



**HAL**  
open science

## Enabling Large Scale Data Production for OpenDose with GATE on the EGI Infrastructure

Maxime Chauvin, Gilles Mathieu, Sorina Camarasu-Pop, Axel Bonnet,  
Manuel Bardiès, Isabelle Perseil

► **To cite this version:**

Maxime Chauvin, Gilles Mathieu, Sorina Camarasu-Pop, Axel Bonnet, Manuel Bardiès, et al.. Enabling Large Scale Data Production for OpenDose with GATE on the EGI Infrastructure. 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), May 2019, Larnaca, France. pp.658-665, 10.1109/CCGRID.2019.00084 . inserm-02869860

**HAL Id: inserm-02869860**

**<https://inserm.hal.science/inserm-02869860v1>**

Submitted on 16 Jun 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Enabling Large Scale Data Production for OpenDose with GATE on the EGI Infrastructure

Maxime Chauvin\*, Gilles Mathieu†, Sorina Camarasu-Pop‡, Axel Bonnet‡, Manuel Bardiès\* and Isabelle Perseil†

\*CRCT, UMR 1037, Inserm, Université Toulouse III Paul Sabatier, Toulouse, France

†Inserm, DSI, Coordination de l'Informatique Scientifique, Paris, France

‡CREATIS, CNRS UMR5220, Inserm U1044, INSA-Lyon, Université Lyon 1, Lyon, France

**Abstract**—The OpenDose collaboration has been established to generate an open and traceable reference database of dosimetric data for nuclear medicine, using a variety of Monte Carlo codes. The amount of data to generate requires to run tens of thousands of simulations per anthropomorphic model, for a total computation time estimated to millions of CPU hours. To tackle this challenge, a project has been initiated to enable large scale data production with the Monte Carlo code GATE. Within this project, CRCT, Inserm CISI and CREATIS worked on developing solutions to run Gate simulations on the EGI grid infrastructure using existing tools such as VIP and GateLab. Those developments include a new GATE grid application deployed on VIP, modifications to the existing GateLab application, and the development of a client code using a REST API for using both. Developed tools have already allowed running 30% of GATE simulations for the first 2 models (adult male and adult female). On-going and future work includes improvements both to code and submission strategies, documentation and packaging of the code, definition and implementation of a long-term storage strategy, extension to other models, and generalisation of the tools to the other Monte Carlo codes used within the OpenDose collaboration.

**Index Terms**—Dosimetry, Nuclear medicine, Radiotherapy, Monte Carlo methods, Numerical simulation, Grid computing, Distributed computing, GATE, EGI, France Grilles.

## I. INTRODUCTION

OpenDose [1] is a collaborative effort to develop an open and traceable reference database proposing dosimetric data applicable in a context of nuclear medicine dosimetry. Such data is essential when performing cancer diagnostic or therapy since it allows assessing the irradiation delivered to the patient organs after the administration of radioactive material (radiopharmaceutical).

The generation of dosimetric data involves radiation transport modelling and energy deposition scoring in anthropomorphic models, usually based on Monte Carlo simulation. Historically, such data was computed from mathematical models - simplistic approximations to human geometry. The advent of voxel-based computational models requires a new appraisal of dosimetric data. For example, the models recently proposed by the International Commission on Radiation Protection (ICRP) include 140 distinct tissues/organs, leading to more than 20,000 source/target possible combinations [2]. The production of data for these 2 models (adult male and adult female) for all possible source regions, radiation types and

energies represents an important computation time (millions of CPU hours).

The OpenDose collaboration was initiated to address this challenge by sharing computing resources and expertise in various Monte Carlo codes. This enables independent production and verification of data with different algorithms and physical process models. At the time of writing the collaboration includes 14 research teams from 18 institutes with expertise in 5 of the most popular Monte Carlo codes used in nuclear medicine dosimetry: EGSnrc/EGS++[3], Fluka[4], Geant4/GATE[5], MCNP/MCNPX[6] and PENELOPE[7]. Some of these codes are under proprietary license (Fluka, MCNP/MCNPX, PENELOPE), and some are open source under GPL based licenses (EGSnrc/EGS++, Geant4/GATE).

Toulouse Cancer Research Centre (CRCT) is one of the teams contributing to OpenDose data production, using Geant4/GATE. Locally available computing resources consist of a cluster of 240 cores. This is already not quite sufficient to generate data in a reasonable time for the two ICRP adult models; and this does not scale up well considering additional models. This paper describes approaches, workflows and on-going developments to take advantage of existing tools and computing resources on the EGI grid infrastructure<sup>1</sup> through its French component France Grilles<sup>2</sup>.

## II. COMPUTING ASPECTS

### A. GATE Simulations for OpenDose

Calculating dosimetric data requires to know the fraction of energy received by a given organ from a single nuclear decay in the source. Each simulation consists in calculating radiation transport in anthropomorphic models for specific parameters, namely: source organ, particle type, energy, model used, and number of events (primaries) to simulate.

Producing data for OpenDose with GATE is straightforward and doesn't need compilation. GATE requires only one macro file and input data to setup the simulation. Input data includes the anthropomorphic model as a 3D image and the corresponding material database in a specific text format. The macro file contains a list of commands to setup the geometry from the input data, the physics list (Livermore) and energy cuts (0.1

<sup>1</sup><http://www.egi.eu/>

<sup>2</sup><http://www.france-grilles.fr/>

mm), the pseudo random generator (Mersenne twister) and the output data formats. The source type (electrons or photons), location (voxel regions ID), energy (in MeV), seed and number of particles to generate are defined as aliases in the macro file. The aliases values are set at runtime by giving command line arguments to GATE using a specific flag.

Every simulation produces binary (3D matrices) and ASCII files. The 3D matrices contain energy deposited per voxels, and ASCII files contain pre-processed data corresponding to energy deposited per regions such as organs and tissues. These raw outputs are later processed into dosimetric data such as Specific Absorbed Fractions (SAFs) and S-values [8].

Each simulation being completely independent from the others, generating OpenDose data is a classical "embarrassingly parallel" problem.

### B. Technical Challenge

Reference models used by OpenDose currently consist of 140 modelled organs leading to 172 possible source regions [2]. The collaboration initially aims at producing reference data for 2 models (adult male and adult female), 2 particle types (electrons and gamma photons) and 91 energies (logarithmic distribution ranging from 0.005 MeV to 10 MeV). Generating the data then requires running a total of 62,608 simulations, each of them consisting of running GATE using a macro file and input data that describe the model and specify the value of appropriate parameters.

To ensure high quality results and keep a reasonable execution time, the number of events to be simulated (primaries) has been set to  $10^8$  per simulation. Simulations with the lowest energies are the quickest to run and take about 30 minutes to complete on an average machine, while those with the highest energies take many hours. Estimated average is around 24h. As a result, running all simulations would require around 1.5 M CPU hours.

This is the main challenge the collaboration has to face, as such a large amount of computing resources is not readily available in any of the participating team, so as to ensure a reasonable time to produce all data. As an example, running all simulations on CRCT's 240-cores cluster would require more than 260 days of exclusive use.

Another important aspect is the amount of generated data. A single simulation produces approximately 150 MB of outputs. While in some cases compression can highly reduce the size of the output archive, storing and handling uncompressed data remains an important challenge.

### C. Resources Adequacy Study: Cluster vs Grid

Data production for OpenDose is clearly of a classical embarrassingly parallel type. Additionally, although input data size is rather small, those calculations will produce a consequent amount of output data. Clusters or distributed computing infrastructures then appear as the most adapted solutions to run this data production work.

The CRCT team in Toulouse has access to a locally deployed and maintained cluster of 240 cores. As an academic research team both affiliated to Toulouse University and Inserm,

Table I  
CLUSTER VS GRID COMPARISON

Criteria	Local cluster	EGI grid infrastructure
Scalability	Limited	Easy
Performances and production planning	General homogeneity. Performances and speed up are limited and depend on local resources.	General heterogeneity. Possible long waiting times, but many more resources available.
Software availability	Locally maintained environment	Deployment needed, but many solutions already available
Reproducibility	Easy to ensure	Requires additional work
Financial costs for the end user	Potentially high	Very low

the team has also access to the EGI infrastructure through its French component France Grilles.

Comparing both solutions based on the criteria presented in Table I show the advantages of the grid solution in our context, where scalability and cost are the most important aspects to consider.

## III. AVAILABLE TOOLS

### A. DIRAC

The DIRAC interware [9] is currently the most widely used scheduler to submit jobs to the EGI Infrastructure. DIRAC offers all needed functions to encapsulate OpenDose simulations and submit them to the grid.

Using DIRAC commands in local scripts requires installing and configuring the DIRAC client and can only be launched with proper grid credentials, namely a X509 personal grid certificate properly registered to an EGI recognised Virtual Organisation. Before launching simulations, the proper GATE release needs to be available from a grid storage node, so that it can be downloaded and unpacked by the job itself on the computing node. The overall application also needs to be encapsulated, so that the environment is properly set up on the computing nodes at runtime.

### B. VIP and GateLab

The Virtual Imaging Platform (VIP<sup>3</sup>) [10] is a web portal for medical simulation and image data analysis. It leverages resources available in the biomed Virtual Organisation of the EGI e-infrastructure to offer an open service to academic researchers worldwide. In January 2019, VIP had more than 1000 registered users, some 20 applications and 44 publications made by VIP users with results computed with VIP. VIP uses the French national instance of the DIRAC service<sup>4</sup> for job submission.

One of the most successful VIP applications is the GateLab [11]. The GateLab is a MOTEUR[12] workflow allowing to

<sup>3</sup><https://vip.creatis.insa-lyon.fr/>

<sup>4</sup><https://dirac.france-grilles.fr/DIRAC>

execute GATE simulations using a split and merge approach through a simple and user-friendly interface. The workflow mainly consists of (i) a computing part, where partial simulation results are produced by Monte-Carlo jobs, and (ii) a merging part, where these results are assembled together. The computing phase can be done using either a static or a dynamic load balancing method. The static method divides the number of events equally among the multiple distributed jobs. With the dynamic method, each Monte-Carlo job contributes to the simulation uninterruptedly until it is stopped by the workflow engine. This second method generally greatly improves the load balancing and reduces the waiting time for results retrieval. Nevertheless, it has the disadvantage of not being reproducible, since it does not guarantee a given number of events for a given job.

### C. Advantages and Limitations

DIRAC offers all low-level tools and mechanisms to launch OpenDose simulations on the grid. However, implementing a full workflow can be burdensome, since it would require developing and orchestrating different modules on client side. On the other hand, using VIP and GateLab for OpenDose simulations would offer many benefits. First, the tool provides a well tested, readily available workflow for GATE simulations. Then, it implements mechanisms to optimise Gate simulations on a grid environment. The amount of work is thus expected to be significantly reduced and the resulting solution more efficient, compared to a DIRAC based, custom developed solution.

Yet, GateLab cannot be used "out of the box" in this context. As described earlier, OpenDose simulations invoke GATE using aliases to allow for the use of the same input macro file for different simulations. This is not implemented in available versions of GateLab, where a single macro file is expected for every single simulation. Additionally, the project aims at automating the OpenDose production through scripts and command line tools, while GateLab was primarily designed as a web-based tool. Those two considerations are the main factor behind the approach described in next sections.

## IV. AN OPENDOSE APPLICATION ON THE GRID

### A. Boutiques

Boutiques<sup>5</sup> [13] is a cross-platform descriptive command-line framework for applications. It relies on Linux containers to facilitate application installation and sharing, and it uses a versatile JSON format to describe the command-line template, inputs and outputs. Boutiques descriptors are intended to be produced by scientific application developers and consumed by execution platforms such as VIP.

While the Boutiques framework enables a descriptive representation of command-line utilities and informs users of their usage, Boutiques Python tool provides users with a variety of operations on these descriptors. Users can thus easily create a

new descriptor, validate and launch it, as well as publish it on open repositories such as Zenodo<sup>6</sup>.

### B. The GateCLforOpenDose Application

The fact that GateLab did not allow for using aliases first led to head for deploying a specific GATE workflow on VIP using Boutiques. This was achieved by defining a JSON descriptor with a command line that includes (i) the download and unpacking of needed GATE release, (ii) the setup of environment variables, (iii) the GATE call itself including aliases, and (iv) the packaging of outputs. Aliases to be passed to GATE are defined in the descriptor as parameters.

Additionally, the application implements a convenient naming convention for its outputs. Each simulation returns a tar.gz archive whose filename includes used values for the particle type, energy and number of primaries parameters.

A first version of the GateCLforOpenDose application was successfully deployed on VIP in October 2018. A few adaptations and optimisations followed, including the deployment of the GATE release to be used on the CernVM-FS service [14] available to the biomed Virtual Organisation. Latest version has been deployed early December 2018. The resulting GateCLforOpenDose application descriptor is available on GitHub<sup>7</sup> and Zenodo<sup>8</sup>.

The "input data" parameter of the application points to a single archive, uploaded through the VIP file transfer system to grid Storage Nodes, and containing input data and macro files for all simulations.

### C. Execution Times Limitations

With a working application deployed and usable on the grid, the next task was to determine the best submission strategy. One of the key aspects for that was to study execution times of the jobs that VIP would submit to the infrastructure through the GateCLforOpenDose workflow.

From a physical perspective, it can be assumed that the more the simulated particle interacts with its environment, the longer the simulation will run. To characterise execution time across the overall production, taken approach was to use the particle energy as a variable, for a given source organ, since energy is the main factor that influences the number of particle interactions. 515 simulations were run with one organ source (the liver) for gammas and electrons from 5 keV to 10 MeV (91 energies). The mean execution time per energy of these simulations is shown on Figure 1.

It is clear from those results that in both cases execution time does indeed grow with energy. While the progression is rather linear for gamma particles, it appears to be exponential for electrons. Additionally, great differences in execution times between simulations with the same input parameters have been observed. These differences, expressed in relative time compared to the mean value, are shown on Figure 2. This

<sup>5</sup><https://github.com/boutiques/boutiques>

<sup>6</sup><https://zenodo.org/>

<sup>7</sup><https://github.com/CISI-INSERM/OpenDose-Computing/>

<sup>8</sup><https://doi.org/10.5281/zenodo.2573071>

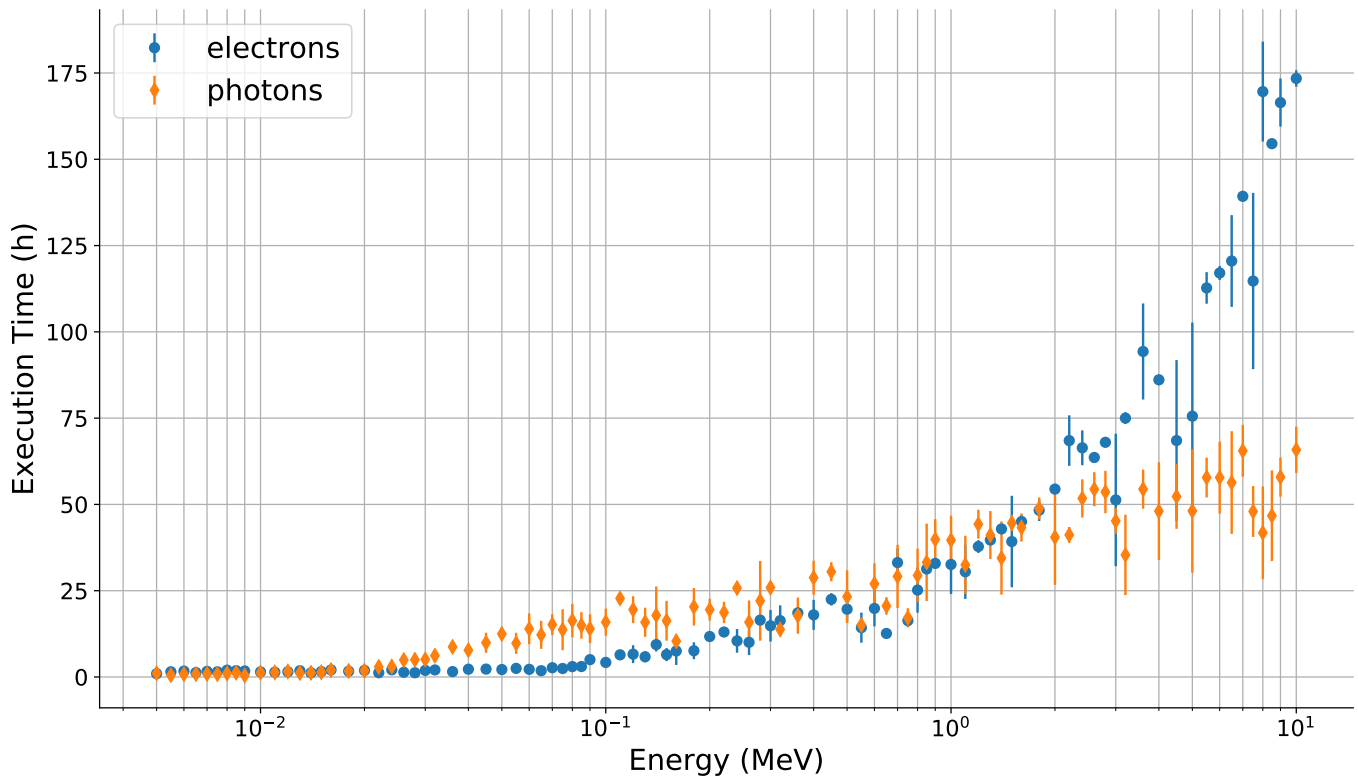


Figure 1. Execution time function of the particle energy for  $10^8$  generated particles (electrons in blue dots, photons in orange diamonds). Data points over 0.75 MeV are extrapolated from execution time with  $10^7$  generated particles. The error bars correspond to one standard deviation.

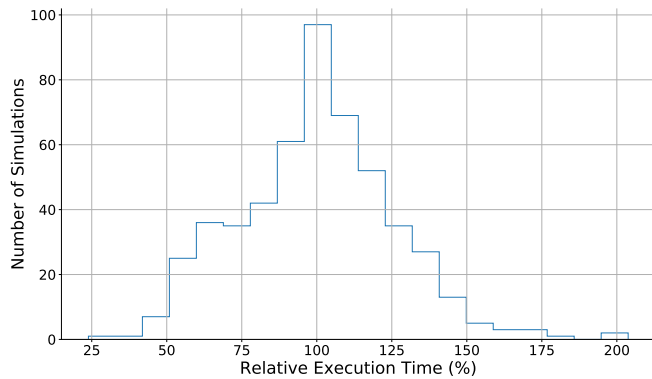


Figure 2. Relative execution time of 515 simulations to the mean execution time per energy.

highlights the high heterogeneity of grid resources used, since for the same simulation this can vary up to 200%.

Such high variations in execution times from one simulation to another is likely to create difficulties in a production environment. While shorter jobs would run without much problems, it is expected that the ones lasting more than 24h would risk getting stalled or killed at some point, thus dramatically reducing performances and success rates of the whole production.

As a consequence, using the GateCLforOpenDose applica-

tion seems only practical for roughly a third of the production, namely simulations for particles with energies lower than 0.06 MeV which, according to tests, generally terminate within a few hours.

Finally, these observed results allowed improving the initial estimation of the overall computation time needed for the project. Data shows that simulating both electrons and photons at all energies for one source organ requires a little less than 5,000 CPU hours. The overall CPU time needed to run these simulations for 172 sources on 2 models amounts to nearly 1.7M CPU.hours, which is about 10% more than the initially estimated 1.5M CPU hours.

#### D. Modifying GateLab Workflow

The existing GateLab workflow has the major advantage of handling the splitting of computational intensive simulations in multiple jobs in a transparent manner for the end user. Nevertheless, it was designed to meet the needs of end-users behind a graphical user interface, limiting as much as possible the number of requested inputs. The user only needs to provide the main macro file, which is automatically parsed in order to generate the necessary parameters for launching the GATE simulation with a split and merge approach.

In order to meet the needs of OpenDose, the GateLab workflow had to become more flexible, allowing to (i) specify a set of additional parameters on the command line and (ii) better handle the pseudo random generator seeds for the

multiple jobs of one GATE simulation, so that reproducibility is ensured.

In order to allow for additional parameters on the command line, an additional optional String input has been created. It allows specifying a number of parameters as GATE aliases, such as the number of particles, the source and the particle type and energy, which were previously hard written in the macro files. This new optional input is only available when using the API and hidden when using the GateLab through the VIP portal, in order to keep it as simple as possible for the end users.

As a Monte Carlo code, GATE needs an initial seed allowing for the generation of random numbers. Users can set the seed themselves in the configuration (macro) file, or can ask GATE to generate it using a seed generator by setting it on "auto". Until now, the GateLab enforced the "auto" mode, allowing to automatically handle the seeds for the multiple GATE jobs. However, when using the "auto" mode, it is difficult to perfectly reproduce the results if different seeds are selected and therefore traceability is not insured. Moreover, as indicated in this web article<sup>9</sup>, in order to avoid duplicate seeds when parallelizing a big simulation by spreading out multiple copies, it is safer to seed each process with sequential seeds (the seeds don't have to be random, they just have to be unique). Therefore the GateLab has been modified in order to allow for the setting of a unique seed in the macro file, which is then incremented with the number corresponding to the job id (which is unique for a given simulation).

## V. A CLIENT FOR THE APPLICATION

### A. The CARMIN API

CARMIN [15] is an open-source web API specification for platforms offering computing and/or storage services. It allows other services to use these functionalities in a programmatic way and to integrate them easily. CARMIN aims at being supported on as many platforms as possible to standardise these accesses and to improve the interoperability across executing and data services.

VIP is the leading platform concerning the CARMIN support, and its users can use this API to launch new pipeline executions, to monitor them, to provide input data and to retrieve the results.

### B. The OpenDose Client

An OpenDose Client was first implemented as a JAVA class, using the CARMIN API functions to set up and launch executions on the EGI infrastructure through VIP. It was first developed to submit workflows using the GateCLforOpenDose application, and was subsequently extended to allow for submissions using GateLab as well.

GateCLforOpenDose submits only one grid job per simulation but these jobs can be launched in batches; the split and merge approach used in GateLab results however in many grid

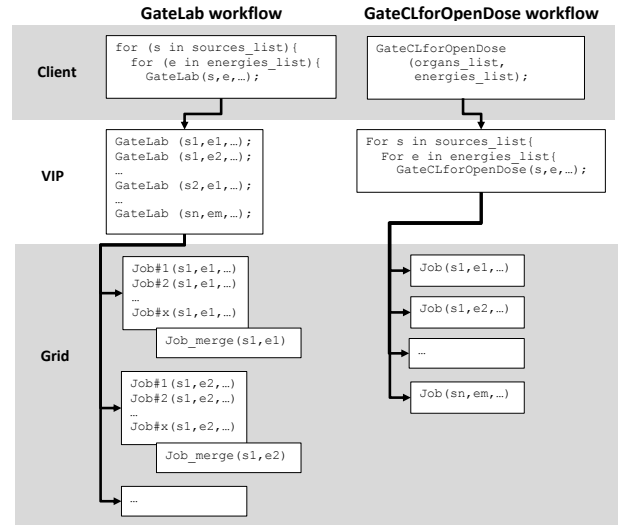


Figure 3. Diagram of the overall GateLab and GateCLforOpenDose workflows

jobs being launched for a single simulation. Client code thus has to reflect this difference in terms of submission patterns.

The code calling GateCLforOpenDose passes some of the parameters (namely, the particle energy and source organ) as lists, and the workflow manager on VIP side goes through these lists to submit as many simulations as there are combinations. This approach is especially useful with regards to handling outputs, because all output files of a given batch execution will be placed in the same directory, thus facilitating results gathering. While this was already possible through VIP web interface where a single workflow can run many jobs by combining different parameters values, the behaviour was not available in CARMIN. This required a slight modification of the API code to allow for lists to be taken as inputs.

The code calling GateLab, on the other hand, actually implements loops that go through all parameters. Many workflow executions can then be launched through one single client call, each of them corresponding to a batch of "sub-tasks" for a single simulation. The two different workflows are shown on Figure 3. The client code uses a local configuration file to set up all parameters for the simulation, and define which application to call (GateLab or GateCLforOpenDose).

Beyond making all necessary calls to actually submit simulations to the grid through VIP, the client also logs all executions with date, used parameters and the workflow identifiers returned by VIP through CARMIN. While these are not used at the moment, this is an anticipation of further developments on results gathering that will require tracking all submissions.

Client code, as well as examples of configuration files, are open and available from the VIPclient directory of the project's GitHub repository<sup>10</sup>.

<sup>9</sup><https://www.johndcook.com/blog/2016/01/29/random-number-generator-seed-mistakes>

<sup>10</sup><https://github.com/CISI-INSERM/OpenDose-Computing/>

## VI. RESULTS

First production results have been generated with the GateCLforOpenDose application. Between December 2018 and January 2019, 20,640 simulations have successfully run on the EGI infrastructure, representing 57,804 CPU.hours. These simulations correspond to both electron and gamma particles, for the adult male and adult female models, for the first 30 particle energies (ranging from 0.005 to 0.06 Mev). While this accounts for 33% of the overall production in terms of number of simulations to run, it only represents approximately 3.4% of overall estimated computation time.

Those simulations have been launched through the OpenDose JAVA client in 2 waves (5,580 in December 2018 and 14,880 in January 2019), in batches of 360 jobs (each job corresponding to one single simulation), Up to 5 batches at a time. Measured speedup factor is 124 for the December 2018 wave and 118 for January 2019.

Individual job execution times are consistent with the tests ran before launching production and that are described on Figure 1, with an average of about 2 hours per electron simulation and 4 hours per gamma simulation. It was observed, especially in the case of gamma simulations, that a few jobs took up to 50 hours to complete, highlighting the necessity to use GateLab's split and merge approach for the rest of the production, where simulations are expected to take even longer on average. Moreover, it is expected that the split and merge approach will also result in a consequent improvement with regards to the overall speedup. This is of particular interest considering that simulations remaining to run represent more than 96% of the overall estimated computation time.

Great differences can also be observed in waiting times between the different production batch executions. The mean waiting time calculated for each batch of 360 jobs ranges from 8 minutes to 33 hours. This clearly shows the opportunistic use of the grid. While this makes time predicting quite non-deterministic, it has little impact on the final speed-up since this can somehow be counterbalanced by an efficient submission strategy to maximise the number of running jobs at a given time.

## VII. DISCUSSION

### A. Further Improvements to the Client Code

In its current state, the OpenDose client submits workflows to the EGI infrastructure through VIP but does not automatically get the outputs. Output files currently have to be downloaded manually through VIP file transfer system. One first obvious improvement to the tool would be to implement a mechanism to retrieve job outputs when they are available. This retrieve function of the OpenDose client would make use of the CARMIN API to (i) scrutinise the output directory at regular intervals, (ii) download all new files, and (iii) update a reference configuration file with downloaded filename on success, so that the file is not retrieved again next time the function runs. Alternatively, the retrieve output function could

make use of the GFAL2<sup>11</sup> library that allows for mounting grid folders locally, thus facilitating synchronisation.

The first production results have been generated through the GateCLforOpenDose workflow. Next production stages will however require the use of the modified GateLab workflow. This highlights the fact that, depending on the range of simulations to run, using GateLab and the split and merge approach can either be a necessity, or an overhead. Another possible improvement would then be to implement a client side mechanism that takes estimated execution times for simulations to run, and uses this to determine (i) which of the 2 applications to use and (ii) in case GateLab is selected, set how many sub-jobs the simulation has to be split into.

### B. Result Storage Strategy and Solutions

The default storage system in VIP uses the distributed Storage Elements (SEs) provided by different EGI sites supporting the biomed VO. This system allows to distribute and replicate files on multiple SEs, which can be very useful for managing the inputs and outputs of computing jobs. Nevertheless, it is less adapted for long term storage or data sharing with users not belonging to the biomed VO.

One possible solution would be an external storage system, such as the iRODS<sup>12</sup>[16] storage infrastructure provided by France-Grilles. The Integrated Rule-Oriented Data System (iRODS) is an open source data management software used by research organisations and government agencies worldwide. iRODS should be soon integrated within VIP, similarly to the Girder<sup>13</sup> Data Management Platform.

### C. Future Use and Studies

The CRCT team in Toulouse recently earned access to 600,000 CPU hours at CALMIP<sup>14</sup>, a regional High Performance Computing Centre with a performance of 1.365 PFLOPS. Some of this time will be used to produce OpenDose data which will allow to make some comparisons with EGI. The performance on CALMIP HPC is expected to be higher than EGI but the overall effective computation time, ease of access and available tools will be assessed.

Another aspect worth studying in the scope of the OpenDose collaboration is data generation using other available Monte Carlo softwares. This would require additional work on porting those software: while it is possible to implement a new application similar to GateCLforOpenDose but using EGSnrc/EGS++ for example, such an application would face a similar challenge when it comes to execution times, and an equivalent to GateLab's split and merge approach would have to be developed.

### D. Tool Packaging and Diffusion

While already developed components are already available online on platforms such as GitHub and Zenodo, the fully

<sup>11</sup><https://dmc.web.cern.ch/projects/gfal-2/home>

<sup>12</sup><https://irods.org>

<sup>13</sup><https://github.com/girder/girder>

<sup>14</sup><https://www.calmip.univ-toulouse.fr/>

implemented set of tools will be properly published and documented to ensure its reusability within and outside of the OpenDose collaboration.

## VIII. CONCLUSION

In the context of OpenDose, a large number of Monte Carlo simulations have to be run to build a reference dosimetric database for nuclear medicine dosimetry. The CRCT, Inserm CISI and CREATIS have developed tools to enable large data production with GATE on the EGI grid infrastructure.

A new GATE application (GateCLforOpenDose) has been deployed on VIP using Boutiques and is available as a service executing on the computing resources of the biomed Virtual Organisation. Using this application for the first set of simulations led to the observation of large computing performance heterogeneity on the grid (up to a factor 8) and very long execution time for some jobs (up to 175 h). This result motivated the adaptation to our needs of the existing GateLab application, which has the advantage of handling the split and merge strategy for long simulations. A client has been developed in JAVA using the CARMIN API to use both the new GateCLforOpenDose and GateLab. This client handles the submission strategy, and logs all execution dates, parameters and VIP variables to ensure traceability and reproducibility.

At time of writing, 30% of all simulations for the initial 2 models considered have successfully been executed on the grid using GateCLforOpenDose. On the other hand, the remaining simulations to run represents 96% of the estimated total computation time. Using the updated GateLab with a split and merge strategy is expected to significantly improve the overall speedup on the grid.

Once the data production for the initial models (ICRP adults) is complete, the project will extend to future models such as paediatric phantoms. Developed workflows and tools will ensure these new calculations benefit from the full scalability potential of used infrastructures. Production constraints may lead to new ways of running simulations, gathering results or ensuring reproducibility. Yet the work already done shows that used tools have enough flexibility to allow for future improvements and adaptations, in particular to other Monte Carlo codes within the OpenDose collaboration. Future evolution will likely improve quality on the technical ground, but also enlarge usage of developed tools to benefit the wider user community.

## ACKNOWLEDGMENT

The authors acknowledge the support of France Grilles for providing computing resources and services on the French National Grid Infrastructure.

This work was achieved using the biomed Virtual Organisation of the EGI infrastructure, with the dedicated support of resource centres BEIJING-LCG2, IN2P3-IRES, OBSPM, INFN-FERRARA, GARR-01-DIR, INFN-CATANIA, INFN-ROMA3, INFN-BARI, CREATIS-INSALYON, NCG-INGRID-PT, INFN-PISA, CESNET-MetaCloud and CLOUFIN, resource centres in UK hosted by GridPP

collaboration, and the additional support of the resource centres listed here: <http://operations-portal.egi.eu/vapor/resources/GL2Browser?VOfilter=biomed>

## REFERENCES

- [1] M. Chauvin, D. Borys, F. Botta, P. Bzowski, M. A. C. Pérez, M. Cremonesi, J. Dabin, A. M. Denis-Bacelar, A. Desbrée, Z. E. Bitar, N. Falzone, L. Ferrer, D. Franck, N. Lanconelli, A. Mairani, A. Malaroda, K. Matusik, E. McKay, M. Pacilio, J. Pieter, A. P. Robinson, J. L. Rodríguez, L. Struelens, L. A. T. Aroches, A. V. Gil, and M. Bardiès, "Opendose: A collaborative effort to produce reference dosimetric data with monte carlo simulation software," *Physica Medica*, vol. 42, pp. 32–33, oct 2017. [Online]. Available: <https://doi.org/10.1016/j.ejmp.2017.09.081>
- [2] ICRP, "ICRP Publication 110. Adult Reference Computational Phantoms," *Annals of the ICRP*, vol. 39, no. 2, pp. 1–164, 4 2009.
- [3] I. Kawrakow, "Accurate condensed history Monte Carlo simulation of electron transport. I. EGSnrc, the new EGS4 version," *Medical Physics*, vol. 27, pp. 485–498, Mar. 2000.
- [4] T. Böhlen, F. Cerutti, M. Chin, A. Fassò, A. Ferrari, P. Ortega, A. Mairani, P. Sala, G. Smirnov, and V. Vlachoudis, "The fluka code: Developments and challenges for high energy and medical applications," *Nuclear Data Sheets*, vol. 120, pp. 211 – 214, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0090375214005018>
- [5] D. Sarrut, M. Bardiès, N. Bousson, N. Freud, S. Jan, J.-M. Létang, G. Loudos, L. Maigne, S. Marcatili, T. Mauxion, P. Papadimitroulas, Y. Perrot, U. Pietrzyk, C. Robert, D. R. Schaart, D. Visvikis, and I. Buvat, "A review of the use and potential of the gate monte carlo simulation code for radiation therapy and dosimetry applications," *Medical Physics*, vol. 41, no. 6Part1, p. 064301, 2014. [Online]. Available: <https://doi.org/10.1118/1.4871617>
- [6] L. S. Waters, G. W. McKinney, J. W. Durkee, M. L. Fensin, J. S. Hendricks, M. R. James, R. C. Johns, and D. B. Pelowitz, "The mcnp monte carlo radiation transport code," *AIP Conference Proceedings*, vol. 896, no. 1, pp. 81–90, 2007. [Online]. Available: <https://aip.scitation.org/doi/abs/10.1063/1.2720459>
- [7] J. Baró, J. Sempau, J. Fernández-Varea, and F. Salvat, "Penelope: An algorithm for monte carlo simulation of the penetration and energy loss of electrons and positrons in matter," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 100, no. 1, pp. 31 – 46, 1995. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0168583X95003495>
- [8] K. Eckerman and A. Endo, *MIRD: Radionuclide Data and Decay Schemes, 2nd Edition*. Society of Nuclear Medicine, 2008.
- [9] A. Tsaregorodtsev, M. Bargiotti, N. Brook, A. C. Ramo, G. Castellani, P. Charpentier, C. Cioffi, J. Closier, R. G. Diaz, G. Kuznetsov, Y. Y. Li, R. Nandakumar, S. Paterson, R. Santinelli, A. C. Smith, M. S. Miguelez, and S. G. Jimenez, "Dirac: a community grid solution," *Journal of Physics: Conference Series*, vol. 119, no. 6, p. 062048, 2008. [Online]. Available: <http://stacks.iop.org/1742-6596/119/i=6/a=062048>
- [10] T. Glatard, C. Lartizien, B. Gibaud, R. Ferreira da Silva, G. Forestier, F. Cervenansky, M. Alessandrini, H. Benoit-Cattin, O. Bernard, S. Camarasu-Pop, N. Cerezo, P. Clarysse, A. Gaignard, P. Hugonnard, H. Liebgott, S. Marache, A. Marion, J. Montagnat, J. Tabary, and D. Friboulet, "A virtual imaging platform for multi-modality medical image simulation," *IEEE Transactions on Medical Imaging*, vol. 32, no. 1, pp. 110–118, 2013.
- [11] S. Camarasu-Pop, T. Glatard, H. Benoit-Cattin, and D. Sarrut, "Enabling grids for gate monte-carlo radiation therapy simulations with the gate-lab," in *Applications of Monte Carlo Method in Science and Engineering*, S. Mordechai, Ed. Rijeka: IntechOpen, 2011, ch. 3. [Online]. Available: <https://doi.org/10.5772/15904>
- [12] T. Glatard, J. Montagnat, D. Lingrand, and X. Pennec, "Flexible and efficient workflow deployment of data-intensive applications on grids with MOTEUR," *IJHPCA*, vol. 22, no. 3, pp. 347–360, 2008. [Online]. Available: <https://doi.org/10.1177/1094342008096067>
- [13] T. Glatard, G. Kiar, T. Aumentado-Armstrong, N. Beck, P. Bellec, R. Bernard, A. Bonnet, S. T. Brown, S. Camarasu-Pop, F. Cervenansky, S. Das, R. F. da Silva, G. Flandin, P. Girard, K. J. Gorgolewski, C. R. G. Guttman, V. Hayot-Sasson, P.-O. Quirion, P. Rioux, M. Étienne Rousseau, and A. C. Evans, "Boutiques: a flexible framework to integrate command-line applications in computing



- platforms,” *GigaScience*, vol. 7, no. 5, 03 2018. [Online]. Available: <https://dx.doi.org/10.1093/gigascience/giy016>
- [14] J. Blomer, P. Buncic, and T. Fuhrmann, “Cernvm-fs: Delivering scientific software to globally distributed computing resources,” in *Proceedings of the First International Workshop on Network-aware Data Management*, ser. NDM '11. New York, NY, USA: ACM, 2011, pp. 49–56. [Online]. Available: <http://doi.acm.org/10.1145/2110217.2110225>
- [15] T. Glatard, Y. Cointepas, O. Commowick, M. Kain, B. Laurent, F. Leray, and C. Barillot, “Carmin: a common web api for remote pipeline execution,” *Frontiers in Neuroscience*, no. 53, 2015. [Online]. Available: <http://www.frontiersin.org/neuroscience/10.3389/conf.fnins.2015.91.00053/full>
- [16] A. Rajasekar, R. Moore, C.-Y. Hou, C. A. Lee, R. Marciano, A. de Torcy, M. Wan, W. Schroeder, S.-Y. Chen, L. Gilbert, P. Tooby, and B. Zhu, *iRODS Primer: Integrated Rule-Oriented Data System*. Morgan & Claypool Publishers, 01 2010, vol. 2. [Online]. Available: <https://doi.org/10.2200/S00233ED1V01Y200912ICR012>