Abstract

Nuclear physics experiments are currently conducted in many countries around the world. They have ambitious goals. Such as: the fight against cancer, the study of the Big Bang theory, and so on. Among them are the LHC (The Large Hadron Collider) experiment and the ATLAS experiment. The LHC experiment is taking place at CERN, Geneva, Switzerland. Its main goals are:

1. Experimental validation of a standard model and hence the discovery of a large mass Z boson and a Higgs boson
2. Study of the Big Bang theory
3. Confirmation of super symmetry

The above issues are being researched by proton collisions in the LHC experiment.

Simulation is one of the main cornerstones for the success of ATLAS Experiment. The aim of the simulation is to conduct a series of experiments not on real detectors but on their mathematical models. The simulation allows the non-existent, theoretical events to be investigated. The ATLAS experiment simulation is based on the Geant4 modeling package.

Approximately 5-7 billion artificial events are produced annually and Geant4 needs more than 77 million hours to process them. This is why simulation performance is a very important parameter. Performance is significantly influenced by the geometric descriptions of the ATLAS detector used in the simulation packages, which are created by different programming methods. The influence of geometric description programming methods on the performance of the simulation process has not been studied and its investigation is an important task.

<u>Chapter I</u> discusses various experiments in nuclear physics in the world today. Their goals and objectives. Also discussed are the ATLAS experiment software and the physical process simulation process. The Geant4 modeling package and various ongoing projects are discussed here, including the ATLAS experiment simulation infrastructure. At the end of the chapter, the research task and the methodological plan are formulated.

<u>Chapter II</u> contains information on the formation of test samples required for research. The formation of test samples was carried out in several stages, as follows:

1. Selection of the typical geometric descriptions of the ATLAS detector
2. Selection of programming methods for test examples
3. Analysis of selected programming methods
   3.1. Exclude similar topologies
   3.2. Exclude irrational methods

3.3. Exclude theoretical methods

After the completion of the 1st stage, 66 test samples were produced, which were divided into 3 classes: hollow, cylindrical and mixed. After step 2, 2713 programming methods were selected for 66 examples. After the third stage, 42 test samples were produced for the study with its 416 programming methods. The conclusions of Chapter 2 are included in the same chapter.

Chapter III covers the issues of defining research infrastructure. This includes the following topics: Determining input information for the simulation, determining the criteria for evaluating the performance of the simulation process and the CPU Time noise ratio. As a result of the of the 1st task, the input information of the simulation was determined:

1. Simulated events created by Monte Carlo generators
2. Geometric descriptions of the detector
3. Condition Tag

The simulation performance assessment criteria were defined and these criteria are RAM memory measured in KBs and CPU Time measured in milliseconds.

The servers that rim simulation also work other tasks in parallel, hence the best server in terms of CPU Time noise ratio has been determined. 3 different servers were analyzed: lxplus703.cern.ch, lxplus723.cern.ch and lxplus782.cern.ch. The lxplus703.cern.ch server was selected for testing, because it showed the best result in terms of CPU Time noise.

Chapter IV provides information on simulation test sessions. The sessions were conducted in several stages. Testing was performed on cylindrical, polygon and mixed objects. As a result, ranking of programming methods in terms of performance was determined. Different number of objects were also tested to investigate the impact of the number of objects on the simulation performance. Studies have shown that programming methods have an impact on simulation performance, and that changing the number of objects has an impact on performance. At the end of the chapter are the conclusions of Chapter 4.

Chapter V discusses research findings and its approbation. Approbation was performed on one of the ATLAS detector sub-detectors NSW (New Small Wheel). NSW geometry was described using different programming methods. Optimal and non-optimal versions of its code were created. Their study found that the efficiency of the adopted methods was 62.1% according to the RAM load and 70.3% according to the CPU Time.

The dissertation also contains the main conclusions of the research:

1. 66 typical geometric descriptions of the Atlas Detector were selected

2. Developed classification of programming methods for geometric descriptions, which combined - typical programming methods for the cylindrical class 29, 135 for the concave class and 252 for the combined class
3. Identified Typical classes and methods for programming geometric descriptions of the Atlas Detector were identified
4. For programming polygonal objects, it has been proven that polygon methods provide better performance compared to solid-body primitive methods, which increases with the number of objects - 8% for CPU and 20% for RAM
5. For the programming of cylindrical objects, it was found that the change of methods causes a slight change in performance, which reaches 4% with the increase in the number of objects
6. Polygon methods for programming combined objects have better performance than solid-body primitive methods, and this difference increases in direct proportion to the number of objects
7. A simulation performance study infrastructure has been created
8. The effectiveness of the Tube method for the programming tasks of cylindrical objects was revealed
9. The effectiveness of the Arbitrary Polygon method for the programming of polygonal objects was revealed
10. Slight differences were found between the Cube and Pyramid methods for both polygonal and combined object programming tasks.
11. Programming methods were ranked by performance