DZero Monte Carlo Simulation

G. E. Graham*

Department of Physics, University of Maryland, College Park, Maryland 20742-4111 †

Abstract. The DZero Monte Carlo is discussed. In particular, the generation and GEANT simulation programs are described. The DZero Parametrized Monte Carlo System (PMCS) is also discussed. Finally, the DZero Monte Carlo mass production system is described.

INTRODUCTION

Monte Carlo simulation at DZero is quickly gearing up in preparation for Run II. In this note, we will discuss the various components of the DZero Monte Carlo; including the available generators, the GEANT-based simulation program D0gstar, the digitization and minimum bias/pile-up simulation D0sim, and the DZero Parametrized Monte Carlo System (PMCS). In addition to the core software effort, a Monte Carlo mass production effort for the GEANT-based Monte Carlo using a new set of tools and the SAM system will be discussed.

Core Software

Software at DZero is written in C++ and organized into modules using cvs in order to support a multi-developer environment. All production executables at DZero make use of a framework(1) which further organizes DZero code into packages. Each package roughly corresponds to a logical unit of work, e.g.- event reconstruction in the central preshower, geometry management, or pythia Monte Carlo event generation. At runtime, packages are registered within the framework and are invoked through known hook methods in a well defined order on a set of events. The ordering of the hook methods are determined at initialization time by user specification or by probing of package dependencies by the framework itself.

Each event comprises a collection of chunks(2). Each chunk contains a logical unit of data along with parentage information, so that it is possible to identify later the exact package which created the chunk and its runtime configuration. Runtime configuration parameters are tracked by a custom built database(3).

Monte Carlo data is stored at DZero using the SAM system(4). SAM is a distributed file management and storage system produced by the Fermilab Computing Division. SAM retains parentage information at the file level. Events are recorded using D0OM(5) (DZero Object Model) for persistancy. Though events are stored in files, random access is still possible through the use of event keys in each file header. D0OM is loosely based on ODMG standards, and supports the CERN DSPACK and EVPACK I/O packages.

Monte Carlo Generators

There are several physics generators available in the DZero software environment.

- Pythia, version 6.129.
- Isajet, version 7.44.
- Herwig, version 6.1
- Custom generators for single particles and cosmics

The generators can all use the TAUOLA package for correct tau decay dynamics and the QQ libraries for the correct B decay dynamics. They also make use of the STDHEP interface and Monte Carlo particle numbers, but the format(MCPP) is internal to DZero.

GEANT-based simulation : D0gstar

D0gstar is a simulation program for the DZero detector in Run II based upon GEANT 3.21. The GEANT 3 FORTRAN library code is made available to the DZero framework by wrapping the FORTRAN in C++ where

* for the DZero collaboration
† Present Address : Fermi National Accelerator Laboratory, Batavia, IL 60510

needed. Most of the DZero sub-detector systems are included, and the remaining will be integrated soon. These include the Forward Proton Detector which requires the use of a double precision version of GEANT 3, and the luminosity monitor.

The performance of D0gstar was estimated on a 200 MHz Origin 2000 using Pythia generated $t\bar{t}$ events including all decays with no minimum bias. On average, D0gstar required 110 seconds to process each event and wrote 0.9 MB of output. These numbers are preliminary, as issues of output content and optimization are still being resolved.

Digitization, minimum bias and pile-up : D0sim

D0sim is a multi-purpose program which does the digitization of the hits data provided by GEANT. Other packages are included with D0Sim which do the pile-up simulation and the minimum bias simulation. Minimum bias events need to be added in the current crossing and also for a number of previous and future crossings to simulate luminosity effects. This simulation takes into account hardware baseline subtraction done by the DZero calorimeter electronics(6).

The performance of D0sim was also estimated on a 200 MHz Origin 2000 using Pythia generated $t\bar{t}$ events including all decays and processed by D0gstar with no minimum bias. On average, D0sim required 80 seconds to process each event and wrote 1.0 MB of output. In addition, tested with a Monte Carlo sample of Isajet generated QCD events with a lower limit on P_T of 5 GeV/c, it was found that D0Sim required about 1 additional minute of running time plus one additional megabyte of output per event for each additional minimum bias event added in the crossing of interest. These numbers are also preliminary, as issues of output content and optimization are still being resolved.

Parametrized Monte Carlo System : PMCS

DZero also has a parametrized Monte Carlo System (PMCS) for generating large numbers of events quickly. PMCS is logically divided into three tracks.

- FAST track : Simulation of high level physics objects only.
- MEDIUM track : Simulation of high level physics objects plus some important detectors.
- SLOW track : Simulation of detectors.

As other DZero code, PMCS is organized in packages. This plus the above organization has facilitated the development of PMCS. While the management focus is on the FAST track packages to simulate high level physics objects, PMCS can still easily accept contributions of parametrized detector simulations at any time from developers outside of the normal PMCS development team.

The output of PMCS packages is in the form of standard DZero chunks compatible with other DZero tools such as analysis programs and visualization tools. The lower level detector packages in PMCS should also be suitable for providing input to some of the standard reconstruction packages.

The performance of PMCS was also estimated on a 200 MHz Origin 2000 using Pythia generated $t\bar{t}$ events including all decays. processed by D0gstar with no minimum bias. On average, PMCS and Pythia together created and processed about 22 events per second with Pythia accounting for about 50 percent of the processing time.

EVERYBODY WORKING TOGETHER : MC_RUNJOB PACKAGE

The mc_runjob package consists of about 10K lines of Python code which sets up and produces shell scripts to run most D0 offline executables. All of the interesting input parameters are settable through the mc_runjob interface, and repetitive tasks such as random number seed generation are handled automatically. Furthermore, it is possible to configure and chain together multiple executables in a single job. Thus generation, simulation, digitization, reconstruction, and analysis can be configured together through a united interface. Intermediate files are retained and handled automatically. Furthermore, and perhaps more importantly, mc_runjob is script driven so that an mc_runjob script amounts to specification of a processing chain. This aids immensely the tasks associated with large scale production and makes possible completely automated large scale Monte Carlo production.

mc_runjob is modular so that the addition of Python code to handle new executables or tasks is greatly simplified. Furthermore, the inheritance hierarchy is organized to permit the use of a graphical user interface that is automatically generated from stored information about each executable only; so that minimal maintenance is needed to extend the GUI to new executables once the basic Python code is written.

MONTE CARLO PRODUCTION

DZero is able using the above tools to generate massive amounts of Monte Carlo data. This Monte Carlo data is used to flex the generation and storage tools and to estimate the physics performance of the DZero detector. In the production phase ending April 2000, well over one million Monte Carlo events were generated at five centers around the world(7). Currently, more centers are coming online and 600K events have been generated already for the Fall 2000 production. Capacity is already approaching 100K events per day equivalent no minimum bias.

An automatic Monte Carlo request processing system taking advantage of the mc_runjob system and SAM is envisioned for the start of RunII. Tables in the SAM Oracle database have been completed which describe

- production requests for Monte Carlo
- production input parameters
- generator and detector card files
- physics processes
- processing chain (for input to mc_runjob)

The system as currently planned will accept Monte Carlo processing requests via a WWW interface and generate events with the appropriate DZero generator at FNAL using mc_runjob. The resulting generated events are then stored into SAM for tracking purposes. Later, an automatic process retrieves the generated events from SAM and deposits them in a remote-site specific area with instructions for further processing in the form of an mc_runjob script. Remote sites are responsible for retrieving these generator files and instructions, performing necessary processing remotely, and returning the results back into SAM.

REFERENCES

1. J. Kowalkowski, et al. "D0 Offline Reconstruction and Analysis Control Framework," Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP) 2000, INFN, Padua
2. May, B. and Paterno, M. "The D0 Event Data Model," Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP) 1998, ANL, Chicago, IL
3. Paterno, M. "Run Control Parameters at D0," http://cdspecialproj.fnal.gov/d0/rcp/D0LocalGuide.htm
4. L. Lueking, et al. "The Data Access Layer for D0 Run II: Design and Features of SAM," Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP) 2000, INFN, Padua
5. Greenlee, H. and Snyder, S. "The D0 Object Model for Persistent Data," Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP) 1998, ANL, Chicago, IL
6. The DZero Collaboration. "D0 Upgrade Technical Design Summary," D0 Internal Memo #2962 and references therein.
7. G.E. Graham. "The D0 Monte Carlo Challenge" Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP) 2000, INFN, Padua