

MONTE CARLO SIMULATION FOR RADIOTHERAPY IN A DISTRIBUTED COMPUTING ENVIRONMENT

J. T. Moscicki

CERN

CH-1122 Geneva (Switzerland)

Jakub.Moscicki@cern.ch

S. Guatelli, M. G. Pia, M. Piergentili

INFN - Sezione di Genova

Via Dodecaneso 33, 16146 Genova (Italy)

Susanna.Guatelli@ge.infn.it, MariaGrazia.Pia@ge.infn.it, Michela.Piergentili@ge.infn.it

ABSTRACT

We show how nowadays it is possible to achieve the goal of accuracy and fast computation response in radiotherapeutic dosimetry using Monte Carlo methods, together with a distributed computing model. Monte Carlo methods have never been used in clinical practice because, even if they are more accurate than available commercial software, the calculation time needed to accumulate sufficient statistics is too long for a realistic use in radiotherapeutic treatment. We present a complete, fully functional prototype dosimetric system for radiotherapy, integrating various components based on HEP software systems: a Geant4-based simulation, an AIDA-based dosimetric analysis, a web-based user interface, and distributed processing either on a local computing farm or on geographically spread nodes. The performance of the dosimetric system has been studied in three execution modes: sequential on a single dedicated machine, parallel on a dedicated computing farm, on a grid test-bed. An intermediate software layer, the DIANE system, makes the three execution modes completely transparent to the user, allowing to use the same code in any of the three configurations. Thanks to the integration in a grid environment, any hospital, even small ones or in less wealthy countries, that could not afford the high costs of commercial treatment planning software, may get the chance of using advanced software oncological therapy, based on Monte Carlo methods, by accessing distributed computing resources, shared with other hospitals and institutes belonging to the same virtual organization.

Key Words: radiotherapy, Geant4, simulation, distributed computing, parallel execution, GRID

1 TYPICAL REQUIREMENTS OF DISTRIBUTED GEANT 4 SIMULATIONS

Typical requirements of an efficient distributed Geant4 [2] simulation may vary significantly depending on the application area.

Medical applications, such as brachytherapy [3], require interactive, almost real-time optimization of the treatment planning during the patient's stay in the clinic. The acceptable response time from the Monte Carlo simulation should not exceed a few minutes which is a dominant requirement. However the number of simultaneous jobs is small and typically there is one type of application of interest. Monte Carlo simulation is much slower than the analytical approach, albeit more accurate [4]. To achieve the response time comparable to currently used

analytical approach (based on commercial software in many clinics) one needs to run Monte Carlo simulation in a parallel way. In the clinical practice the computer clusters running such simulations would typically be small, ranging from several to one hundred machines.

The requirements for some astrophysics applications are on the other end of the spectrum. Certain Monte Carlo simulations for Lisa, a joint ESA-NASA experiment in space for measuring the gravitational waves, are of order of 10 processor-years in fully batch mode [5]. The critical issues in this case are reliable error recovery (to preserve the completed parts of the simulation), monitoring of the progress of the jobs and traceability of the failed worker tasks for debugging purposes. Typically, an order of 1000 CPUs is required to achieve expected performance. The computing power of this magnitude may practically be available through the WAN network of computing elements (computers), also known as a Computing GRID.

In case of semi-interactive applications in the multi-user environment there are other important issues to address. Today's cluster Workload Management Systems (such as LSF[6] and PBS[7]) have multi-user and multi-application capability with automatic load balancing. Quite naturally similar features would be required for semi-interactive public clusters which must be able to scale to hundreds of machines, support hundreds of users running concurrently their jobs each one consisting of several tens or hundreds of tasks. Clusters dedicated to a specific experiment or a group of users would certainly have different usage patterns. This requires not only a scalable software system but also a flexible one: it should be easy to fine-tune and modify the core system features according to the type of scientific application, access-patterns, average workload and other factors.

2 DIANE – DISTRIBUTED ANALYSIS ENVIRONMENT

Most often developers (as well as users) do not really care for the implementation details of distributed middleware which runs their distributed application. An application-oriented layer above the generic middleware infrastructure, such as Distributed Analysis Environment (DIANE), hides the complexity of the underlying distributed technology and provide additional functionality required for the parallel execution of jobs.

DIANE is a R&D study, focusing on semi-interactive parallel and remote data analysis and simulation, which has been conducted at CERN. DIANE provides necessary software infrastructure for parallel scientific applications in the master-worker model. Advanced error recovery policies, automatic book-keeping of distributed jobs and on-line monitoring and control tools are provided. DIANE makes a transparent use of a number of different middleware implementations such as load balancing service (LSF, PBS, GRID Resource Broker[8], Condor[9]) and security service (GSI[10], Kerberos, openssh).

Applications which use DIANE may run in a variety of different configurations without changes in their source code. Whether application is distributed by global GRID Resource Broker or local Workload Management System it actually does not matter neither for the application user nor for the developer. It is also easy to modify and adapt given DIANE configuration to the particular needs of a specific application. This capability is achieved with the layered design of the DIANE framework (Fig. 1).

DIANE has been developed in the context of High Energy Physics applications which, in most cases, follow the simple master-worker computational model. However this model is also common in other scientific areas, particularly in the ones which involve Monte Carlo event-level

parallel simulation. The difference comes from the non-functional requirements: user access patterns, execution time and data access patterns, CPU load versus I/O load, etc. The design of DIANE framework allows an easy adaptation to different practical scenarios by plugging the appropriate service components into the core system.

3 RUNNING GEANT 4 WITH DIANE

Further discussion relates to distribution of Geant 4 at the event level, i.e. treating every event as an independent unit of processing.

A typical Geant 4 simulation producing AIDA[11] histograms on output was “ported” to DIANE environment. Usually simulation is linked as an executable. For the distributed environment a shared library is built with the well-known entry point to create an object representing the simulation program. This object implements a very simple abstract interface which allows to run the simulation and to set certain parameters (such as random seed) programatically. The porting process is straightforward and involves a very minimal code change. The simulation ported in such a way is executed in the worker process.

To run a distributed application in parallel one has to know how to partition the job and what to do with the results that workers produce at the end of the job. This is specified in a separate library which acts as a glue between the application itself and the DIANE framework. Typically it is written once per type of application. In the discussed case, the workers’ output, i.e. histograms serialized to XML, were simply added at the end of the job.

Tests were performed using three independent mechanisms:

- ssh-based, explicit placement of distributed workers via remote execution of shell scripts in the local cluster,
- submission of the distributed workers into the LSF queues in the local cluster,
- submission of the distributed workers via the European Data Grid resource broker in the WAN network.

The transition between the three scenarios above required only minimal changes to the core of DIANE, which was the original design goal. It did not require changes to the test application itself.

The explicit placement mechanism allows to accurately measure the execution times. The initial tests involved a small, interactive cluster of 30 machines (35 machines were used for the largest jobs), deploying a Geant 4 simulation distributed at the event level, producing AIDA histograms on the output. Average execution times are shown in Fig. 2. As shown in Fig. 3 DIANE achieved around 75% of theoretical efficiency (measured as a ratio of speed-up and the number of CPU used) in this particular test configuration. The benchmarking environment in this test did not exclude other users from logging-in interactively to the machines what partially explains this result. However great care has been taken to remove random errors from the measurements by averaging the results and analyzing the hourly load on the cluster.

Fig. 4 shows the average execution times for large Geant 4 simulation on a *dedicated* cluster of 15/20 machines. As expected the performance of parallel execution is proportional to the number of available CPUs. The performance gain in this case could be even greater if dual

processor machines were exploited fully, i.e. DIANE could automatically allocate two worker processes per machine. This is one of the subjects of current work.

4 CONCLUSIONS

We successfully demonstrated a realistic medical physics use-case on the GRID. Using a Geant4 simulation and DIANE framework we run distributed simulation of a large number of events for dosimetry in brachytherapy.

Scalable application-oriented framework for distributed scientific applications must be flexible enough to meet diverse non-functional requirements. Fine-tuning for such requirements, including evolution and interchangeability of low-level service components, is an important design goal of DIANE. Early prototype of DIANE has been successfully deployed and initial tests in a distributed environment proves that important speed-up in the end-user simulation/analysis cycle is possible.

The paper demonstrates the feasibility of close integration of Geant4 and DIANE and presents the initial performance benchmark results.

5 REFERENCES

1. Distributed Analysis Environment, DIANE R&D Project, <http://cern.ch/diane>
2. Geant4, <http://cern.ch/geant4>
3. M. Tropeano et al, "Leipzig Applicators Montecarlo Simulations: Results and Comparison with Experimental and Manufacturer's Data", summary in [4].
4. M. G. Pia et al, "Overview of bio-medical applications", <http://www.ge.infn.it/geant4>
5. A Howard et al: "Experiences of Submitting UKDMC and LISA GEANT4 Jobs", <http://www.gridpp.ac.uk/gridpp4>
6. Load Sharing Facility, CERN Pages, <http://wwwinfo.cern.ch/pdp/lsf>
7. Portable Batch System, <http://www.openpbs.org>
8. European Data Grid, <http://www.edg.org>
9. Condor, <http://www.cs.wisc.edu/condor>
10. GSI: Grid Security Infrastructure, <http://www.globus.org/security>
11. Abstract Interfaces for Data Analysis, <http://aida.freehep.org>

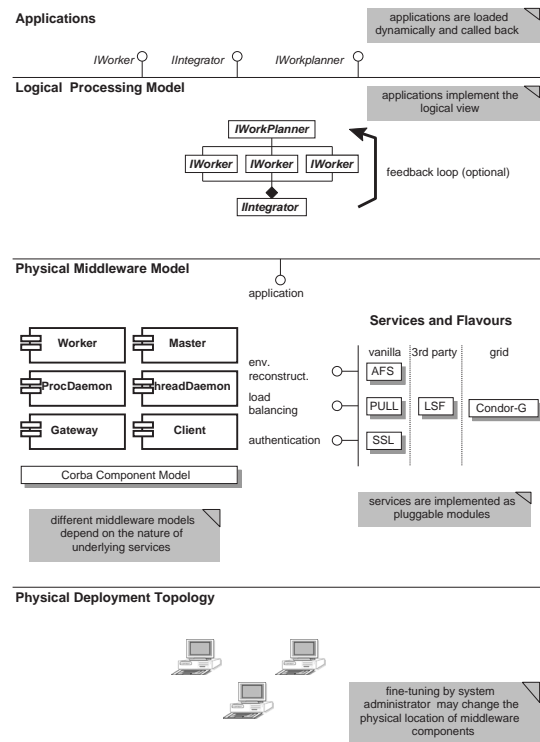


Figure 1: Overview of DIANE's layered architecture.

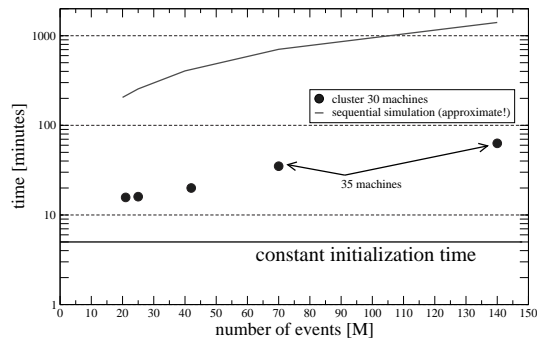


Figure 2: Average execution times for large Geant 4 simulation using explicit worker placement with ssh. Interactive cluster.

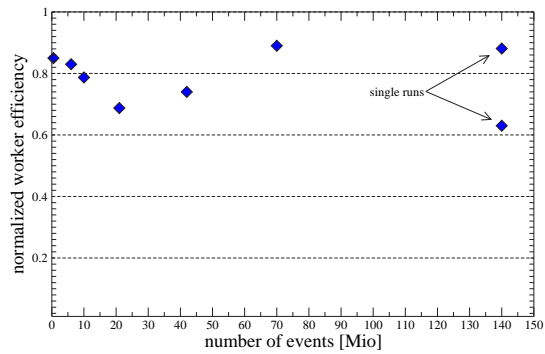


Figure 3: Normalized, average worker efficiency for large Geant 4 simulation using explicit worker placement with ssh. Interactive cluster.

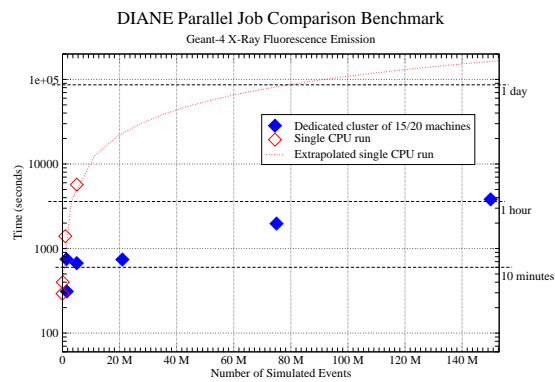


Figure 4: Average execution times for large Geant 4 simulation using explicit worker placement with ssh. Dedicated cluster.