] Department of Radiation Oncology, Stanford University, Stanford, CA 94305 USA
[2] Department of Biomedical Engineering, The Catholic University of Korea, Seoul, Korea

E-mail: yhna@stanford.edu

Received 29 January 2013, in final form 12 July 2013
Published 3 September 2013
Online at stacks.iop.org/PMB/58/6525

Abstract

To exploit the potential dosimetric advantages of intensity modulated radiation therapy (IMRT) and volumetric modulated arc therapy (VMAT), an in-depth approach is required to provide efficient computing methods. This needs to incorporate clinically related organ specific constraints, Monte Carlo (MC) dose calculations, and large-scale plan optimization. This paper describes our first steps toward a web-based real-time radiation treatment planning system in a cloud computing environment (CCE). The Amazon Elastic Compute Cloud (EC2) with a master node (named m2.xlarge containing 17.1 GB of memory, two virtual cores with 3.25 EC2 Compute Units each, 420 GB of instance storage, 64-bit platform) is used as the backbone of cloud computing for dose calculation and plan optimization. The master node is able to scale the workers on an 'on-demand' basis. MC dose calculation is employed to generate accurate beamlet dose kernels by parallel tasks. The intensity modulation optimization uses total-variation regularization (TVR) and generates piecewise constant fluence maps for each initial beam direction in a distributed manner over the CCE. The optimized fluence maps are segmented into deliverable apertures. The shape of each aperture is iteratively rectified to be a sequence of arcs using the manufacture's constraints. The output plan file from the EC2 is sent to the simple storage service. Three de-identified clinical cancer treatment plans have been studied for evaluating the performance of the new planning platform with 6 MV flattening filter free beams (40×40 cm^2) from the Varian TrueBeam$^{\text{TM}}$ STx linear accelerator. A CCE leads to speed-ups of up to 14-fold for both dose kernel calculations and plan optimizations in the head and neck, lung, and prostate cancer cases considered in this study. The proposed system relies on a CCE that is able to provide an infrastructure for parallel and distributed computing. The resultant plans from the cloud computing are

[3] Author to whom any correspondence should be addressed.

identical to PC-based IMRT and VMAT plans, confirming the reliability of the cloud computing platform. This cloud computing infrastructure has been established for a radiation treatment planning. It substantially improves the speed of inverse planning and makes future on-treatment adaptive re-planning possible.

(Some figures may appear in colour only in the online journal)

1. Introduction

Intensity modulated radiation therapy (IMRT) and volumetric modulated arc therapy (VMAT) are increasingly used in radiation therapy. The complex radiation treatment planning (RTP) systems for clinical treatments demand efficient computing, such as non-deterministic Monte Carlo (MC) dose calculation and large-scale plan optimization. Parallel and distributed computing techniques have been applied to the complex computational tasks to perform calculations in a time-efficient manner. One of the major studies published using parallel computing in radiotherapy employed the application of a graphics processing unit (GPU). A GPU provides a competitive computing platform to leverage massive parallel processing in many applications of medical physics (Men *et al* 2010, Jia *et al* 2011, Peng *et al* 2012). These studies have demonstrated a speed-up factor on a GPU implementation compared with a central processing unit (CPU). Although remarkably faster implementations have been achieved in GPU usages, a few limitations exist (Pratx and Xing 2011a). First, GPU programing requires in-depth knowledge of single-instruction multiple-data architecture of the multiprocessors for an adequate data parallelism to perform high computing efficiency. Second, GPU implementation is not always able to carry out a traditional CPU oriented task. A sequential code based algorithm on a CPU is a challenge to convert into the different programming model in a GPU and requires many parameters to configure and optimize complicated memory access patterns, data access patterns, and kernel workflows. Third, GPU code extension takes more time because of the consideration of the parameters with the difficulties of debugging and maintaining the primary code.

As an alternative parallel computing method, cloud computing technology has been used in several studies (Bateman and Wood 2009, Keyes *et al* 2010, Dudley *et al* 2010, Fox 2011, Schadt *et al* 2010, Wang *et al* 2011, Meng *et al* 2011, Pratx and Xing 2011b, Philbin *et al* 2011). These have shown that the cloud computing environment (CCE) can dynamically provide scalable applications and computational resources on an on-demand basis. A cloud infrastructure contains hardware, such as networks, servers, storages components, which are necessary to support cloud service, and software to provide cloud functions such as on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service (Mell and Grance 2011).

The primary goal of cloud computing proposed in this paper is to establish and adapt a web-based treatment planning system (TPS) with the technical advantages of cloud computing. The proposed plan optimization method in CCE is an extension of our previous studies using the total-variation regularization (TVR) method (Zhu *et al* 2008, Zhu and Xing 2009, Kim *et al* 2011, 2012) for fixed gantry IMRT using more beam angles. A large number of incident beams are explicitly considered with the application of the TVR method to the VMAT planning system for a TrueBeam$^{\mathrm{TM}}$ linear accelerator (LINAC) beams (Varian Medical Systems, Palo Alto, CA) with/without the flattening filters available (Mok *et al* 2010, Cho *et al* 2011, 2013). Applying the TVR method to the proposed RTP system offers an effective interplay of planning and delivery to allow for balancing the dose conformity and efficient delivery with flattening filter

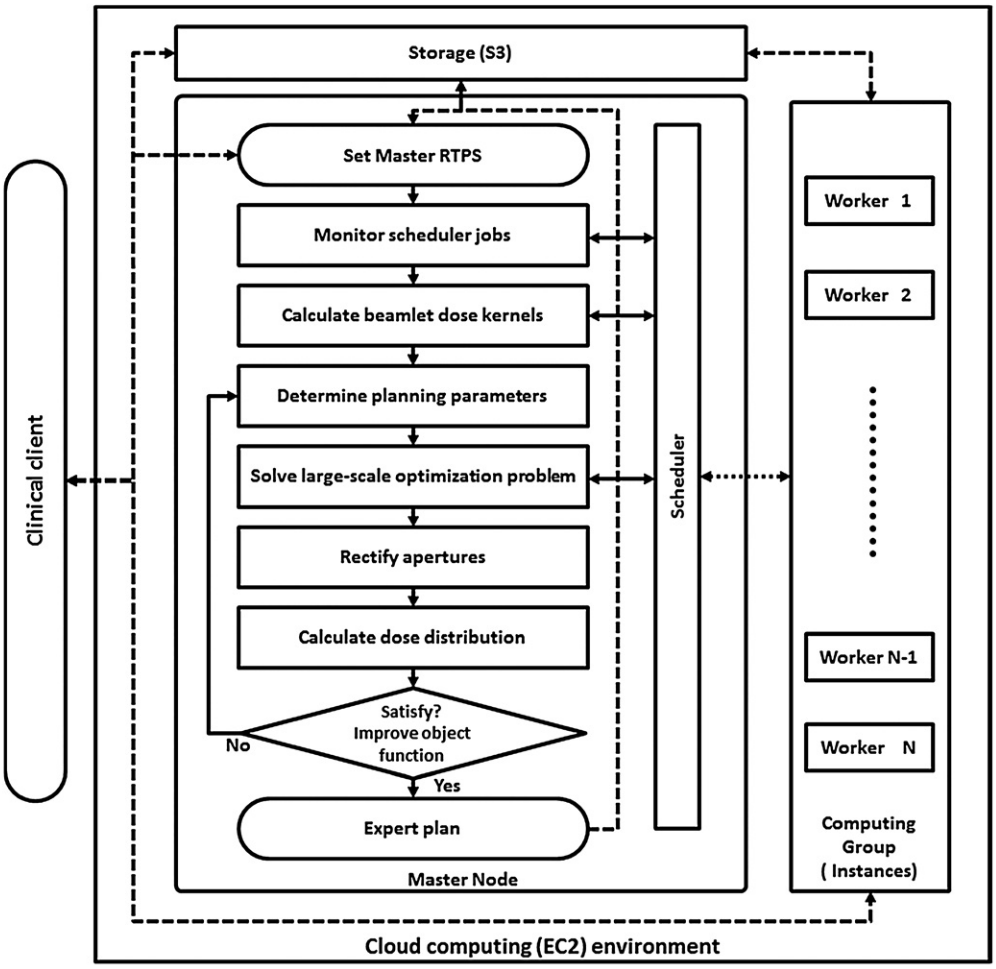

Storage (S3)
Set Master RTPS
Monitor scheduler jobs
Calculate beamlet dose kernels
Determine planning parameters
Solve large-scale optimization problem
Rectify apertures
Calculate dose distribution
Satisfy? Improve object function
No
Yes
Expert plan
Master Node
Scheduler
Clinical client
Worker 1
Worker 2
Worker N-1
Worker N
Computing Group (Instances)
Cloud computing (EC2) environment

Figure 1. Overall scheme for the TPS distributed on the cloud computing infrastructure (EC2; S3). The master schedules two major tasks, MC simulation and large-scale optimization, in parallel and distributed fashion with the workers.

free (FFF) fields. FFF beams can improve the dose delivery rate with reduction of collimator scatter, head leakages, as well as out of field dose to patients (Kry *et al* 2010, Stevens *et al* 2011). In this paper, as our first steps toward a web-based real-time radiation TPS, a detail cloud computing approach is described. Three de-identified clinical cases (Freymann *et al* 2012) of patients with head and neck, lung, and prostate cancers were used for evaluating the performance of the TPS with FFF beams.

2. Materials and methods

2.1. Cloud computing infrastructure

The overall architecture of the web-based TPS is shown in figure 1. The proposed TPS installed on Linux and running on a master node performs two major tasks: (1) non-deterministic MC simulation to generate accurate beamlet dose kernels and (2) large-scale optimization to create deliverable treatment plans. The Amazon Elastic Compute Cloud (EC2)[4] with a master node (named, m2.xlarge contains 17.1 GB of memory, two virtual cores with 3.25 EC2 compute units each, 420 GB of instance storage, 64-bit platform) is used as the backbone of cloud computing for those computations. The master is able to scale seamlessly the number of working group instances, called workers, based on the user-defined setting account for CPU usage. It can launch more workers if the CPU usage exceeds the upper limit for a given time (e.g., over 90% usage for 2 min), and stop the workers when their tasks are finished. Since the user in a clinical site can control the master node through the virtual desktop mode, the TPS on the master node can be used in the same manner as using the PC-based TPS. When the user runs the TSP on the master, it distributes in-house developed software and application packages, such as MC simulations and optimization tools, to communicate with each worker. Once the connection between the master node and all workers is confirmed, the master can create a schedule to work the two major tasks in parallel. According to the schedules, the master can send the individual task to each worker and receive the results from the workers separately. If the number of tasks is larger than the number of workers, the master reschedules the distribution to wait for the next available workers while the other workers are performing the given tasks. The results are independently sent to the master node according to the performance ability of the workers and/or the physical data transmission speed. The output plan file from the EC2 is sent to the user computer from the simple storage service (S3). The output plan file in S3 is then completely deleted from S3 in this study.

2.2. Communications and networking environments

In-house developed software that facilitates building a communication network between the user and the EC2 is implemented in MATLAB®. The internet socket is employed for the proposed TPS to communicate between the master and each worker. The socket is usually referred to as an application programming interface (API) for the transmission control protocol (TCP)/internet protocol (IP) stack. It is able to deliver the data packet to the designated thread and process it as an endpoint of a communication object connected through an IP based computer network. The primary concept of the socket is to use the transport layer with the interfaces between the application layer and the transport layer of the open systems interconnection model (Zimmermann 1980).

The master can identify and specify individual workers by their IP addresses and port numbers. The port numbers are designated to the workers through the secure shell (SSH) authorized by EC2 security groups. The SSH provides an open protocol for securing network communications over a public network. Each socket mapped by the proposed RTP system on the master is then able to communicate with a computational application process of workers. Figure 2 shows a diagram of TCP socket flow. The basic procedures are as follows: (1) Socket() creates a new socket with a designated integer number and allocates the system resources to it; (2) Bind() attaches a socket address associated with the port number and IP address of each worker; (3) Listen() keeps a listening state with the bound TCP socket to be connected with the master and workers; (4) Connect() attempts and, when available, establishes a TCP connection between the master and available workers; (5) Accept() receives an incoming attempt and blocks the caller until the TCP connection is established; (6) Send() & Receives() send and receive data through the established connection; and (7) Close() terminates the TCP connections.

[4] http://aws.amazon.com/ec2/.

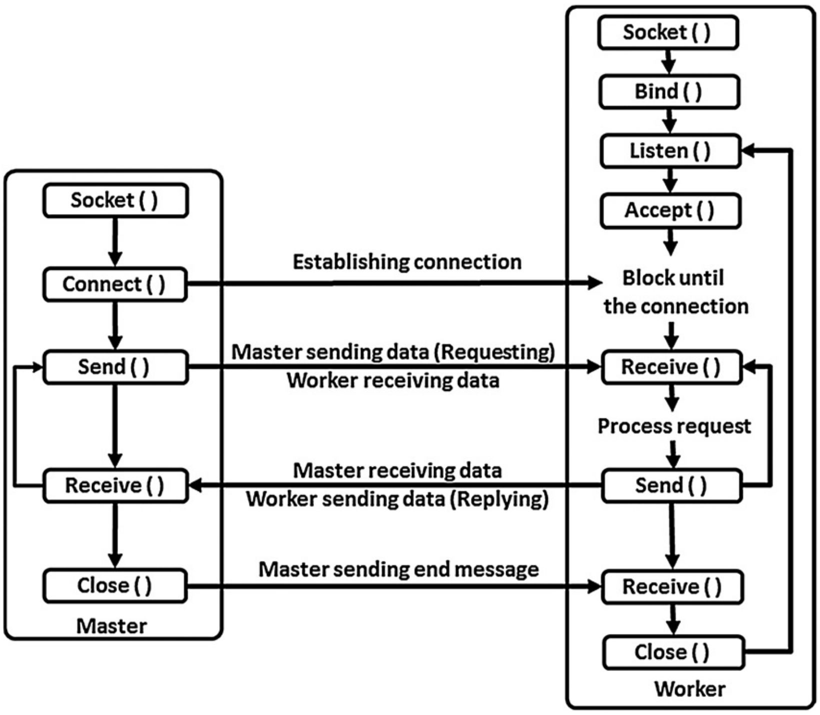


Figure 2. Diagram for TCP socket flow; TCP connection between the master node and the workers are established through the internet socket.

2.3. Dose kernel calculation

To generate accurate beamlet dose kernels by parallel tasks, the voxel-based Monte Carlo algorithm (VMC++) (Kawrakow 1997, 2001, Kawrakow and Fippel 2000) is employed. VMC++ has been validated with well-established MC simulation codes such as DOSXYZnrc and BEAMnrc (Gardner *et al* 2007, Hasenbalg *et al* 2008). VMC++ can be used for radiation transport through beam modifiers such as jaws, wedges, blocks, and multileaf collimators (MLCs) (Tillikainen and Siljamäki 2008). In this study, the beam modifier is parameterized in the multisource model as an opened ratio factor through the edge of the blocks or MLCs (Cho *et al* 2011). The output of the multisource model is used as input for the VMC++ code. VMC++ is distributed on the workers and is controlled by VMC++ input file. The file contains simulation geometry, dose scoring options, beamlet source position and edge coordinates, source spectrum, variance reduction parameter, and MC control parameter (number of particle tracks, batches to be used, initial random number seeds, cut-off energy and step size).

An in-house developed scheduler in the master node manages the workload distributing the VMC++ input file and collecting the simulation results. The distributed VMC++ on workers is set to run with the input file from the scheduler, and send back the simulation result to the master node. The scheduler supervises and administrates the computing resources connecting with the workers. The master node collects the results from all workers to update the dose kernel matrix in correct indices. Pseudocode (a) and (b) below illustrate how the dose kernel is calculated in serial and parallel fashions. The number of parallel tasks is approximately the same as the total number of beamlets in the iterative simulation. The output plan file from the EC2 is sent to the S3.

Pseudocode (a): MC dose kernel calculation in a serial fashion:

```
S1:  Set BEV of target for each beam direction
S2:      for beamlet 1 ← to $N_{\text{beamlets}}$
S3:              Set MC simulation input data: dose kernel calculation
S4:              for structure 1 ← to $N_{\text{structures}}$
S5:              Update dose kernel matrix in correct indices
S6:              end for
S7:      end for
S8:  Go for next beam direction
```

Pseudocode (b): MC dose kernel calculation in a parallel fashion:

```
P1:  Set BEV of target for each beam direction
P2:      Distribute $N_{\text{beamlets}}$ in parallel (Set MC simulation Input data: dose kernel calculation)
P3:      for structure 1 ← to $N_{\text{structures}}$
P4:              Update dose kernel matrix in correct indices
P5:      end for
P6:  Go for next beam direction
```

2.4. Inverse treatment planning optimization

Inverse treatment planning has been increasingly used to optimize the desired dose distribution to the planning target volume (PTV) and the organ at risks (OARs). The advantage of the total-variation regularization (TVR) formulated as a quadratic programming (QP) problem permits the quadratic objective function with volumetric constraints to be expressed as a function of the aperture shapes and weights of the incident beams. The TVR based inverse treatment plan optimization problem is simply described in minimize $\sum_i^N \|A_i x - d_i\|_2^2 + \beta \|\Phi x\|_1$, subject to $x \geqslant 0$, where the beamlet intensity $x \in \mathbb{R}^n$, and the beamlet kernel $A \in \mathbb{R}^{m\times n}$, and the prescribed dose $d \in \mathbb{R}^m$ with $m_1 + \cdots + m_n = m$ of structure $i = \{1, \cdots, N\}$ and total number of beamlets n. A and d are composed of $A_i \in \mathbb{R}^{m_i \times n}$ and $d_i \in \mathbb{R}^{m_i}$. β is the regularization parameter, $\Phi \in \mathbb{R}^{p\times n}$ is the difference matrix with $p = (N_u - 1) \times N_v \times N_f + N_u \times (N_v - 1) \times N_f$, N_u and N_v represent the number of MLC leaf positions and the number of pairs per field, N_f is the number of fields.

As shown in our previous IMRT plan optimization studies (Zhu *et al* 2008, Zhu and Xing 2009, Kim *et al* 2011), the TVR based plan optimization solved as a QP problem was able to efficiently provide a piecewise constant fluence map for a relatively small number of fixed beams. However, the increasing number of beams for dynamic arcs in conjunction with the additional constraints for MLC leaf motions in arc therapy forces the plan optimization formulated as a QP problem to deal with expensive and intensive computational issues. These include a requirement for a large amount of memory and a slow convergence time. The employment in this study of a sparse optimization method utilizing a sparsity-inducing complex valued l_1-minimization problem (Daubechies *et al* 2004, Combettes and Wajs 2006, Hale *et al* 2007) reduces the computational cost and time. If $f_i^1(u_i) = \|A_i u_i - d_i\|_2^2$, and $f^2(\Phi u) = \beta \|\Phi u\|_1$, the objective function of the optimization problem can be decomposed by two functions as: $\text{minimize}_{u_j \in \mathbb{R}^n} f_i^1 (A_i u_i - d_i) + f^2 (\Phi u)$, and if $\Phi u = v$, the TVR based optimization problem is able to be rewritten with local variables $u_i \in \mathbb{R}^n$ and a global variable v:

$$\text{Minimize } \sum_{i}^{N} f_i^1\,(u_i) + f^2(v), \text{ subject to } u_i = v = \Phi u, i = 1, \ldots, N, u \geqslant 0. \tag{1}$$

From the resulting alternating direction method of multipliers (ADMM) algorithm (Gabay and Mercier 1976, Boyd *et al* 2011),

$$u_i^{k+1} \in \text{ arg min}_{u_i}\ f_i^1\,(A_i u_i - d_i) + \frac{\mu}{2}\,\|\Phi u_i - v_i^k - r_i^k\|_2^2 \tag{2}$$

$$v^{k+1} \in \text{arg min}_v f^2(v) + \frac{N\mu}{2}\|\Phi\bar{u}^{k+1} - v - \bar{r}^k\|_2^2 \tag{3}$$

$$r_i^{k+1} = r_i^k - \Phi u_i^{k+1} + v^{k+1}. \tag{4}$$

The framework of distributed computing of ADMM is summarized as follows:

```
1: Set $k = 0$, $\mu > 0$, $v_0$, and $r_{0.}$
2: Distribute the initial settings to the $N$ workers
3: **repeat**
4:     **for all i = 1 . . .  . . . N in parallel do**
5:             $u_i^{k+1} \in \text{ arg min}_{u_i}\ f_i^1(A_i u_i - d_i) + \frac{\mu}{2}\|\Phi u_i - v_i^k - r_i^k\|_2^2$
6:      Send $u_i^{k+1}$ to master
7:     **end for**
8:      Collect $u_i$ and $r_i$ from $N$ workers and broadcast the update $v$
9:             $v^{k+1} \in \text{arg min}_v\ f^2(v) + \frac{N\mu}{2}\|\overline{\Phi u}^{k+1} - v - \bar{r}^k\|_2^2$
10:     **for all i = 1 . . .  . . . N in parallel do**
11:             $r_i^{k+1} = r_i^k - \Phi u_i^{k+1} + v^{k+1}$
12:     **end for**
13:     $k \leftarrow k + 1$
14: **until** acceptance of stopping criterion[a]
```

[a] The iteration k is terminated if $\frac{\left\|u^t - u^{t-1}\right\|}{\|u^t\|} \leqslant 10^{-4}$.

The quadratic problems in steps 5 and 9 are solved by proximal operators. There are many reports on proximal operators with useful software packages available (Boyd *et al* 2011, Combettes and Pesquet 2011, Parikh and Boyd 2013). For parallel processing, a fixed row splitter is used to partition the beamlet kernel A to a set of row blocks, with an assumption that there are N workers and their stores. Three de-identified clinical cases in this study used different numbers of workers: 12, 8, and 6 for head and neck, lung, and prostate cancers, respectively. The optimized fluence maps in the master are segmented into deliverable apertures. The L-1 norm regularization on the weights of the derived segments in a solution space is then used for the final solution with re-optimization. An initial arc spacing of 6° creates 60 beams directions for a single 360° arc. The shape of each aperture is iteratively rectified to be a sequencing of arcs for rotational delivery using the manufacture's constraint (Ma *et al* 2010, Bzdusek *et al* 2009). The constraint is

$$d \leqslant d_{\max} = \frac{v_{\text{leaf}}\Delta\theta}{w_{\text{gantry}}} \tag{5}$$

where, d is leaf displacement, v_{leaf} is MLC leaf travel speed, ω_{gantry} is gantry angular rotational speed, $\Delta\theta$ is the angular separation between the adjacent angle.

2.5. Evaluation

The scalability of the proposed TPS associated with the type of virtual hardware specifications for the master and worker was evaluated for head and neck, lung, and prostate cancers. The de-identified clinical data encrypted with a 256-bit Advanced Encryption Standard (AES) algorithm (NIST-FIPS 2001) are pre-uploaded to S3. After the MC dose calculation and plan optimization, the output plan file is encrypted with the same algorithm in S3 to be downloaded to the user computer. The worker was composed of a set of t1.micro instance, which has a 64-bit Linux platform, 613 MB memory with up to two EC2-compute units (to demonstrate the system performance with the basic compute unit in the EC2). The time factors in the dose calculations are compared for 50 000 photon particles per beamlet for the different number of workers. The de-identified clinical studies for head and neck, lung, and prostate used 1.6×10^8, 2.9×10^8, 4.0×10^8 particles for IMRT, and 5.6×10^9, 8.6×10^9, 1.0×10^{10} particles for VMAT, respectively. The statistical uncertainty in the voxel for doses larger than 50% of maximum dose was found to be less than 0.5% for all plans. To evaluate the quality of the treatment plans and efficiency of the planning platform of the proposed TPS, typical IMRT and VMAT plans are generated and compared for the three de-identified clinical cases.

The FFF beam profiles are generated by the multisource model (Cho *et al* 2011), and the photon spectra are generated by the spectrum model (Cho *et al* 2013). The calculated doses from the photon spectra are validated by the measured doses from 3×3 to 40×40 cm^2 field sizes at 6 and 10 MV from a Varian TrueBeam$^{\mathrm{TM}}$ STx linear accelerator (Cho *et al* 2011, 2013). The 6 MV FFF beams for a 40×40 cm^2 field size are used for all plans. The head and neck cancer IMRT plan, with 66 Gy prescribed dose to the PTV, used step and shoot beams at seven fixed gantry angles. The lung cancer IMRT plan, with 74 Gy prescribed dose to PTV, was generated at six fixed gantry angles. The prostate cancer IMRT plan, with a 78 Gy prescribed dose to the PTV, used five fixed gantry angles. For all three of the cases, the VMAT plans used a single arc with a gantry spacing of 2°. All plans were normalized to cover the 95% volume of the PTV with the prescription doses.

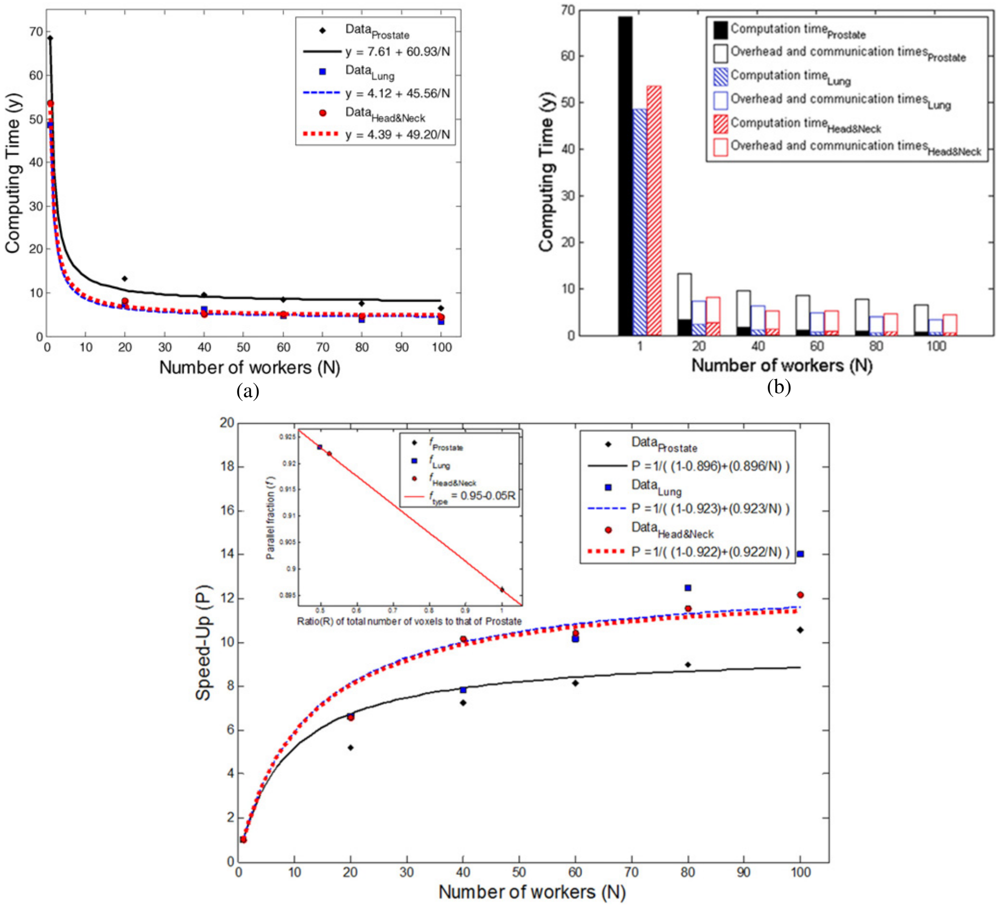
3. Results

The calculation times for beamlet dose kernels of three de-identified clinical cases, head and neck, lung, prostate plans cancers, are compared. The speed-up factors (times for calculation on single worker cloud-based system divided by times for calculation on cloud-based system with multiple workers) for each beamlet dose kernel calculation for the cases can be improved up to 12.1-fold, 14.0-fold, and 10.6-fold, respectively, based on the current basic set of CCE (using 1–100-workers). It is also observed that the computation efficiency tends to be improved as the total numbers of voxels are decreased as shown in table 1.

A reciprocal regression model, $y = a_{\text{type}} + b_{\text{type}}\,(1/N)$, is used to fit the measured data in figure 3(a), where y is the computing time, N is the number of workers or instances, a_{type} includes the times of overhead and data communication, and b_{type} represents the primary computation time for difference cases, such as head and neck, lung, and prostate cancers. The models were fit to the data with $r^2 > 0.999$ for the head and neck cancer plan, $r^2 > 0.998$ for the lung cancer plan, and $r^2 > 0.997$ for the prostate cancer plan. The overall simulation time is limited by the term of a_{type}. In a further analysis of a_{type}, the measured data were fitted using the Amdahl's law formula, $P = \frac{1}{(1-f_{\text{type}})+\frac{f_{\text{type}}}{N}}$, where P is the speed-up, N is the number of workers or instances, and parallel fraction, f_{type}, is the portion of computation which can be parallelized ($0 \leqslant f_{\text{type}} \leqslant 1$) (Amdahl 1967). To estimate pure overhead and data communication times in the cloud system, we assume that f_{type} is 1 for the completely parallelized MC dose calculation

Table 1. Calculation times of beamlet dose kernels for one field in three de-identified clinical plans of head and neck, lung, and prostate cancers.

Cancer	Field			Running time (sec)		Speed-up factor
location	size (cm)	Volume size	Resolution (mm)	$T_{1\text{-worker}}$	$T_{100\text{-worker}}$	$T_{1\text{-worker}}/T_{100\text{-worker}}$
Head neck	18 × 16	256 × 256 × 161	1.95 × 1.95 × 2.5	53.55	4.41	12.14
Lung	15 × 16	256 × 256 × 152	1.95 × 1.95 × 2.5	48.62	3.47	14.01
Prostate	12 × 13	256 × 256 × 308	1.95 × 1.95 × 2.5	68.44	6.48	10.56



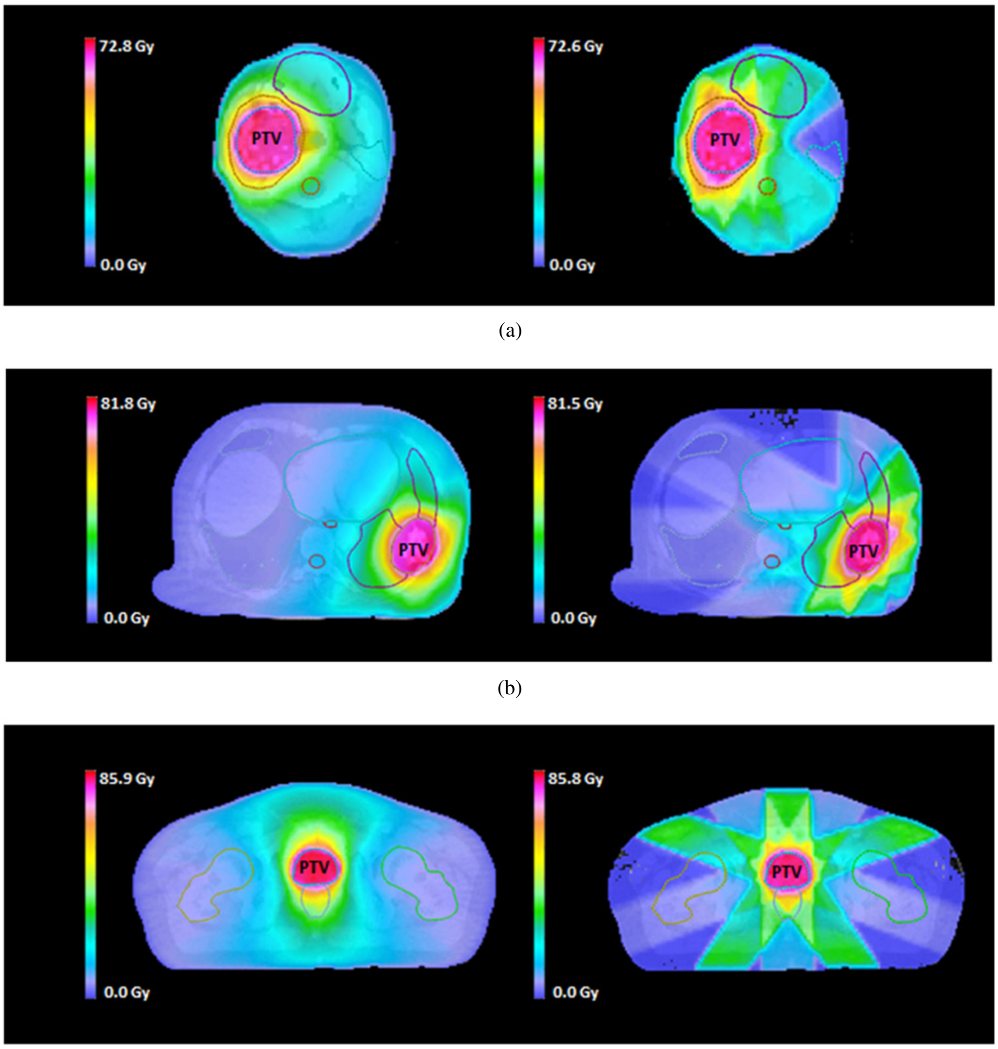
(a)

(b)

(c)

Figure 3. Computing times in (a) and (b) and speed-ups in (c) with a varying number of workers for difference cases. The computation times can be approximated by the reciprocal regression model in (a). The pure computation times are depicted as solid and hatched boxes, and the empty boxes indicated the overhead and data communication times in (b). Insert graph in (c): a plot of the parallel fractions against ratio of total number of voxels to that of the prostate. The measure data (diamonds, squares, circles) and fitted lines (solid, dashed, dotted) represent the prostate, lung, and head and neck cancer plans, respectively.

in CCE. Figure 3(b) shows the total computing times based on this assumption. One stacked bar represents the total computing time with two execution regions, such as computation time (solid and hatched boxes) and overhead and data communication times (empty boxes), with varying number of workers for each type of case. The average amount of time spent on overhead and data communication with the ideal case of $f_{\text{type}} = 1$ for all the different number of workers is 3.62 ± 1.87, 3.36 ± 1.84, 6.22 ± 3.32 s for head and neck, lung, and prostate cancer cases, respectively. According to Amdahl's Law, the speed-up as a number of workers is shown in figure 3(c). f_{type} for the three cancer treatment plans were 0.92 with $r^2 > 0.966$ for the head and neck, 0.93 with $r^2 > 0.877$ for lung, and 0.896 with $r^2 > 0.900$ for prostate cancer cases, respectively. The insert in figure 3(c) shows a linear regression model between the ratio of the total number of voxels for the prostate cancer case and f_{type}, demonstrating a strong negative correlation. Small total numbers of voxels correspond to high portions of



(a)

(b)

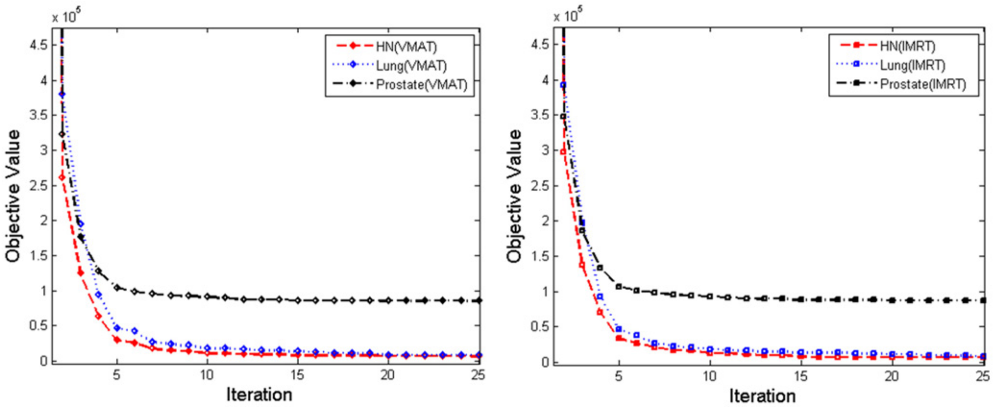
(c)

Figure 4. Dose distributions calculated for de-identified clinical patients with head and neck cancer in (a), lung cancer in (b), and prostate cancer in (c); left for VMAT and right for IMRT plans.

parallelization. The results shown in figure 3 suggest that the restrictions are linearly related to the size of the transferring data, which implies that the workload to communicate large data in CCE limits the relative improvement in performance time. Figure 4 shows the dose distributions of VMAT (left) and IMRT (right) plans for the head and neck cancer in (a), lung cancer in (b), and prostate cancer in (c). These were all obtained with the cloud system.

The optimization times for IMRT and VMAT plans for the three cases are compared in table 2. The IMRT and VMAT plan optimizations take less than a minute and 3 min respectively for the cases using the same CCE. Speed-up factors of the plan optimizations varied from 1.4-fold to 14.0-fold dependent on the specific case and plan type. Figure 5 shows the convergences of the proposed optimization with different numbers of iterations.

Note that we assume that the total computation time on both PC- and cloud-based treatment plans is calculated by the sum of the dose calculation time and plan optimization time. Although



(a) (b)

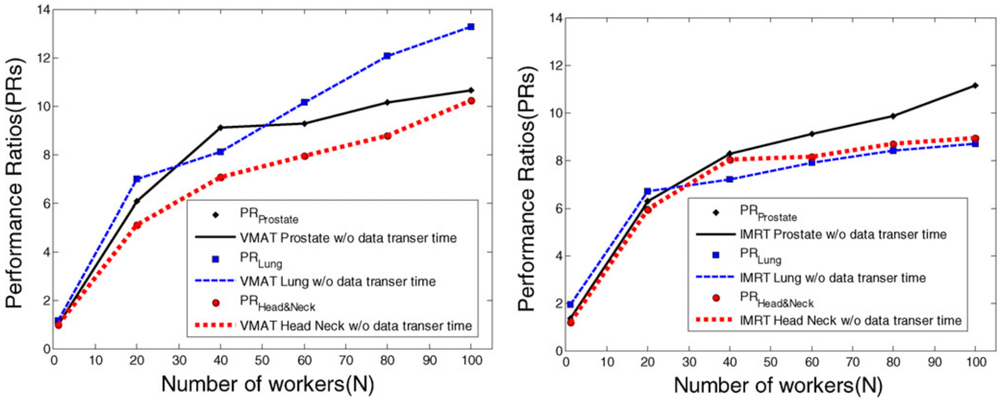
Figure 5. Convergences of (a) VMAT and (b) IMRT plans using the proposed optimization algorithm for the three different cases. The dashed, dotted, dash-dot lines represent the head and neck, lung, and prostate cancer plans, respectively.

Table 2. IMRT and VMAT optimization times comparison with three de-identified clinical cases.

		Optimization time (sec)		Speed-up factor
Cancer location	Plan type	$T_{\text{single-ADMM}}$	$T_{\text{distributed-ADMM}}$	$T_{\text{single-ADMM}}/T_{\text{distributed-ADMM}}$
Head neck	IMRT	105.22	21.27	4.95
	VMAT	154.86	110.91	1.40
Lung	IMRT	404.36	58.03	6.97
	VMAT	1858.21	147.46	12.60
Prostate	IMRT	144.84	10.35	13.99
	VMAT	245.31	24.57	9.98

the data transfer time between the user computer and S3 is not considered for the cloud-based calculation time in this study, it takes on average 46 s to upload and 29 s to download all de-identified clinical cases through the Amazon web services (AWS) management console. The average upload/download data transfer rate of the AWS management console is 17.32 Mbps/28.91 Mbps with 1Gbps network connection, which has 121.63 Mbps/233.72 Mbps upload/download Internet connection bandwidths. For the 130 Mbps network connection which has 33.65 Mbps/39.33 Mbps upload/download internet connection bandwidths, the AWS management console takes on average 80.36 s for upload and 54.44 s for download with a 9.92 Mbps/15.37 Mbps upload/download data transfer rate. As expected, the higher upload/download internet connection bandwidths can reduce the data transfer time.

The performance ratios (PRs) (i.e., PC-based calculation time divided by cloud-based calculation times with a specific number of workers) indicate the actual amount of improvement of performances. Both VMAT and IMRT plans were done for the three de-identified clinical cases using the cloud-based system described in this report and compared with the plans obtained for the identical cases using the PC-based TPS. The isodose curves for the plans done on both systems were identical (results not shown). This was to be expected since the software tools for the treatment planning and the treatment planning software used were the same for the cloud-based and PC-based systems. The time for the treatment planning was, however, shorter for both the IMRT and VMAT planning when the cloud-based system was compared with the PC-planning as summarized in table 3.



(a) (b)

Figure 6. PRs of (a) VMAT and (b) IMRT plans for the three different cases compared with different numbers of workers. The upload and download data transfer times are excluded. The dashed, dotted, dash-dot lines represent the head and neck, lung, and prostate cancer plans, respectively.

Table 3. PRs (PC-based computing/cloud-based computing) for VMAT and IMRT plans for three de-identified clinical cases.

		Performance ratios (PRs)		
Cancer location	Plan type	1-worker	40-worker	100-worker
Head neck	IMRT	1.17	8.02	8.94
	VMAT	0.99	9.09	10.63
Lung	IMRT	1.96	7.18	8.68
	VMAT	1.15	8.10	13.28
Prostate	IMRT	1.35	8.27	11.16
	VMAT	0.99	7.06	10.25

The PRs varied between the clinical sites studied. While the planning took longer for all of the VMAT plans compared with the site specific IMRT plans, the PRs were mostly better when the cloud-based system planning was compared to the PC-based system planning for the VMAT plans ($0.99 \leqslant \text{PRs} \leqslant 10.63$ for the head and neck case, $1.15 \leqslant \text{PRs} \leqslant 13.28$ for lung case, and $0.99 \leqslant \text{PRs} \leqslant 10.25$ for prostate cancer cases). However, the PRs were approximately 1 for VMAT plans when the cloud-based system was used with only 1-worker for the planning of head and neck and prostate cancers. It is also observed that the PRs of VMAT plans still tend to be increased with large variances, while those of IMRT plans begin to plateau as the total number workers are increased past 40-worker for the lung and head and neck cancers, as shown in figure 6.

4. Discussion

The cloud computing resources in terms of virtual hardware specifications for the proposed TPS are composed of a master node and a computing work group, called workers or instances. The system installed on the master node is seamlessly able to communicate with workers through the internet sockets. Two major performance objectives, MC dose kernel calculation and large-scale optimization, are efficiently distributed and computed with parallel tasks under the supervision of the scheduler on the cloud. This proof of concept study shows that the speed-up is up to 14-fold for both dose kernel calculations and plan optimizations for different plan types and clinical evaluations. The factor was estimated with a given number of fundamental computing workers, such as 1 to 100 of t1.micro instances. The number of workers is scalable on an on-demand basis with the given capacity of the Amazon EC2. The t1.micro instance in this study comes with the lowest computing price among all EC2 instances.

While, as demonstrated in this study, a cloud computing platform can improve computational performance, there are a few related issues that should be considered. First, the commercial cloud computing price is relatively high for pay-as-you-go. The Amazon EC2 computing cost depends on the type and number of instances in different regions. The price range between micro and high memory instances, such as t1.micro and m2.4xlarge varies from $0.025 to $1.840 per hour running Linux/UNIX for the US Northern California region (http://aws.amazon.com/ec2/pricing/). There is an additional charge for regional data transfers between the regions if a user wants to use instances in different regions. Currently, the cluster compute instances are not available in US Northern California, Asia Pacific, and South America regions. However, the operation in CCE is more cost-effective than those of a dedicated in-house cluster (Zhai *et al* 2011). In general, there are significant savings, on average between 20% and 50%, depending on various cloud computing migrations, privacy and security protection, file server and storage utilization, and labor costs for maintenance and management (West 2010). This study demonstrated that the proposed TPS implemented in a CCE (even with the basic computing group composed of micro instances), has improved computation efficiency, thus offering an opportunity to invest in the expansion of the computing work group involving high performance compute units. Second, although cloud service providers are still being studied concerning efficient big data transfer solutions, a network throughput should be carefully considered in a CCE as compared with a GPU shared on-chip memory. If the multiple data packages are concurrently transferred to a master node across the network to be processed, the network can become congested. To avoid such data congestion, the scheduler in this study always maintains a global first-in-first-out data queue controlling the status of jobs and availability of workers in a given network throughput. Once one job has been completed in a worker, the scheduler immediately lets the distributed application be re-initiated for the next available job without unnecessary memory usage and other interruptions. Alternatively one type of queuing system, the simple queue service provided by AWS, can be used with simple APIs at an inexpensive cost. Third, a cloud service provider has to take into account the safeguarding of patient privacy and security under the Health Insurance Portability and Accountability Act of 1996 (HIPAA) requirements subsequently expanded by the Health Information Technology for Economic and Clinical Health (HITECH) regulations. AWS provides technical information and prescriptive guidance for cloud application developers to comply with HIPAA and HITECH. In this pilot study implementing the web-based TPS, each data package used three de-identified clinical cases which were encrypted by the 256-bit AES algorithm (NIST-FIPS Standard 2001) before transmission. The package is transmitted to S3 through secure socket layer encrypted endpoints over the internet from the EC2. In order to secure access for EC2 data communication, a key pair with private and public keys created by the 2048-bit Rivest–Shamir–Adleman (RSA) algorithm (Rivest *et al* 1978) is used for unique identification. Each output plan file package in S3 from the EC2 is also encrypted by the same AES algorithm and transmitted to the user site. Such a data package of a de-identified clinical case can be transmitted to a third party organization in accordance with the agreement between the parties to use the same control mechanism.

From a technical point of view, the on-demand virtualized hardware resources along with a commercial CCE offer additional features with the distributable application packages associated with the advanced development of GPU-based computational methods in the radiotherapy community. The type of instance of computing work group in the proposed TPS can be easily updated and switched to the cluster GPU instances. The new computing work group is then able to employ such GPU-based dose calculation engines as well as GPU-based optimization algorithms through a simple modification of these methods.

5. Conclusion

This study has proposed a detailed strategy for developing a web-based TPS in a cloud computing environment (CCE). The results demonstrate that the proposed TPS provides efficient computation for dose kernel calculation and large-scale plan optimization in the cloud. The resultant plans of IMRT and VMAT plans from the cloud computing are found to be identical to those obtained using PC-based plans indicating the reliability of the cloud computing platform. The CCE substantially improves the speed of inverse planning and makes future on-treatment adaptive re-planning possible. Eventually, we plan on using the CCE approach of this study to enable interdisciplinary computing, sharing, and updating of the treatment planning system in a web-based environment.

Acknowledgments

This work was supported in part by the National Cancer Institute (1R01 CA133474) and by a Leading Foreign Research Institute Recruitment Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (MSIP) (grant no. 2009-00420).

References

Amdahl G M 1967 Validity of the single processor approach to achieving large scale computing capabilities *AFIPS Proc. Spring Joint Computer Conf.* vol 30 pp 483–5

Bateman A and Wood M 2009 Cloud computing *Bioinformatics* **25** 1475

Boyd S, Parikh N, Chu E, Peleato B and Eckstein J 2011 Distributed optimization and statistical learning via the alternating direction method of multipliers *Found. Trends Mach. Learn.* **3** 1–122

Bzdusek K, Friberger H, Eriksson K, Hårdemark B, Robinson D and Kaus M 2009 Development and evaluation of an efficient approach to volumetric arc therapy planning *Med. Phys.* **36** 2328–39

Cho W, Bush K, Mok E, Xing L and Suh T S 2013 Development of a fast and feasible spectrum modeling technique for flattening filter free beams *Med. Phys.* **40** 041721

Cho W, Kielar K N, Mok E, Xing L, Park J H, Jung W G and Suh T S 2011 Multisource modeling of flattening filter free (FFF) beam and the optimization of model parameters *Med. Phys.* **38** 1931–42

Combettes P L and Pesquet J-C 2011 Proximal splitting methods in signal processing *Fixed-Point Algorithms for Inverse Problems in Science and Engineering* vol 49 (New York, NY: Springer) pp 185–212

Combettes P L and Wajs V R 2006 Signal recovery by proximal forward-backward splitting *Multiscale Model. Simul.* **4** 1168–200

Daubechies I, Defrise M and De Mol C 2004 An iterative thresholding algorithm for linear inverse problems with a sparsity constraint *Commun. Pure Appl. Math.* **57** 1413–57

Dudley J T, Pouliot Y, Chen R, Morgan A A and Butte A J 2010 Translational bioinformatics in the cloud: an affordable alternative *Genome Med.* **2** 51

Fox A 2011 Cloud computing—what's in it for me as a scientist? *Science* **331** 406–7

Freymann J B, Kirby J S, Perry J H, Clunie D A and Jaffe C C 2012 Image data sharing for biomedical research—meeting HIPAA requirements for de-identification *J. Digit. Imaging* **25** 14–24

Gabay D and Mercier B 1976 A dual algorithm for the solution of nonlinear variational problems via finite element approximation *Comput. Math. Appl.* **2** 17–40

Gardner J, Siebers J and Kawrakow I 2007 Dose calculation validation of VMC++ for photon beams *Med. Phys.* **34** 1809–18

Hale E T, Yin W and Zhang Y 2007 A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing *CAAM Technical Report TR07-07* (Houston, TX: Rice University)

Hasenbalg F, Fix M, Born E, Mini R and Kawrakow I 2008 VMC++ versus BEAMnrc: a comparison of simulated linear accelerator heads for photon beams *Med. Phys.* **35** 1521–31

Jia X, Gu X, Graves Y J, Folkerts M and Jiang S B 2011 GPU-based fast Monte Carlo simulation for radiotherapy dose calculation *Phys. Med. Biol.* **56** 7017–31

Kawrakow I 1997 Improved modeling of multiple scattering in the voxel Monte Carlo model *Med. Phys.* **24** 505–17

Kawrakow I 2001 VMC++, electron and photon Monte Carlo calculations optimized for radiation treatment planning *Advanced Monte Carlo for Radiation Physics, Particle Transport Simulation and Applications: Proc. Monte Carlo 2000 Meeting (Lisbon, 23–26 Oct. 2000)* pp 229–36

Kawrakow I and Fippel M 2000 VMC++, a MC algorithm optimized for electron and photon beam dose calculations for RTP *Proc. 22nd Annu. Int. Conf. IEEE Engineering in Medicine and Biology Society (Piscataway, NJ)* vol 2 pp 1490–3

Keyes R, Romano C, Arnold D and Luan S 2010 Medical physics calculations in the cloud: a new paradigm for clinical computing *Med. Phys.* **37** 3272

Kim H, Suh T S, Lee R, Xing L and Li R 2012 Efficient IMRT inverse planning with a new L1-solver: template for first-order conic solver *Phys. Med. Biol.* **57** 4139

Kim T, Zhu L, Suh T S, Geneser S, Meng B and Xing L 2011 Inverse planning for IMRT with nonuniform beam profiles using total-variation regularization (TVR) *Med. Phys.* **38** 57–66

Kry S F, Vassiliev O N and Mohan R 2010 Out-of-field photon dose following removal of the flattening filter from a medical accelerator *Phys. Med. Biol.* **55** 2155–66

Ma Y, Chang D, Keall P, Xie Y, Park J Y, Suh T S and Xing L 2010 Inverse planning for four-dimensional (4D) volumetric modulated arc therapy *Med. Phys.* **37** 5627–33

Mell P and Grance T 2011 The NIST definition of cloud computing *NIST Special Publication 800-145* (National Institute of Standards and Technology) pp 1–7

Men C, Romeijn H E, Jia X and Jiang S B 2010 Ultrafast treatment plan optimization for volumetric modulated arc therapy (VMAT) *Med. Phys.* **37** 5787–91

Meng B, Pratx G and Xing L 2011 Ultrafast and scalable cone-beam CT reconstruction using MapReduce in a cloud computing environment *Med. Phys.* **38** 6603–9

Mok E, Kielar K, Hsu A, Maxim P and Xing L 2010 Dosimetric properties of flattening filter free photon beams from a new clinical accelerator *Med. Phys.* **37** 3248

NIST-FIPS Standard 2001 *Announcing the Advanced Encryption Standard (AES)* vol 197 (Gaithersburg, MD: Federal Information Processing Standards Publication)

Parikh N and Boyd S 2013 *Proximal Algorithms Found. Trends Optimization* **1** 123–231 (at press)

Peng F, Jia X, Gu X, Epelman M A, Romeijn H E and Jiang S B 2012 A new column-generation-based algorithm for VMAT treatment plan optimization *Phys. Med. Biol.* **57** 4569

Philbin J, Prior F and Nagy P 2011 Will the next generation of PACS be sitting on a cloud? *J. Digit. Imaging* **24** 179–83

Pratx G and Xing L 2011a GPU computing in medical physics: a review *Med. Phys.* **38** 2685–97

Pratx G and Xing L 2011b Monte Carlo simulation of photon migration in a cloud computing environment with MapReduce *Biomed. Opt.* **16** 125003

Rivest R L, Shamir A and Adleman L 1978 A method for obtaining digital signatures and public-key cryptosystems *Commun. ACM* **21** 120–6

Schadt E E, Linderman M D, Sorenson J, Lee L and Nolan G P 2010 Computational solutions to large-scale data management and analysis *Nature Rev. Genet.* **11** 647–57

Stevens S, Rosser K and Bedford J 2011 A 4 MV flattening filter-free beam: commissioning and application to conformal therapy and volumetric modulated arc therapy *Phys. Med. Biol.* **56** 3809–24

Tillikainen L and Siljamäki S 2008 A multiple-source photon beam model and its commissioning process for VMC++ Monte Carlo code *J. Phys.: Conf. Ser.* **102** 012024

Wang H, Ma Y, Pratx G and Xing L 2011 Toward real-time Monte Carlo simulation using a commercial cloud computing infrastructure *Phys. Med. Biol.* **56** N175–81

West D M 2010 Saving money through cloud computing *Brookings Institution paper* (Washington, DC: Brookings Institution) (available at www.brookings.edu/research/papers/2010/04/07-cloud-computing-west)

Zhai Y, Liu M, Zhai J, Ma X and Chen W 2011 Cloud versus in-house cluster: evaluating Amazon cluster compute instances for running MPI applications *SC'11: Proc. Int. Conf. on High Performance Computing, Networking, Storage and Analysis (SC), Seattle, WA, 12–18 November 2011* pp 1–10

Zhu L, Lee L, Ma Y, Ye Y, Mazzeo R and Xing L 2008 Using total-variation regularization for intensity modulated radiation therapy inverse planning with field-specific numbers of segments *Phys. Med. Biol.* **53** 6653–72

Zhu L and Xing L 2009 Search for IMRT inverse plans with piecewise constant fluence maps using compressed sensing techniques *Med. Phys.* **36** 1895–905

Zimmermann H 1980 OSI reference model–The ISO model of architecture for open systems interconnection *IEEE Trans. Commun.* **28** 425–32