

A COMPUTER CLUSTER SYSTEM FOR PSEUDO-PARALLEL EXECUTION OF GEANT4 SERIAL APPLICATION

MEMMO FEDERICI^{a,*}, BRUNO L. MARTINO^a

^a *Istituto di Astrofisica e Planetologia Spaziali, IAPS INAF Via fosso del Cavaliere 100, 00133 Roma, Italy*

^b *Istituto di Analisi dei Sistemi ed Informatica “Antonio Ruberti”, IASI-CNR Viale Manzoni 30, 00185 Roma, Italy*

* corresponding author: memmo.federici@iaps.inaf.it

ABSTRACT. Simulation of the interactions between particles and matter in studies for developing X-rays detectors generally requires very long calculation times (up to several days or weeks). These times are often a serious limitation for the success of the simulations and for the accuracy of the simulated models. One of the tools used by the scientific community to perform these simulations is Geant4 (Geometry And Tracking) [2, 3]. On the best of experience in the design of the AVES cluster computing system, Federici et al. [1], the IAPS (Istituto di Astrofisica e Planetologia Spaziali INAF) laboratories were able to develop a cluster computer system dedicated to Geant 4. The Cluster is easy to use and easily expandable, and thanks to the design criteria adopted it achieves an excellent compromise between performance and cost. The management software developed for the Cluster splits the single instance of simulation on the cores available, allowing the use of software written for serial computation to reach a computing speed similar to that obtainable from a native parallel software. The simulations carried out on the Cluster showed an increase in execution time by a factor of 20 to 60 compared to the times obtained with the use of a single PC of medium quality.

KEYWORDS: Geant4, cluster, Monte Carlo method.

1. INTRODUCTION

The system discussed here is essentially an implementation of the Geant4 software on a cluster computing platform. The toolkit uses the object-oriented programming paradigm; its application area includes experiments in high energy physics, nuclear studies, medical applications, accelerators and astrophysics. Writing a software simulation using Geant4 generally requires a significant design effort, so if possible developers of simulation models try to adapt programs that are already been implemented and tested for new projects by making only the needed changes. Large computing capability and long execution times are required in order to improve the accuracy and the quality of a simulation model. These capabilities are often obtainable only through the use of a cluster computer system able to execute software developed for parallel platforms. Unfortunately, the migration of software written for serial execution to parallel systems often requires a complete rewrite. The originality of our project is the capacity to reuse software for Geant4 that was written to operate on serial platforms, obtaining calculation performance similar to running native parallel system, without the need to make substantial changes to the software.

1.1. SCENARIO

The main goal in this Cluster design is to obtain the execution of simulations using the Monte Carlo method within reasonable times and at affordable

costs. The design approach has therefore focused on the following considerations:

- Minimizing the cost of the computer systems.
- Achieving acceptable computation times.
- Intensive reuse of serial simulation software.

This work shows the real possibility of reusing serial applications developed for Geant4 without major rehabilitation efforts to be performed in a pseudo-parallel mode.

1.2. PARALLEL APPLICATIONS

Writing parallel applications generally implies a challenging design. Converting serial into parallel applications can require a complete rewrite of the source which is code, very expensive whenever executed on high-end servers (machines with a large number of processor sockets, dozens, even hundreds, and a large amount of shared memory). This cluster overcomes the unfavorable aspects of the use of Parallel systems and transforms them to advantage.

1.3. COMPUTING ENVIRONMENT

Figure 1 shows the structure of the hardware of the Cluster, which currently consists of 8 PC (Nodes) with the following characteristics:

- Hardware: Intel I7 processor (8 cores), 4 GB of DDR3 RAM 1333 MHz, 500 GB Hard Disk SATA3

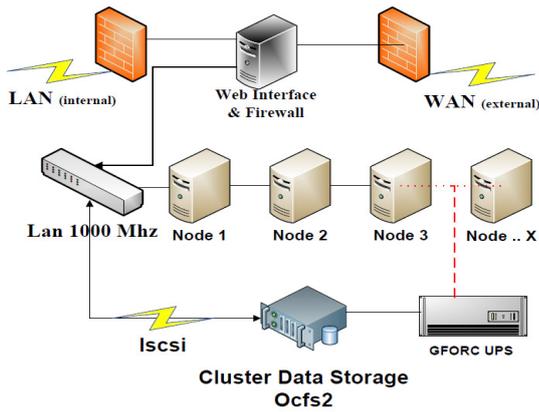


FIGURE 1. Cluster Hardware block diagram.

- Software: operating system: Linux Debian 6 Squeeze (x86-64), resource manager: SLURM [4], customized Bash script interfaces for distribution of computer load
- Data Storage: Storage on NAS RAID, filesystem OCFS2 (a GPL Oracle Clustered File System, Version 2), transport protocol: iSCSI

The hardware consists of commercial entry-level units; the software is free. The heart of the project consists of a set of scripts written in Bash (Bourne again shell) that handle all the operations related to the management of graphical interfaces and distribution of the workload across the nodes of the cluster in a definable pseudo parallel mode. To obtain an optimal result, the scripts split the overall workload on all cores allocated by the user, generates a “seed” randomly different for each instance of calculation and a macro file containing all the necessary parameters for the simulation. The cluster in its current configuration has 64 calculating cores and has multiuser capability. The Data Storage on which the “home” area is housed is made with a 6 TB NAS and is configured in RAID5. The File System is an OCFS2 that in the free version can handle up to 16 TB of disk space. The use of this file system developed for Cluster computing systems is able to manage input output data access simultaneously on all nodes. This feature offers great advantages, speeding input-output operations from the distributed computing systems. Data transport is performed using the iSCSI protocol which manages data storage very efficiently.

2. LOGIN AND SETTING OF PARAMETERS FOR SIMULATIONS

The user connects to the Cluster (login) through an SSH connection, providing his credentials. Once authenticated, a particular user profile script does the following:

- authenticates the users,
- sets the execution environment,



FIGURE 2. Example of login graphical interfaces.

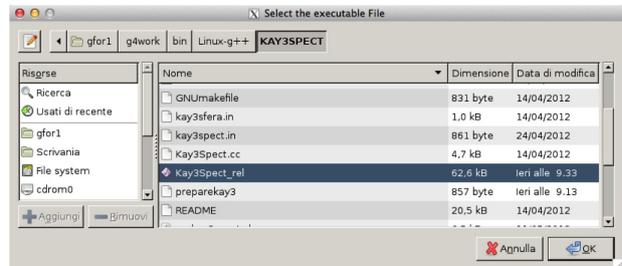


FIGURE 3. Example of run time graphical interfaces.

- sets the number of nodes devoted to the simulation,
- manages reconnection to active simulations.

In this phase, the system automatically performs the setting of the necessary environment variables to Geant4 and lets the user choose the parameters needed to perform the simulation. Figure 2 shows an example of graphical user interfaces shown on login. During this phase, one of the fundamental aspects of the Cluster is highlighted, i.e. the ability to resume sessions still active. This specially developed feature for the Cluster was necessary because of the long duration of the simulations. Thanks to this feature, the user may start the simulation and detach his local terminal from the cluster without interrupting the execution of his running jobs; at the next login she/he can check the progress of the simulation in progress not yet finished. At this stage, the user can manage the number of nodes in the cluster at its disposal to carry out other simulations. Each node provides the user 8 calculating cores. The system automatically frees the resources that have become available.

2.1. RUNNING SIMULATIONS

At run time a script makes it possible for the user to select the application to be run and generate the corresponding configuration files (one for each instance of the process). Using the appropriate graphical interface (see Fig. 3), the user can select parameters concerning the simulation such as: the executable file, the macro file containing all the parameters for the simulation, the creation of the work directory, and he can then start the run simulation.

3. SIMULATION CAMPAIGN

As an example of the cluster activity, we present two simulations: the first one involves the effect of cosmic particles on the ATHENA XMS microcalorimeter [5] to study the “anti-coincidence” system efficiency and

the effect of the non-vetoed background on the performance of the detector (see Fig. 4). This simulation is characterized by a large number of events; the time spent by the cluster to complete the simulation using 4 nodes for a total of 32 cores was approximately 10 days. This simulation performed on a PC identical to those used for the Nodes of the Cluster would take about 200 days of uninterrupted computing. The practical advantages of the use of the Cluster in this simulation relate mainly to the speed, which is increased by a factor of 20, while the relative cost undergoes an increase of only a factor of 4. In fact 4 nodes calculates with a relative speed 20 times higher than that of a single node. Another great advantage is that it makes possible more detailed simulations and a more realistic environment. It is also very unlikely that an uninterrupted run of 200 days on a single PC can get to the end without interruption.

The second simulation, in the area of medical physics, concerns the simulation of small detectors for gamma survey tomographic SPECT; single photon emission computed tomography can be used for research in medical oncology and in particular in the diagnosis of breast cancer. For this simulation, 7 nodes were used for a total of 56 cores, and the time employed was approximately 5 hours. The time that the simulation would take on a single PC has been estimated at about 15 days. In this case, the time for the simulation is decreased by approximately a factor of 60. This enables the development of more efficient detectors allowing changes and enhancements to the simulated model. Verification requires only a very short time.

4. CONCLUSIONS

The Cluster has been optimized for the purposes Geant4, it improves the speed for simulations that require large computational resources by a factor from 40 to 60 (compared with a single PC of the same category). It drastically decreases the probability of failure thanks to its great speed. It is inexpensive currently composed of 8 commercial PCs for a total of 64 cores. It is modular easily expandable without substantial changes, and can be easily reused on other projects.

ACKNOWLEDGEMENTS

The authors wish to thank: Lorenzo Natalucci, Maria Nerina Cinti, Sergio Lomeo.

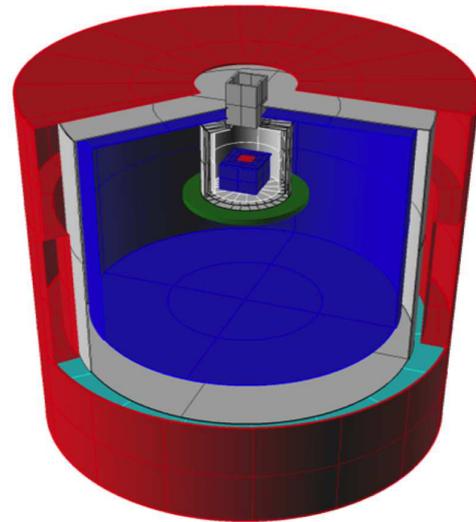


FIGURE 4. The Model of the XMS microcalorimeter.

REFERENCES

- [1] Federici, M. et al. 2009, POS, Published online at <http://pos.sissa.it/cgi-bin/reader/conf.cgi?confid=96>, p. 92
- [2] Agostinelli, S. et al.: Geant4 – a simulation toolkit, Nuclear Instruments and Methods in Physics Research A 506 (2003) 250–303
- [3] Allison, J. et al.: Geant4 developments and applications, IEEE Transactions on Nuclear Science 53 No. 1 (2006) 270–278
- [4] Yoo, A., Jette, M., Grondona, M.: Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science, 2003, 2862, 44–60
- [5] Lotti, S. et al 2012, Estimate of the impact of background particles on the X-Ray Microcalorimeter Spectrometer on IXO, arXiv:1205.3002v1 [astro-ph.IM]

DISCUSSION

James H. Beal — What is the highest data rate possible between processors in your system?

Bruno Martino — The communication between processors of different machines is via an ethernet LAN 1 Gb/s. The processes synchronization is managed by a master machine, which also takes care of monitoring.