**manuscript No.**
(will be inserted by the editor)

# Radiotherapy Monte Carlo simulation using cloud computing technology

**C M Poole, I Cornelius, J V Trapp,
C M Langton**

**Abstract** Cloud computing allows for vast computational resources to be leveraged quickly and easily in bursts as and when required. Here we describe a technique that allows for Monte Carlo radiotherapy dose calculations to be performed using GEANT4 and executed in the cloud, with relative simulation cost and completion time evaluated as a function of machine count. As expected, simulation completion time decreases as $1/n$ for $n$ parallel machines, and relative simulation cost is found to be optimal where $n$ is a factor of the total simulation time in hours. Using the technique, we demonstrate the potential usefulness of cloud computing as a solution for rapid Monte Carlo simulation for radiotherapy dose calculation without the need for dedicated local computer hardware as a proof of principal.

**Keywords** cloud computing, Monte Carlo, GEANT4, radiotherapy

## 1 Introduction

Significant computational overhead prevents the routine use of Monte Carlo simulation applied to radiotherapy problems in the clinical setting, generally as a consequence of limited access to suitable computing hardware. The advent of cloud computing however provides a low cost and easy to maintain alternative to the setup of dedicated computing hardware in the clinic [18, 19]. Indeed, several authors have explored the usefulness of "the cloud" for Monte Carlo simulation [7,10,14,23], the most notable of which uses Fluka [11] for proton beam dose calculations on the Amazon Elastic Compute Cloud (EC2, Amazon Web Services LLC, USA) [19]. This work too uses EC2 as the host cloud computing platform, however Geometry and Tracking 4 (GEANT4) has been selected to simulate a clinical radiotherapy linear accelerator. Here we aim to show the immediate capability of the cloud for the purpose of radiotherapy Monte Carlo simulation whilst within the clinical environment.

C. M. Poole is with Cancer Care Services, Royal Brisbane and Womens Hospital, Herston, QLD 4029, Australia. E-mail: christopher.poole@qut.edu.au · C. M. Poole, I. Cornelius, J. V. Trapp and C. M. Langton are with the Discipline of Physics, Faculty of Science and Technology, and the Institute of Health and Biomedical Innovation, Queensland University of Technology, Brisbane, QLD 4000, Australia.

Cloud computing is the use of virtual and remote computer hardware for the purposes of scalable service provision such as high demand web-hosting with volatile loading conditions and scientific computation problems requiring significant and variable computer or memory resources [4,16,26]. This sounds very much like the typical function of super computer clusters where a fixed number nodes may be configured to perform many possible functions, and resources are allocated based on job priority. The cloud computing paradigm, however, differs from this fix-resource model by enabling the number of nodes in the cluster to expand and contract dynamically based on demand. This is accomplished with the use of nodes, or instances, that are usually configured and booted once to perform a single task before being shutdown again; ready to accept a new configuration. The advantage of this is that many similar nodes can be launched simultaneously to perform a single task in parallel. As these instances are virtual, the configuration is not limited to software. Hardware can be configured as well, such as the amount of RAM, CPU power and disk storage space for example.

In the following we describe the process of executing a pre-existing GEANT4 simulation of a clinical linear accelerator [8] on the Amazon EC2 computing resource. The method has application outside of radiotherapy and is not restricted to AWS, however a radiotherapy linear accelerator Monte Carlo simulation executed using AWS is used here as proof of principal.

## 2 Methods

GEANT4 is a C++ toolkit for the simulation of particle transport through geometry and is the Monte Carlo toolkit selected to carry out this work. It is used widely in the field of high energy physics [2] and sees increasing adoption for radiotherapy treatment simulation [2, 5, 13, 17, 20, 24]. Flexible geometry definition and physics process customisation provides the user with a high level of control, and the opportunity to simulate a wide range of radiotherapy techniques including brachytherapy, hadron therapy and intensity modulated radiotherapy [3]. Additionally, as it is a developers toolkit, it is an ideal platform for experimenting with new parallelization techniques and simulation hosts such as cloud computing platforms.

For this study, EC2 provides scalable computing instances and the Amazon simple storage solution (S3) provides the off-instance data storage area. Compute capability of a particular AWS instance type is described using the “EC2 compute unit”, where one compute unit is the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor [1]. At creation, any EC2 instance may have custom user data parsed to it, the user data itself may take on any form whether it be binary, ASCII or otherwise - subsequently this user data may be used to uniquely configure running tasks on the instances. Two modes of access are available to the users, on-demand instance creation billed at a fixed hourly rate, or a variable rate where the user may bid for unused instances with the time to availability varying depending on current demand – this is known as the spot market. Access to the resources provided by AWS cloud services such as EC2 and S3 and the cloud services provided by other vendors can be performed programatically using the *boto* Python module [12].

## 2.1 AWS Instance Set-up

A single instance of type *t1.micro* was launched using the pre-built and official Ubuntu 10.04 LTS 64 bit Amazon Machine Image (AMI) with identifier *ami-3202f25b*. The boot process itself was similar to the normal boot process for a default install of any recent version of the Ubuntu server distribution [25]. Unlike a conventional local install however, the *libcloud* [6] package was installed by default on the AMI enabling access to user data parsed to the instance at the time of creation. GEANT4 version 9.3 and its dependencies were compiled and installed, and the instance was then saved as a custom AMI using the menu options available in the AWS dashboard. Once, saved, this custom AMI was available to boot up to 20 instances with the default AWS account set-up. In the case of booting 20 High CPU Extra Large EC2 instances, 160 CPU cores were made available to the user with a total compute capability of 400 EC2 units.

## 2.2 Distributing Jobs in the Cloud

A Monte Carlo model of Varian Clinac was commissioned for dose calculation as described elsewhere [8]. Along with a Python (Python Software Foundation, USA) [21] interface to the simulation, the *boto* Python cloud computing module was used to automatically distribute jobs to the cloud environment from the local user machine. A job launcher was created that managed the packing of a job description and data into a compressed archive and the launching of a group instances, see figure 1. For a given job, the simulation configuration included a manifest of all files and folders to be included as job data. Using the *tarfile* Python module, part of the Python standard library [21], each file or folder in the manifest was added to an archive, followed by compression and writing to disk. From the local user machine, the compressed job archive was uploaded to S3 one time per unique simulation using *boto*. An EC2 reservation was requested which launched the prescribed number of instances for the job; a process fully managed by the *boto* Python module and EC2. Each instance had user data containing the simulation configuration including the location of the job archive on S3 transmitted to it automatically.

At instance boot time, a Python script was automatically executed, recovering the simulation configuration from the pre-transmitted user data. A pool of worker processes with a pool size equal to the number of processor cores available on the instance was then created using the *multiprocessing* Python module [21], see figure 2. This worker pool enabled a simulation described in a Python function to be executed multiple times and concurrently across a number of processes equal to the pool size. On each instance, the master process managing the pool of worker processes, waited for all workers to finish execution, and subsequently combined and compressed the results returned by each worker process. Finally, the compressed result was uploaded to S3 to a location specified in the simulation configuration and the instance was terminated as soon as possible, thus minimising the potential of cost escalation. Retrieving results from S3 could be performed using the AWS dashboard and a web-browser. For execution of instances on the spot market, a maximum bid price could be specified at the time of reservation and configured as a parameter along with all other simulation parameters. From the user perspective, there was no difference between a instance acquired on-demand or bid for on the spot market.

### 2.3 Benchmarking Performance and Relative Cost

High CPU Extra Large EC2 instances were chosen for all jobs executed in the cloud as they provided the highest on-demand compute density for absolute cost. A series of test simulations were performed so as to examine simulation performance as a function of EC2 instance count. Using the GEANT4 geometry primitive *G4Box*, a 40 $cm$ cubic water phantom was defined and positioned with its center at the iso-center of the linear accelerator; with a 100 $cm$ source to axis distance (SAD) or 80 $cm$ source to surface distance (SSD). Irradiated with a jaw defined $5 \times 5$ $cm$ field and gantry and primary collimator angles set to zero, $2.5 \times 10^6$ electrons incident on the copper target in the linear accelerator treatment head were simulated. The simulation was repeated for a range of EC2 instance counts ($1 \leq n \leq 20$) on the spot market with simulation completion time (the time elapsed from starting a job to uploading a result to S3), instance uptime, total simulation time (the total real CPU time used) and total simulation cost were recorded. On-demand instance cost was calculated from the billed instance hours multiplied by the on-demand rate for the High CPU Extra Large instance type and compared to the actual cost incurred as a result of simulating the above using instances on the spot market.

## 3 Results

### 3.1 Simulation Output

Figure 3 shows typical output for the simulation described in section 2.3 using a 2 $mm$ scoring dose grid. All dose values are shown normalised to the maximum central axis dose. The size in memory for the entire dose grid with $200 \times 200 \times 200$ voxels using single precision floating point values was 32 $MB$ per worker process for a total of 256 $MB$ per instance.

### 3.2 Compute Performance

For the simulation described in section 2.3 the average time from instance boot to the start of the simulation on the same node was $59 \pm 1s$. Figure 4(a) shows the simulation completion time $t_c$ as a function of instance count; it was found to follow

$$t_c = \frac{t_s}{n_i n_p}, \tag{1}$$

where $t_s$ is the total simulation time required, $n_i \in \mathbb{N}^\star = \{1, 2, 3, \dots, 20\}$ is the number of instances used per job and $n_p \in \mathbb{N}^\star = \{1, 2, 3, \dots, 8\}$ is the number of processors available per instance. Noting that the default AWS accounts allowed for a maximum of $n_i = 20$ instances, and the maximum number of processors available per instances was $n_p = 8$ as of August 2012 [1]. Total simulation time or the total real CPU time consumed for the simulation as a function of instance count is shown in figure 4(b). Mean total simulation time required for the simulation described in section 2.3 was $t_s = 26.1 \pm 0.2$ *hours* where the uncertainty represents one standard deviation about the mean.

### 3.3 Relative Usage Costs

Where the instance count was greater than the simulation completion time in hours, cost escalation was linear with increasing instance count, see figure 5. Billable instances hours required to complete a given job requiring $t_s$ total compute hours were found to follow

$$t_i = n_i \left\lceil \frac{t_s}{n_i n_p} \right\rceil = n_i \left\lceil t_c \right\rceil , \tag{2}$$

where $t_i \in \mathbb{N}^\star = \{1, 2, 3, \ldots\}$ is the total billable instance hours and $\lceil \ldots \rceil$ indicates the ceiling function, noting that the uptime of a given instance was rounded up to the nearest hour for the purposes of billing. Simulations running at least total cost were found where the simulation time in hours was wholly divisible by the total number of instances running for that job, corresponding to the factors of $\lceil t_s/n_i \rceil \in \mathbb{N}^\star = \{1, 2, 3, \ldots\}$.

## 4 Discussion & Conclusion

Using a GEANT4 simulation of a clinical linear accelerator, executed on the Amazon Elastic Compute Cloud, we have demonstrated the potential usefulness of cloud computing for rapid radiotherapy dose calculation. Additionally, a simple formulation allowing for the optimal selection of instance count for least cost has been proposed, given some estimate of total simulation time required. Figure 4(a) shows simulation time decreasing as $1/n$ with increasing instance count as observed by others [19], cost however increases linearly with increasing instance count when simulation time in hours is less than the instance count, as shown in figure 4(b). For a given simulation, if time is not a critical factor, the number of instances used can be tuned for least cost by ensuring each instance is in use for whole hours, as Amazon EC2 instances charges are not prorated for partial instance hour usage. However, in an environment where time is critical, increasing instance count reduces simulation time with a linearly increasing cost penalty. At the user simulation level, this cloud based computing option is no different to current distributed computing technologies, and we find its performance to be suitable enough for application to more complex dose calculations such intensity modulated radiotherapy, and those applied to full CT datasets. The wide range of EC2 instance types and configurations available should allow for its use in any field of scientific computing where large amounts of CPU time and/or RAM are required.

Application of this technique enables a GEANT4 user to perform a simulation in a distributed compute environment, with a low entry cost and no express need for dedicated compute hardware. Whilst we note that the absolute cost of 20 × 8 CPU core EC2 instances used continuously for 12 months would be sufficient (approximated based on 2012 prices) to purchase and operate an equivalent local computer cluster, this cloud based solution is almost free of ongoing system administration. It is also dynamically scalable based on current demand, and the user does not need to consider ongoing hardware maintenance and upgrades; such activities are performed transparently by AWS in this case. Furthermore, it is not unreasonable to expect benefits of future hardware innovations will be pass on to users either via lower usage costs or high performance EC2 instance options.

For clinics in developing countries for example, which may not have sufficient resources to provide adequate cancer care [15] much less manage dedicated compute hardware, this may be of particular benefit. Indeed, the shortfall in the quality of cancer care in developing countries has been identified by others [15, 22], in particular the relationship between inadequate staff training and suboptimal treatment delivery [22]. Systems to remedy this have been proposed by others, and of particular note is the Hospital Platform for E-health (HOPE) [9] enabling the remote verification of radiotherapy treatment plans and other diagnostic and therapeutic tests. Adoption of initiatives such as HOPE, coupled with the computational resources provided by the cloud and the simulation techniques described here within may offer significant scientific and social benefit.

Presently this work is part of a software toolkit using GEANT4 for the simulation of clinical linear accelerators [8]. Source code for running GEANT4 simulations on EC2 as described here is freely available and may be obtained from: `http://code.google.com/p/manysim/`

The authors declare that they have no conflict of interest, and have no affiliation with Amazon Web Services.

**Acknowledgements** This work is funded by the Queensland Cancer Physics Collaborative, and Cancer Australia (Department of Health and Ageing) Research Grant 614217.

## References

1. Amazon EC2 Instance Types. Amazon Web Services LLC. http://aws.amazon.com/ec2/instance-types/ (2011)
2. Agostinelli, S., Allison, J., Amako, K., Apostolakis, J., Araujo, H., Arce, P., Asai, M., Axen, D., Banerjee, S., Barrand, G., et al.: Geant4 - a simulation toolkit. Nuclear Instruments and Methods in Physics Research-Section A Only **506**(3), 250–303 (2003)
3. Allison, J., Amako, K., Apostolakis, J., Araujo, H., Dubois, P., Asai, M., Barrand, G., Capra, R., Chauvie, S., Chytracek, R., et al.: Geant4 developments and applications. Nuclear Science, IEEE Transactions on **53**(1), 270–278 (2006)
4. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al.: A view of cloud computing. Communications of the ACM **53**(4), 50–58 (2010)
5. Caccia, B., Andenna, C., Cirrone, G.A.P.: MedLinac2: a GEANT4 based software package for radiotherapy. Annali dell'Istituto superiore di sanità **46**, 173–177 (2010)
6. Computer software. http://ci.apache.org/projects/libcloud/apidocs/: Libcloud: a unified interface to the cloud, v0.4.2 edn. (2011)
7. Constantin, M., Sawkey, D., Mansfield, S., Svatos, M.: Su-e-e-05: The compute cloud, a massive computing resource for patient-independent monte carlo dose calculations and other medical physics applications. Medical Physics **38**, 3392 (2011)
8. Cornelius, I., Hill, B., Middlebrook, N., Poole, C., Oborn, B., Langton, C.: Commissioning of a Geant4 based treatment plan simulation tool: linac model and DICOM-RT interface. Arxiv preprint arXiv:1104.5082 (2011)
9. Diarena, M., Nowak, S., Boire, J., Bloch, V., Donnarieix, D., Fessy, A., Grenier, B., Irthum, B., Legré, Y., Maigne, L., et al.: HOPE, an open platform for medical data management on the grid. Studies in health technology and informatics **138**, 34 (2008)

10. Farbin, A.: Emerging Computing Technologies in High Energy Physics. Arxiv preprint arXiv:0910.3440 (2009)
11. Ferrari, A., Sala, P., Fasso, A., Ranft, J.: Fluka. CERN-library in: http://fluka. web. cern. ch/fluka (2005)
12. Garnaat, M., et al.: Boto Python interface to Amazon Web Services Documentation. Computer software. http://code.google.com/p/boto/, v2.0 edn. (2010)
13. Grevillot, L., Frisson, T., Maneval, D., Zahra, N., Badel, J.N., Sarrut, D.: Simulation of a 6 MV Elekta Precise Linac photon beam using GATE/GEANT4. Physics in Medicine and Biology **56**, 903 (2011)
14. Gruntorad, J., Lokajicek, M.: International Conference on Computing in High Energy and Nuclear Physics (CHEP'09). In: Journal of Physics: Conference Series, vol. 219, p. 001001 (2010)
15. Hanna, T., Kangolle, A.: Cancer control in developing countries: using health data and health services research to measure and improve access, quality and efficiency. BMC International Health and Human Rights **10**(1), 24 (2010)
16. Hayes, B.: Cloud computing. Communications of the ACM **51**(7) (2008)
17. Jan, S., Benoit, D., Becheva, E., Carlier, T., Cassol, F., Descourt, P., Frisson, T., Grevillot, L., Guigues, L., Maigne, L., et al.: GATE V6: a major enhancement of the GATE simulation platform enabling modelling of CT and radiotherapy. Physics in Medicine and Biology **56**, 881 (2011)
18. Keyes, R., Romano, C., Arnold, D., Luan, S.: Cloud Computing as a Monte Carlo Cluster for Radiation Therapy. In: Proceedings of the XVIth International Conference on the Use of Computers in Radiation Therapy (ICCR) (2010)
19. Keyes, R., Romano, C., Arnold, D., Luan, S.: Radiation therapy calculations using an on-demand virtual cluster via cloud computing. Arxiv preprint arXiv:1009.5282 (2010)
20. Rodrigues, P., Trindade, A., Peralta, L., Alves, C., Chaves, A., Lopes, M.C.: Application of GEANT4 radiation transport toolkit to dose calculations in anthropomorphic phantoms. Applied Radiation and Isotopes **61**(6), 1451–1461 (2004)
21. van Rossum, G., Drake, F.L.: Python Reference Manual. Python Software Foundation. http://python.org/, v2.7.1 edn. (2011)
22. Shakespeare, T., Back, M., Lu, J., Lee, K., Mukherjee, R.: External audit of clinical practice and medical decision making in a new Asian oncology center: results and implications for both developing and developed nations. International Journal of Radiation Oncology* Biology* Physics **64**(3), 941–947 (2006)
23. Silverman, A., Fedorko, I., Lapka, W., Lo Presti, G.: CHEP 2010 Report. CHEP - Computing in High Energy and nuclear Physics. Tech. Rep. CERN-IT-Note-2010-007, CERN, Geneva (2010)
24. Spezi, E., Lewis, G.: An overview of Monte Carlo treatment planning for radiotherapy. Radiation protection dosimetry (2008)
25. Ubuntu Community Documentation. https://help.ubuntu.com/community/EC2StartersGuide: Ubuntu EC2 Starters Guide (2011)
26. Vouk, M.: Cloud computing–issues, research and implementations. Journal of Computing and Information Technology **16**(4), 235–246 (2004)

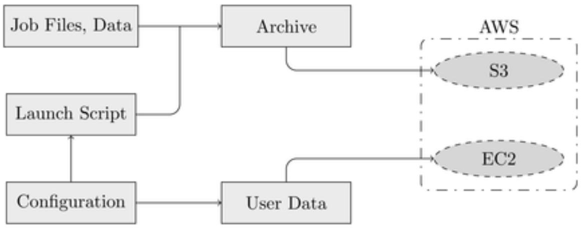


**Fig. 1** Launching EC2 instances from the local user machine. The launch script takes a configuration and parses it as user data to a pool of EC2 instances, and compresses the job files for upload to S3.

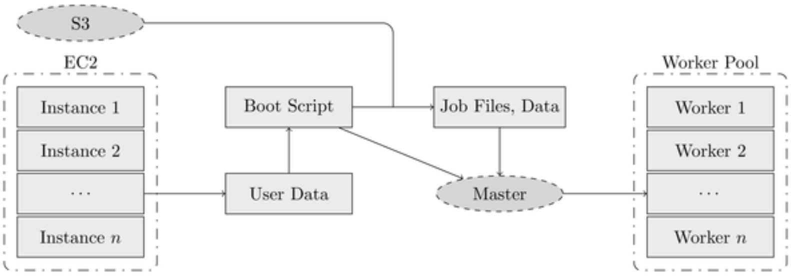
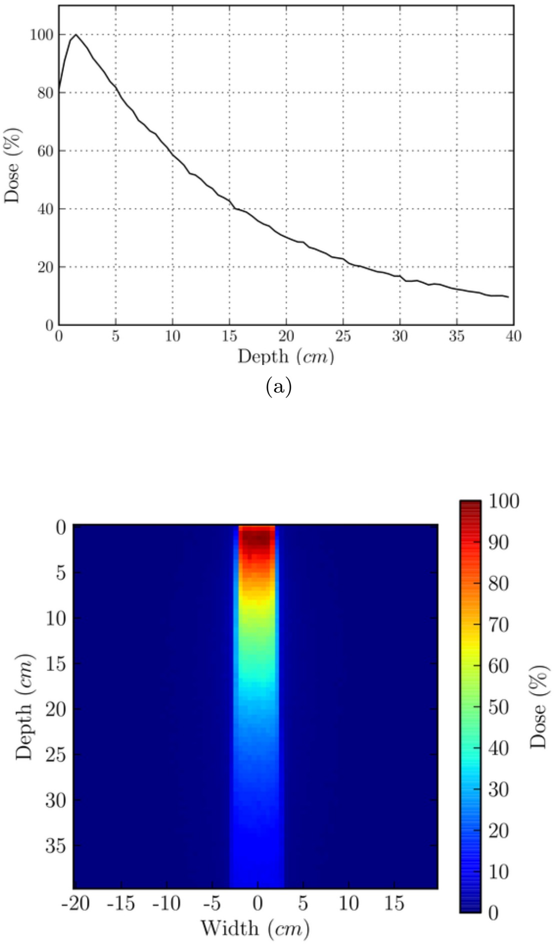


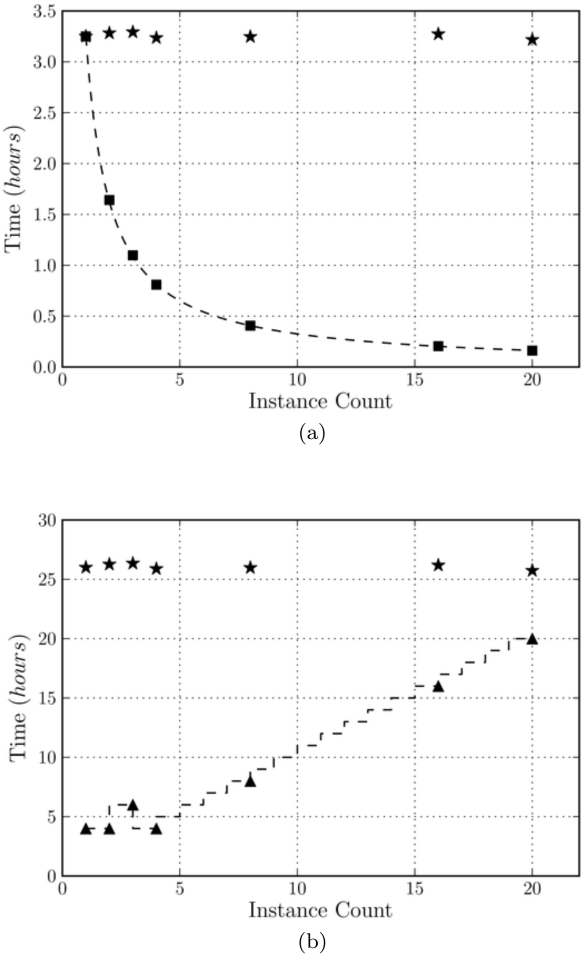
**Fig. 2** Simulation configuration and worker pool creation on each EC2 instance. On each EC2 instance, user data is parsed to a boost script which downloads the jobs files from S3 and launches a master process which subsequently creates a pool of worker processes.



(a)

(b)

**Fig. 3** Simulation output; (a) shows the central axis depth dose and (b) shows the dose distribution of the central slice in the water phantom. Note that the iso-center of the simulated linear accelerator was positioned at (0, 20) in (b).



(a)

(b)

**Fig. 4** Simulation time (a) where stars indicate the total instance up-time, squares indicate the time to simulation completion and the dashed line indicates the predicted simulation completion time (equation 1); marker size indicates 2 standard deviations about the mean, $R^2 = 0.97$. Billable instance time (b) as a function of instance count where stars indicate the total compute required, triangles indicate the billable instance time, and the dashed line indicates the predicted billable instance time (equation 2).

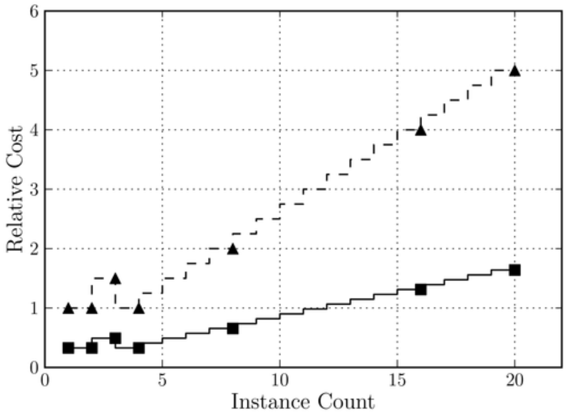


**Fig. 5** Simulation cost as a function of instance count where squares indicate the incurred cost as a result of bidding for Amazon EC2 High CPU Extra Large instances on the spot market (0.223 USD/hour), triangles indicate the equivalent cost had the on-demand rate of 0.68 USD/hour been charged, and the solid and dashed lines indicate the predicted instances hours (equation 2) multiplied by the hourly rate.