Computing Models of CDF and DØ in Run II

Stephan Lammel

Fermi National Accelerator Laboratory,
Batavia, Illinois 60510

The next collider run of the Fermilab Tevatron, Run II, is scheduled for autumn of 1999. Both experiments, the Collider Detector at Fermilab (CDF) and the DØ experiment are being modified to cope with the higher luminosity and shorter bunchspacing of the Tevatron. New detector components, higher event complexity, and an increased data volume require changes from the data acquisition systems up to the analysis systems. In this paper we present a summary of the computing models of the two experiments for Run II.

1 Introduction

Computing has a very special role in high-energy physics (HEP). It enables us to push the frontier of particle physics and thus uncover more and more of nature's secrets. Without advanced computing this kind of research would not be possible.

The Fermi National Accelerator Laboratory, located near Chicago, is home of the world's highest energy collider, the Tevatron. Two experiments analyse the proton—anti-proton interactions. The Collider Detector at Fermilab (CDF) has been in operation since 1985. The second experiment, DØ, recorded its first collisions in 1991. Both experiments have been very successful since their first data were recorded.

2 Fermilab Run II

Fermilab Run II is defined as the first collider run with the Main Injector. It is scheduled to start in autumn of 1999, i.e. in a little over 2 years. The duration

Preprint submitted to Elsevier Preprint *16 May 1997*

of the run is two years. The center-of-mass energy of the $p\overline{p}$ collisions of the Tevatron will be increased to $\sqrt{s} = 2\,\mathrm{TeV}$. The beam crossing time will be reduced from the current $3.5\,\mu\mathrm{s}$ to either 396 ns (in the case of 36 bunches) or 132 ns (in the case of 108 bunches). The design luminosity for Run II is $2 \cdot 10^{32}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. The goal of Run II is the accumulation of an integrated luminosity of $2\,\mathrm{fb}^{-1}$ by each of the collider experiments.

3 Changes to the Experiments

To handle the higher luminosity the central tracking system of CDF will be replaced completely. The new tracking system will consist of a next-generation silicon vertex detector, an intermediate silicon detector, and a new open-cell drift chamber. The old gas proportional chamber-based calorimeters in the end-cap and forward region are being replaced with scintillating tile calorimeters. New muon chambers are added for increased coverage. All front-end electronics will be replaced to handle the shorter bunch-spacing.

The DØ eperiment will be upgraded with a central magnetic field. A 2.8 m long, 60 cm radius superconducting solenoid will produce a magnetic field of 2 Tesla. The new tracking system inside the solenoid consists of a silicon vertex detector and a scintillating fiber tracker. New muon trigger detectors are required in both central and forward region as the new bunch-spacing will be shorter than the drift time of the muon chambers. Upgrade of all the front-end electronics is driven by the need to handle a smaller bunch-spacing and to provide pipelining of the various front-end signals.

4 The Run II Computing Challenge

In a joint effort with the Computing Division the experiments have estimated the expected Run II data volumes and needs associated with them. Table 1 gives an overview of the anticipated data volumes and their sources. The experiments expect to record between half a PetaByte and one PetaByte of data. About half of the data will be raw data, i.e. direct output from the detectors. We note that the data loggers in both experiments are capable[1] of recording data at much higher rates. The data logging rate is determined by the output rate of the level 2 trigger system and the level 3 trigger rejection factor.

---

[1] The DØ online system has a rather inexpensive option of doubling the data logging rate. The CDF data logger could record data at close to the level 2 trigger rate.

	DØ	CDF
DAQ level 2 output	800 Hz	300 Hz
DAQ peak data logging	53 Hz	75 Hz
total raw data size	300 TByte	500 TByte
total reconstructed data	150 TByte	270 TByte
total physics analysis data	106 TByte	158 TByte
total data volume	**560 TByte**	**928 TByte**

Table 1
Expected data rates and volumes of DØ and CDF in Run II.

To reconstruct these data requires compute power of about $60,000\,$MIPS per experiment. Analysis CPU requirements are estimated to be between $70,000$ and $90,000\,$MIPS per experiment.

The manpower to write all required reconstruction and analysis software for the DØ experiment is estimated to be around 180 FTEyears.

5 Software Development Environment

CDF plans on a mixed language environment. Software development in FORTRAN 77 for both reconstruction and analysis is guaranteed. No extensions of FORTRAN (e.g. no vendor extensions and no FORTRAN-90) are allowed. This allows reuse of Run I software where appropriate. A provision for using software written in ANSI C++ is being made. This is not restricted to object oriented (OO) C++ but includes procedural code, i.e. ANSI C. Although the experiment has made a mixed language decision, there is an agressive activity inside CDF to provide all Run II reconstruction software in C++.

DØ on the other hand has made a clear decision for C++. The experiment plans to write all reconstruction and analysis software in object oriented C++. FORTRAN 90 will be only available for some legacy code.

Most people in the CDF and DØ collaborations are proficient in the use of FORTRAN. About 100 to 150 people in each collaboration[2] have developed software for Run I. The Fermilab Computing Division has sponsored several object oriented analysis and design (OOAD) courses as well as C++ lectures. There are about 20 people in CDF and 50 people in DØ who took such a

---

[2] The CDF and DØ collaborations are about 450 people each.

course during the last two years and are still with the experiment. Currently there are about 15 people in each experiment developing in C++.

The Computing Division sponsored OOAD courses are based on the OMT design methodology and notation which therefore have become a de facto standard. To facilitate development in C++ the Computing Division has made CASE tools available to the experiments. Graphical Designer and Insure++ are currently being evaluated. No final choice of design and development tools has been made. Various C++ compilers have been evaluated. The experiments expect to use compilers from several vendors on different platforms.

For code management both experiments plan to use CVS in a client-server mode. Some extensions for remote access were developed by the Sloan Digital Sky Survey (SDSS) experiment and are now supported by the Computing Division. The BaBar release tools and UNIX make procedures will be used in building the software packages. Fermilab's UNIX Product Support (UPS) and UNIX Product Distribution (UPD) products will be used to access and distribute the software packages.

To control software quality and assure high quality the DØ experiment has a review board for all its software. The collaboration is investigating automatic procedures to help in the process. CDF plans on a multi-level system for quality assurance under oversight of a release manager. In this system level 3 trigger filters would be at the highest level of quality control and analysis software at the lower end.

Both experiments plan to provide provisions for distributed development. However, DØ considers their central analysis system to be the primary platform for all development.

6 Analysis Environment

The CDF experiment is comitted to an OO infrastructure with a modular program architecture, to allow both object oriented and procedural components. For this a new analysis driver (AC++) similar to the successful Run I analysis_control package is currently being developed. Such framework will allow the user to control the execution of selected modules at run time. Reconstruction (or analysis) units form a module which is controlled through the analysis driver. Modules are autonomous and do not communicate directly with each other.

DØ has made the decision to move all large software projects to C++. Their framework approach has a set of modules that execute sequentially, each having a specific task. The glue that holds the individual software packages together will be an interpreted script system. The main task of this framework is to "guide" data between the various modules/packages. A sketch of the approach is shown in Figure 1. A prototype framework based on the Python scripting language has been developed and is ready for use. Also available is a first version of a program builder that combines a list of modules into an executable.

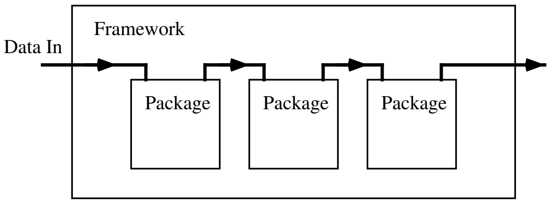

Fig. 1. Sketch of the Run II DØ framework.

7 Data Sharing and Persistency

DØ used the ZEBRA memory management and I/O package from CERN in Run I to organize their data. The experiment would like to separate the memory management functionality from the I/O part. The idea is to use memory management primarily as supplied by the language, i.e. C++, and adopt DSPACK as data handling package. DSPACK will not be used directly by the users but accessed through a thin layer to make the package exchangable. A group in DØ is currently engaged in the design and implementation of a C++ data persistency interface. The interface is loosely based on the ODMG standard.

CDF has re-written the old persistency package, YBOS, to provide access to data in the banks as objects from C++. This rewrite is almost complete and being tested right now. For CDF all data exchange and communication between modules inside the analysis driver will be via banks.

8 Data Storage, Logical

The DØ DSPACK based I/O interface does not follow the ODMG but has a more event-oriented I/O model to flatten the complex data structures. The output of this I/O interface are platform independent sequential files. Prospect of using an OODB is being explored.

CDF has not yet made a decision on how to store its data in Run II. The tool that provides data access in C++ can write sequential YBOS type files. The use of an event database is very desirable and under investigation.

In Run I the production systems created about 20 datasets of roughly equal size (in each experiment). These datasets provided the starting point of most analyses reducing the input of the first selection by a factor of 20. Tapes with full reconstruction output but also with only the physics analysis data were created for most of the datasets. This reduced the data volume seen by the physicist by another order of magnitude. For Run II both experiments would like to maintain a dataset oriented organization of their data.

For Run II both experiments have agressive plans to control the size of reconstructed data. In the current estimates only about 50% of the event data would be reconstructed information, of which roughly 15% would be physics analysis data.

9 Online Systems

The CDF online is a complex system. The basic architecture is very similar to that used successfully in Run I. Front-end and trigger electronics are housed in VME crates. Commercial processors read data from modules in their local crate and deliver it to the VME readout boards and event building subsystem. This system concentrates the data and delivers it to the level 3 trigger system through a network switch. The three logical components of the CDF online system are:

- Run Control
- Event Monitors/Consumer Processes
- Alarms Subsystem (accelerator and detector monitor)

Consumer processes will run in the same environment as the offline. Monitor systems will have a client/server implementation for data collection and display/analysis. All CDF online software will be ANSI C, Java, and Tcl/Tk.

The DØ online system is driven by accellerator changes similar to CDF. Also in DØ the architecture is largely unchanged from the current system. Front-end electronics are fully pipelined. VME buffer drivers in each front-end electronics crate will drive one of eight parallel high-speed data cables feeding multi-port memories accessed by a farm of event-building and software filtering level 3 processor nodes. The four logical components in the architecture of the DØ data acquisition system are:

- Readout Electronics with embedded processors

– Trigger Subsystem
– Host System with data logger and monitoring nodes
– Control Subsystem

Plans are to run data monitoring processes in the offline framework. Run I online software was mostly written in FORTRAN and PASCAL. For Run II this will become FORTRAN and C++. DØ plans to have its third level trigger based on Windows NT machines.

10 Production Systems

Both experiments currently plan on using CPU farms running UNIX to reconstruct their data. PCs will most likely be the most cost effective CPUs. In a joint effort with the Computing Division the use of commodity PCs is currently being investigated. The production systems will most likely be stand-alone and not tightly integrated into the analysis systems.

In Run I CDF had an expressline system, processing in real-time the most interesting events (about 10%). The expressline provides an excellent place for debugging reconstruction software. It restricts frequent re-processing to a small subset of the data. Due to the real-time processing the expressline events can also be used for detector monitoring. Having the most interesting events always available with latest reconstruction version is also important in expediting important analyses. The expressline system was very successful in Run I and CDF plans to have a similar system for Run II.

DØ plans for sufficient recources to reconstruct the full event stream in semi real-time, i.e. within one or two weeks.

11 Data Storage and Management

A decision for a serial media has not yet been made. The decision for Run II will be made end of summer 1998.

With the huge data volume hierarchical storage solutions are required. Both experiment expect the bulk of the data to be shelf-resident, i.e. operator mounted. In a joint effort of DØ, CDF, and the Computing Division the minimum amount of robotic served serial storage was estimated to be around 200 TByte per experiment. Disk resident storage of order 20 TByte per experiment are envisaged.

Fermilab is member of the HPSS collaboration. The software is currently used to manage the data from the fixed-target run as a production test. At this time, no selection of hierarchical storage management (HSM) software for Run II has been made.

During Run I the experiments logged their data at the experiment. For Run II the experiments would like to move toward remote datalogging, i.e. directly into the storage system in the Feynman computer center.

12 Non-Event Databases

CDF has currently several different catalogue and database management systems (DBMS). They are used for tracking of data logger and production output, to store calibration, alignment, and run condition information, as well as administrative purposes. With the Fermilab migration off VMS several of the VMS based systems are already being moved to mini SQL on UNIX.

DØ is in a similar situation with even larger number of DBMSs. A working group developed a requirements list and selected four object-oriented databases (OODB) as potential candidates for Run II databases. The DØ candidates are UniSQL, Objectivity, Object Store, and O2. Evaluation and testing to understand how these DBMS products could meet DØ needs has begun.

13 Analysis Systems

DØ kept data and analysing CPU rather separated during Run I. The experiment has identified this as a problem and would like to provide a central analysis cluster that can handle all development and analysis load. Figure 2 shows a sketch of the anticipated DØ computing systems of Run II. In the DØ analysis model all data is directly accessible by the central analysis cluster. The model assumes that there is no large data replication.

CDF plans to build an environment that supports distributed development and analysis. The experiment is not fixed on either centralized or only desktop based analysis. In Run I desktop CPU resources were not optimally utilized. They are considered valuable resources and for Run II an attempt is made to better integrate such systems. CDF likes to keep data and analysing CPU tightly coupled as was in Run I. No data replication other than caching is envisaged. Similarly to DØ also CDF has not budgeted large resources for

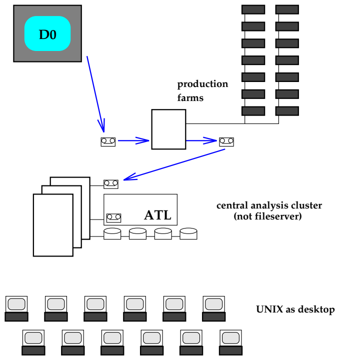

Fig. 2. Sketch of the anticipated DØ Run II computing systems.

data distribution. A sketch of the anticipated computing systems is shown in Figure 3.

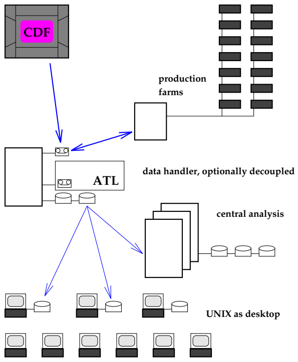

Fig. 3. Sketch of the anticipated CDF Run II computing systems.

14 Simulation

CDF currently has two detector simulation packages that were written in the collaboration. The experiment intends to move to standard-HEP tools. A C++ framework based on GEANT-3 (with wrappers around the FORTRAN) is currently being developed for Run II detector simulation.

DØ will continue to use the GEANT simulation framework. Over the next months DØ plans to move to a new framework for the geometry definition based on the CERN TZ package. This is done to enhance portability and transition to GEANT-4. DØ is also investigating the fast simulation package of the Computing Division, MCFast.

15 Graphics

DØ and CDF expect all their graphics packages to be X11 based (motif, Tcl/Tk, etc.) for 2-dim or OpenGL based for 3-dim applications.

People from both experiments are members of the HEPVis collaboration, which is developing a general purpose graphics toolkit (based on the OpenInventor product).

16 Documentation and Communication

The WWW has become the standard for documentation (HTML and PostScript). Both experiments are making heavy use of the web and plan to increase their usage further.

Both collaborations are including remote institutions into their physics meetings via EVRN, Picture Tel, or mbone based tools (nv, vat, wb, etc.) on a regular basis. For the widely spread collaborations of DØ and CDF robust wide area communication tools are essential.

17 Conclusions

The strategy of both experiments is to stay very flexible and make use of appealing new technologies that may become available at the Run II timescale. Both experiments are eager to move from custom developed tools to more standard tools.

The computing model of both experiments are based on the successful Run I models with adjustments were neccessary or where technology has opened new options or provides new appealing solutions. For DØ this means a move towards a more centralized analysis system/cluster.

Object-oriented technologies are being explored by both experiments for Run II software engineering. DØ has made a clear decision for object-oriented C++ software while CDF plans on a mixed language environment. However, none of the experiments has reached the required expert base to develop all core reconstruction software in OO.

Investigation of solutions for how to handle the huge amount of data have just started in both experiments.