CDF/DOC/COMP_UPG/PUBLIC/4810

CDF Production for Run II

Jaroslav Antos Antonio Chan Paoti Chang
Yen-Chu Chen Stephen Wolbers GP Yeh Ping Yeh

November 15, 1998

1 Introduction

To achieve the physics goals of CDF in Run II the data from the detector has to be processed through the reconstruction program. This should be done as quickly as possible in order to allow further processing steps such as splitting of the data into physics streams which are then analyzed by CDF collaborators. CDF production therefore needs to have the capacity and the flexibility to process all the data quickly and easily. In addition, there will be demands placed upon the production system by reprocessing, Monte Carlo generation and reconstruction, etc. Prioritization of various tasks also has to be possible in this system. The production system that is being designed and built will attempt to satisfy all the demands that will be placed on the system as described in this note.

2 Production Goals

There are many goals for CDF production. Some of the major goals are the following:

1. Sufficient capacity to reconstruct raw data.

   Clearly the first goal of a production system is to have the capacity to reconstruct all of the raw data from the experiment in "quasi-real time", meaning that the capacity is there to process data taken in a given time period (e.g. a month) in that same time period, with a small delay which is required for calibration constants preparation. There should be no design constraints in the system which preclude reconstruction with essentially no delay.

One caveat that must be placed on this goal is that the system will be designed and built with some assumptions about the total L3 output rate. If the L3 output rate exceeds this initial assumption then it will be necessary to delay the reconstruction of some low-priority streams until time can be found during accelerator downtimes or other processing power can be obtained. This was done in Run 1b with Stream C.

2. Sufficient capacity to reprocess some data.

   In addition to processing data for the first time there is a requirement that data be reprocessed in certain instances. The farm is not being sized to allow reprocessing of all the data, and it is felt that this will not be necessary. Early processing of Run II data will likely need to be reprocessed once the detector is well-understood and the reconstruction packages are tuned. We need to plan capacity to be able to meet that need. There may be smaller subsets of the data (specific triggers, for example) that may require reprocessing to achieve consistent datasets. It may also be the case that reprocessing is accomplished on the output and not the raw data files. The system should be able to schedule this.

   Other goals for the CDF production can be grouped into two categories: ease of use and bookkeeping.

3. Ability to easily configure farm processing to control priorities.

   The ease of use requirements include the ability to control priorities on the farms. This is important to allow the farm to be configured quickly to handle rapidly changing needs of the collaboration. There may be a new trigger which needs to be understood and tuned, a physics result which needs rapid processing, or other conditions under which the farm has to be configured to process certain sets of data using certain code and constants. In addition to the ability to reconfigure the farm we need to be able to manage the farm in steady-state production. This should be accomplished in such a manner that many people can run the farms, bringing the potential for standard "CDF shifts" to the farms.

This could also allow physics groups or other projects to make use of parts of the farms for special projects. Finally, it should be possible for Monte Carlo simulation jobs to be run on the farms. This includes the simulation of events as well as the reconstruction of the simulated events.

It should be noted that the ability to reconfigure the farm, change executables, etc. should be left to the experts. The "CDF shift" people should submit jobs and monitor their progress.

4. Excellent bookkeeping.

   The bookkeeping that is required on the farms includes the details of versions of code, constants, etc. which were used as well as a clean and clear way to correlate events or files with those version numbers. It also may include the capability to handle various requests about the number of events run through each version of the reconstruction package, etc. The exact information which needs to be kept will be driven by the needs of the people who run the farms, as well as the needs of the physicists who will analyze the data.

5. Excellent error handling and error prevention.

   One additional goal of such a system is to have excellent error handling. This makes such a system much easier to run and to maintain, and good attention to this will pay off over the long run. Another aspect of this is to have a well-designed system in which errors are less likely to occur. In other words, attention will be paid to the experience of Run I farms and other experiences as the system is designed.

3 Capacity Required

The capacity required for the farms is driven by the rate events are to be taken by the experiment and by the time it takes to reconstruct each event. Additional time will have to be added to handle reprocessing, Monte Carlo, splitting and any other anticipated needs.

To handle the DC rate of the experiment we have chosen to take the rates as given in table 1 of CDF note 4100 (also known as the DMNAG report). The farm is required to achieve a steady-state rate of 28 Hz to keep up with the data coming from the experiment. The CPU time/event, also given in CDF 4100 in table 1, is 1200 MIPS-s/event. If one multiplies 28 Hz by 1200 MIPS-s/event and assumes 100% processing efficiency, 33,600 MIPS are required. If one assumes a pessimistic 70% processing efficiency (where processing efficiency = total CPU delivered to reconstruction/total CPU possible) then the need is 48,000 MIPS. It is thought that a more appropriate way to present the capacity, given the uncertainty in the rate and the CPU required per event, is a range. A reasonable assumption at this time is that CDF will require:

$$48,000 - 72,000 \text{ MIPS}$$

Note that other reconstruction needs exist in CDF for Run II on the farms. One can categorize these as Monte Carlo, Analysis, Reprocessing, and splitting. There are no very good ways to estimate these numbers but for the purpose of this note and for planning the farms the following numbers have been discussed.

Table 1: CPU needed on CDF Run II Farms

Reconstruction	48,000 - 72,000 MIPS
Monte Carlo	10,000 - 40,000 MIPS
Analysis	0 MIPS
Reprocessing	24,000 MIPS
Splitting	10,000 MIPS
TOTAL	92,000 - 146,000 MIPS

It should be pointed out that this is not the need on day one of Run II. Reconstruction needs will ramp up as the detector and accelerator are commissioned. Reprocessing can't really begin before processing occurs. Monte Carlo may or may not be a big issue early in the run and secondary and tertiary splitting probably won't happen immediately on the farm until some data sets are written out. (This splitting is not the same as the splitting that will occur as part of initial reconstruction. This splitting includes primarily things like B dataset spins.)

4 History

Providing large amounts of computing at competitive prices is something that Fermilab has been accomplishing for many years, first with the ACP I, continuing with ACP II, the UNIX farms and finally now with PC farms [1][2]. The most recent purchases of farms at Fermilab were made to provide substantial computing (26,000 MIPS) for the fixed-target runs of 1996-97 and 1999. These machines replaced and supplemented the original UNIX farms, which were purchased in 1991-1992. These older farms were used to handle the offline needs of the 1990 and 1991 fixed-target runs, and of course for Run I. CDF and D0 each used farms with approximately 3,000 MIPS to reconstruct Run I data.

The newest commercial UNIX farms, purchased in 1996 and 1997, consist of SGI and IBM workstations and small SMP's. They are of course much faster than the older workstations, have much more memory, more local disk, and are connected to a fast switch rather than to routers as in the past. The newer configuration allows for much more flexibility in processing models than in the past.

The newest farm is a PC farm[3] and consists of approximately 30 Dual 333 MHz Pentium II PC's. These newest machines are running Linux. The machines have even more memory, local disk and network bandwidth. The experiment E871 will use these systems to reconstruct their 1996-97 data and will test and prove the concepts of processing which have been developed to make use of the new capabilities that these system give us.

5 Proposed Farms

To meet the needs of the event reconstruction (48,000 MIPS) one can estimate the number of PC's which will be required. First, the MIPS ratings for many of the PC's are known. This is accomplished by running a Fermilab program (called "TINY") and using the performance to give a rating. The VAX 11/780 was initially rated at 1.0 on this scale. Some representative numbers are given in Table 2.

The Pentium MIPS scales with the clock speed almost exactly. Therefore, we will assume that a 500 MHz Pentium will be rated at 281 MIPS. We assume that 500 MHz Pentiums will be the first set of machines purchased in 1999. If we assume that all the PC's will be 500 MHz Pentiums then:

$$48,000/281 = 171 \text{ processors}$$

will be required. If these are purchased as dual machines then 86 duals would be required. Each PC would be purchased with a sufficient amount of memory to allow 2 CDF processes to run and sufficient local disk space for staging files and a fast ethernet interface for data transfer. The final decisions on these parameters must be made only after real testing is made on a prototype farm.

Table 2: Some sample CPU ratings

Machine	MIPS	Time/event(CDF code)
SGI R10000 (cdfsga)	191	
SGI R4400 (cdfsga)	84	3.02
Pentium Pro 200	113	2.16
Pentium II 266	147	
Pentium II 300	168	
Pentium II 350	197	

There are two uncertainties in the estimation of the size of the farm that is required. First, the total amount of processing power that is needed is not yet known and current estimates range up to 146,000 MIPS. Second, the Pentium will continue to speed up beyond 500 MHz and this will reduce the number of PC's required, implying that PC purchases should be delayed as much as possible. The strategy for acquisition will have to include a baseline of processors to handle initial data-taking, with increments of faster processors to provide the full processing required.

5.1 Cost

Cost is a consideration when building systems of this size. Farm processing can be the most cost-effective processing because of its high ratio of CPU to I/O and its predictability. One can estimate that a dual Pentium PC with a reasonable disk and a fast ethernet connection and 256 MB or 512 MB of memory should cost approximately $4K. If one adds $50K for the switch then the total for the baseline system of 86 duals is:

$$86 \times \$4K + \$50K = \$394K$$

This is not a huge cost and is well within the parameters of the DM-NAG for computing hardware expenditures for Run II. Additional cost for the I/O systems have to be estimated and added to this cost. This is harder to estimate at this time without having a final model for I/O systems. Uncertainties include lack of knowledge of detailed performance of input file transfer to the worker nodes and a need for a plan to handle the output files. A reasonable guess is $100K for a full I/O system for a farm this size, leading to a total baseline cost of:

$$\$394K + \$100K = \$494K$$

This will be refined as the I/O systems are specified properly and a better understanding of the network requirements and staging requirements are made with the prototype farm.

5.2 I/O bandwidth

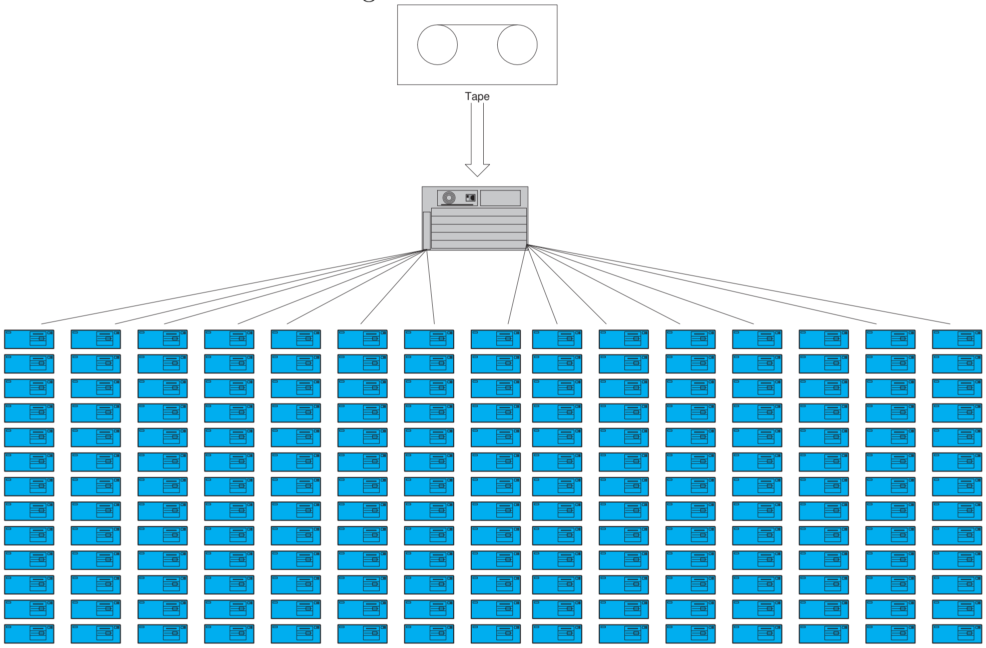
Even though these systems are going to be used for fairly CPU-intensive computing, the need for sufficient I/O cannot be overlooked. The total amount of I/O is given by the size of the events and the rate of event processing. For a steady-state rate of 28Hz that leads to a total input I/O rate of:

$$28Hz \times 250KB = 7MB/s$$

The output rate should be similar. The peak rate is higher, and depends in some detail on the model of processing. One can assume a peak input rate of 75Hz, which is 3 times the steady-state rate. Assuming an average I/O rate of 14MB/s and 86 PC's, each PC will require 163KB/s of I/O, which should not be difficult using fast ethernet. Even if the processor speed doubles, fast ethernet should suffice. In addition, two interfaces could be attached to each PC if that proved to be necessary.

The prototype PC farm will be used to study the bandwidth issues, to ensure that this I/O rate can be achieved using the technology which we are planning to deploy. An overview of the proposed farm is shown in figure 1. All of the PC's are connected to a fast switch (e.g. CISCO 5500) via fast

Figure 1: Run II Farms



ethernet. The switch in turn is connected, probably via Gigabit ethernet, to the mass storage system. It is not clear yet how this connection will be made, and this decision will be made based on what is learned from the data handling studies and the prototype farms. The I/O nodes, whatever form they take, must be designed so that they can provide the necessary I/O rates to/from the mass storage system and to/from the farms.

6 Proposed Processing Model

6.1 Processing Steps

The following processing steps are felt to be reasonably generic for a farm which is in steady-state processing of raw data from the experiment or in a state of processing a backlog of files from any source.

1. Read a list to find a file and an executable and calibrations (these can all be lists) for scheduling on the farm.

2. Copy a file to be processed to disk. (Time taken = 1GB/10MB/s = 100s.)
3. Copy the executable and any associated calibration or other files.
4. Start processing job 1.
5. While job 1 is running, perform steps 1,2,3 for job 2.
6. When job 1 finishes, copy output files to a dedicated splitting system.
7. While job 2 is running, perform steps 1,2,3 for jobs 3 and 4.
8. Continue

It should be noted that the time to process one file, given current estimates of CPU time and file size is:

$$((1GB/250KB) \times (1200MIPS - s))/(281MIPS) = 17081secs = 4.7hours$$

Note that this gives the farm sufficient time to move files around and to prepare for scheduling of jobs. This of course is true only up to a limit, and part of understanding the detailed organization of the farm is to determine this limit. If a large number of worker nodes are asking for files then some network or process or robot will eventually be overloaded, meaning that some thought has to go into optimizing the farm.

6.2 Comments on processing steps

Steps 1 and 2: This can be shown schematically in figure 2. One aspect which has not yet been decided is the hardware and software which will be used to move the data from the mass storage system to the PC (worker node). It is clear that no matter what we use there is a strong need to batch the requests to the mass storage system in order to use the tapedrives effectively.

Step 6: Step 6 includes the merging of the output files. This is probably the most complicated part of the farms, as it includes the merging of many small files for some streams, and of only a few large files for other streams. Keeping all of this in synch will be a major scheduling issue. See figures 3,4,5 and 6 for some possibilities for the merging steps. It is clear that this part of the system needs a detailed design.

Figure 2:

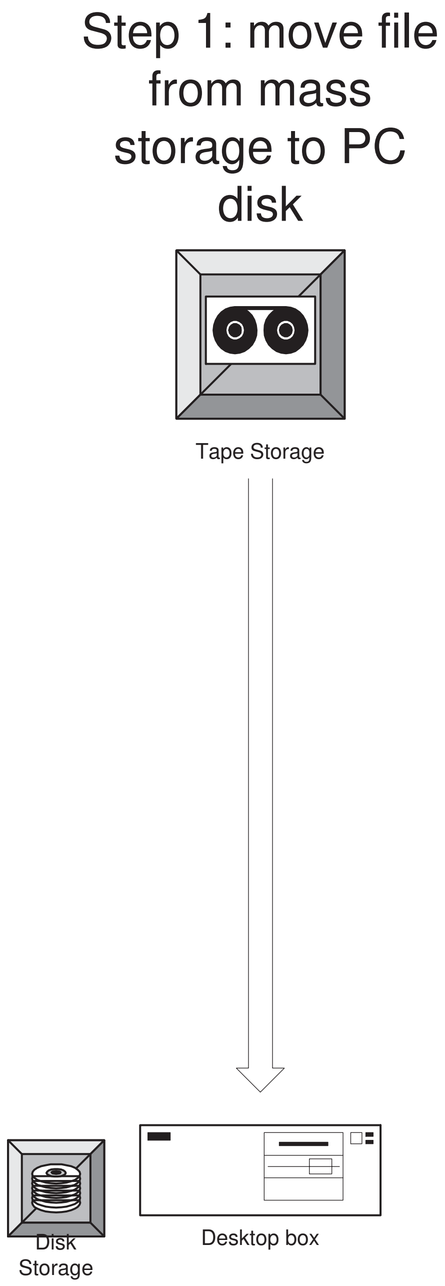

Step 1: move file from mass storage to PC disk
Tape Storage
Disk Storage
Desktop box

Figure 3:

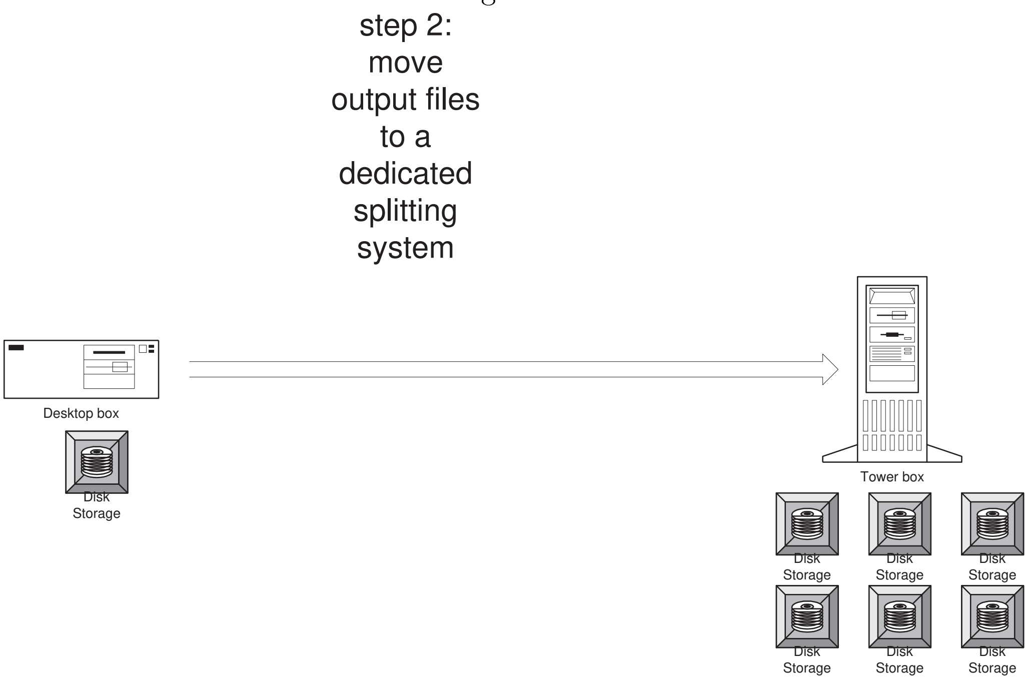

step 2:
move
output files
to a
dedicated
splitting
system
Desktop box
Disk
Storage
Tower box
Disk
Storage
Disk
Storage
Disk
Storage
Disk
Storage
Disk
Storage
Disk
Storage

Figure 4:

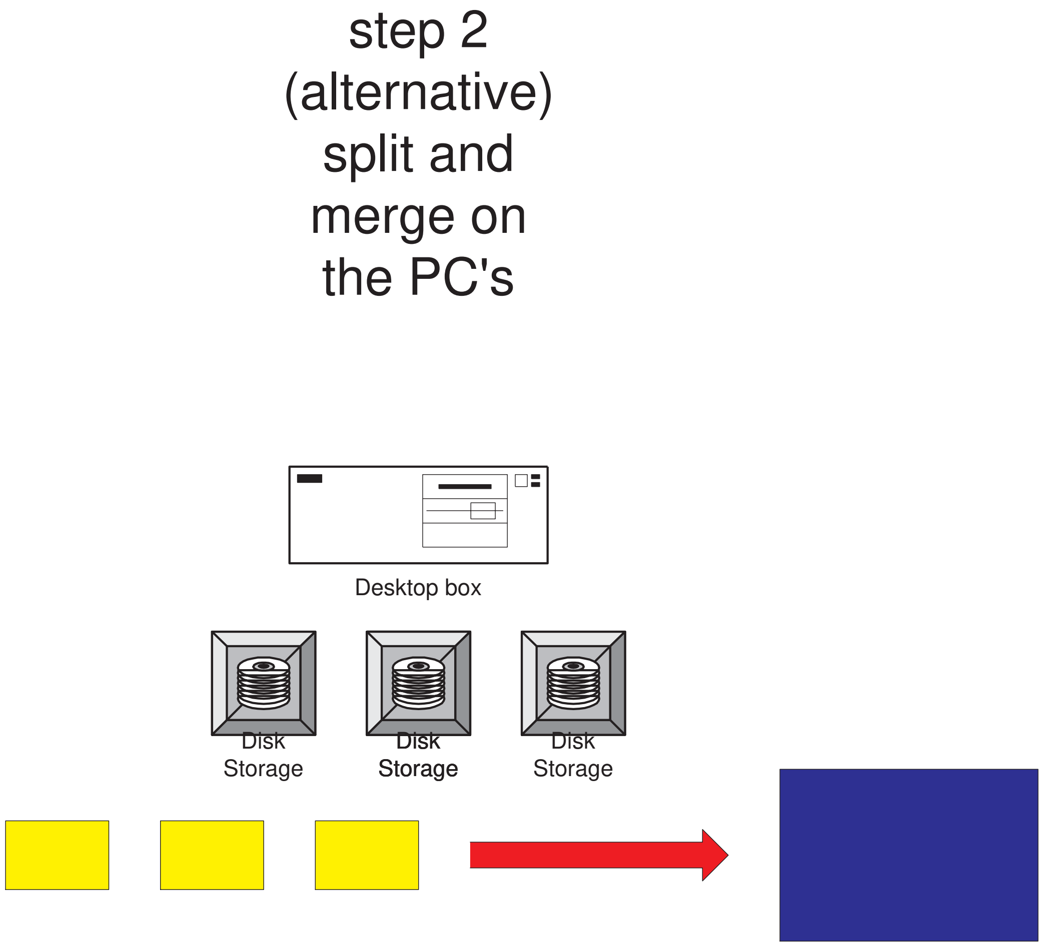

step 2
(alternative)
split and
merge on
the PC's
Desktop box
Disk
Storage
Disk
Storage
Disk
Storage

Figure 5:

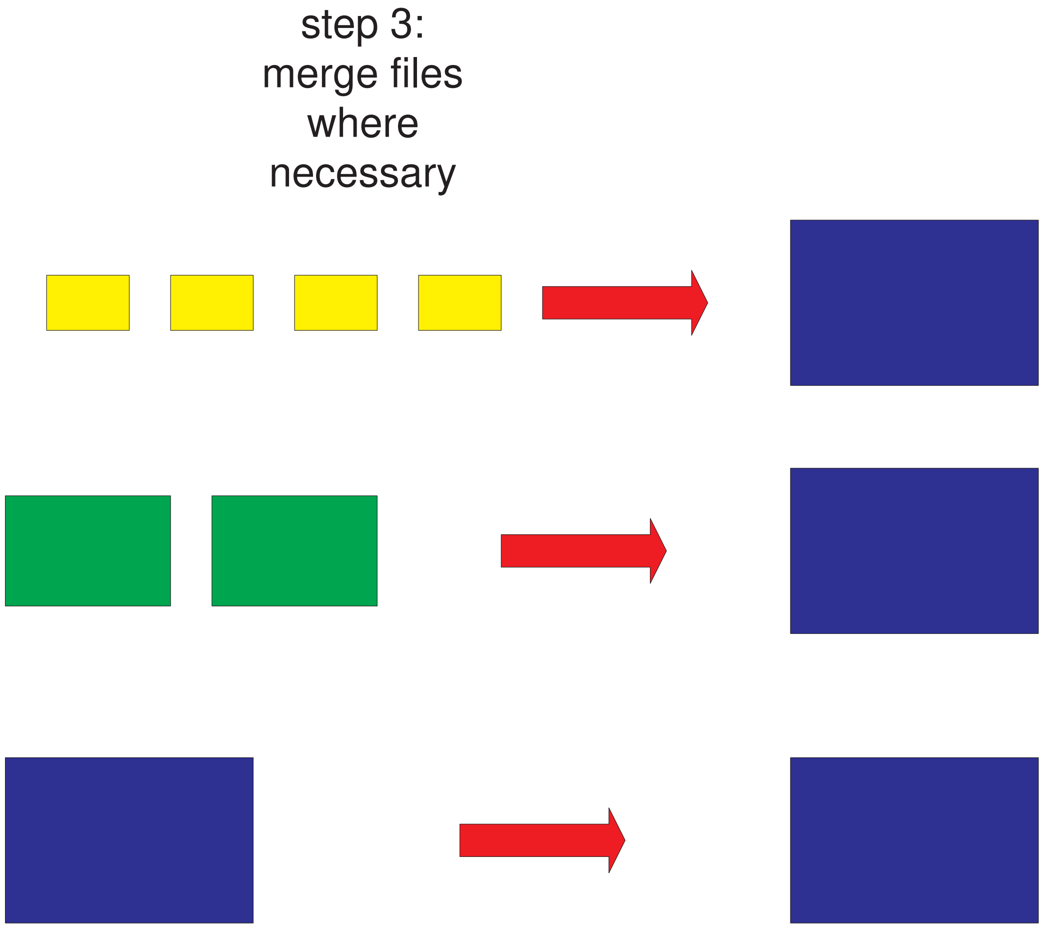
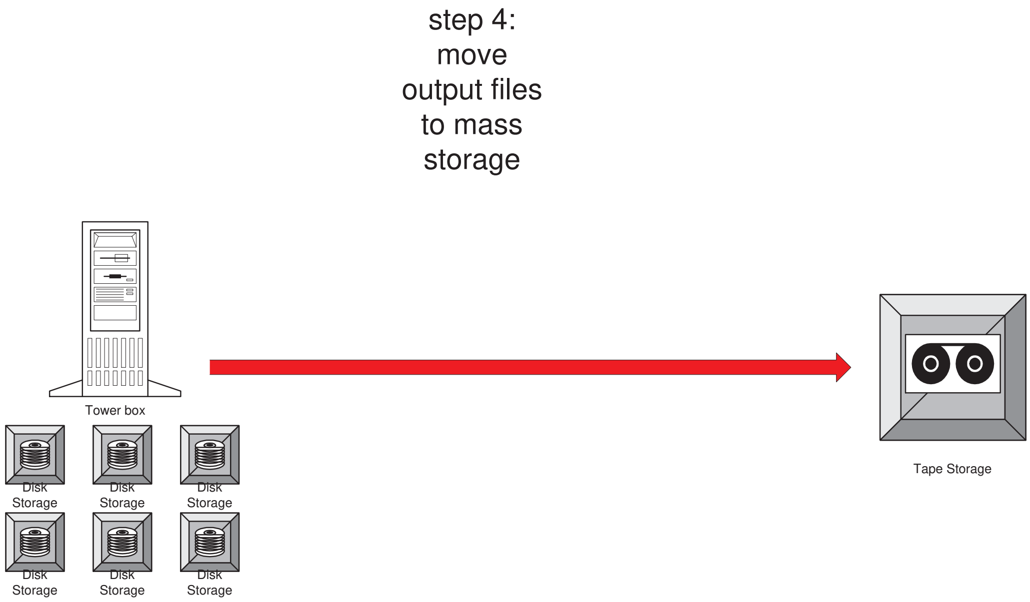


Figure 6:



6.3 Strategy for steady-state farming

There are many ways to run the farm, assuming that it is flexible enough. Some possible scenarios include:

1. Run each L3 stream one at a time on the farm, on all nodes.

2. Run all L3 streams concurrently.

3. Run some but not all L3 streams concurrently, changing the mix with time.

The best way to control the farm may be to use external control files which specify the number of processes that are to work on each stream. This could be combined with absolute minimum and maximum numbers of processes, as well as priorities for choosing when there is a conflict. Using such an arrangement should allow the people who are running the farm easy control over the way the whole farm runs. It should not be very difficult to implement such a scheme.

6.4 Executables

Ideally, the farm should run a single executable for all streams that the farm is processing. The reasons for this include:

1. This makes keeping track of the farm very easy.
2. It makes the farm simpler.
3. A simpler arrangement usually means fewer mistakes.

However, it is clear that executables will have to change. It is also clear that there will be times when more than one executable will be running on the farm. When executables do change the best strategy would be to have all streams change to a new executable at the same time.

The farm will be designed so that each node and even each process can run its own executable with its own set of constants. This will allow the flexibility to handle any arbitrarily combination of executables.

6.5 Streams

There are N input streams to the farm coming from L3. These are inclusive streams, so each stream is self-contained and because of that there will be events which are written to more than one stream. Each input stream or set of streams should be written on a dedicated set of tapes in the mass storage system. This data packing will help ensure that access to the mass storage system from the farms is as efficient as possible. The output streams of the farm need to be defined. Once the definition is known and the relative sizes are better understood it will be possible to plan for packing of the data on output tapes in the mass storage system. The goal should be to produce output which is easily analyzed to produce physics results.

6.6 Reprocessing

New executables may require some reprocessing of data. Reprocessing of large datasets should be contemplated only if the benefits (i.e. physics) are substantial. Small streams can be reprocessed often with new executables, though this does bring with it the danger of confusion when analysis results are presented from those streams. Small set reprocessing may be necessary to track the improvements that are being made or to provide the most up-to-date information for some analyses that require it. The reprocessing of large streams requires large resources and thus needs to be carefully considered and planned. Large streams should be reprocessed only if the new executable is stable and represents a significant improvement over previous versions.

6.7 Bookkeeping

Good bookkeeping is critical to the success of Run II processing. The design for any databases that will be used by Run II processing should be made taking into account all of the things that were learned during Run I about what information will be required from the production.

6.8 Validation

Validation of the results of processing on the farms is an extremely important task. We believe that the validation process should be separate from the other processes which occur on the farm and that the production group not be responsible for the validity of the results coming from the farm. The group should be focussed on pushing data as quickly as possible through the system. There is certainly good reason to run validation code along with production. Whether it gets run as part of the production executable or as a separate program needs to be worked out as part of the overall strategy of running the farms.

6.9 Monte Carlo

The farms can be used and are in fact ideal for the generation of simulated events. The small input for each job makes the I/O much simpler. The large CPU required to generate each event also makes this type of job ideal for an array of cheap PC's. Of course, there are other methods that can be used to generate Monte Carlo events, and the collaboration should plan on using resources that are appropriate for this task.

In addition to the generation of the events the farms can easily be used to reconstruct the events. One advantage of using the farms for these activities is good control of program versions, good bookkeeping, etc.

If the farms are to be used for substantial Monte Carlo generation and/or reconstruction the farms should be sized accordingly.

7 Prototype Farm

A prototype farm has been purchased for the purpose of learning how various components and parameters of the farms should be specified for the first large farm purchase in 1999. There are many variables, including the speed of the processors, the amount of memory, the amount of local disk, the network connection speed, the type of connection to the mass storage system, the processing model (especially I/O), etc. These can be tested and various measurements made with this prototype farm.

The prototype consists of 18 PC's, 4 of which are "I/O-type" and 14 "worker-type" nodes. A gigabit ethernet switch and a fast ethernet switch are both to be used in the prototype. This farm is dedicated to Run II testing, and will be available in November, 1998.

8 Startup Strategy

The strategy for getting to full-scale and fully functional farms can follow a number of different approaches. It will depend in the end on the style of the people who are working on the farms and on the availability and functionality of various components (hardware and software) which will be used. One possible strategy is the following:

1. Acquire a small prototype farm.

2. Make measurements of various characteristics of the farm, including I/O measurements to/from disk and to/from mass storage. These measurements can be made by simple programs that copy files around and do not require that the full CDF environment and executable run.

3. Get real CDF executables running on the farm, reading and writing CDF files. This will allow the study of memory use, loading of the systems, CPU times, as well as a study of the behavior of the farms with increasing load.

4. Study bookkeeping and control needed to keep the farms running and to handle various standard modes of running, including a study of handling typical error conditions.

5. Learn how to split and merge streams after files come off the farms after reconstruction. These steps (splitting and merging) may be done on separate systems or they may be logically separate programs.

6. Ramp the farms up to larger and larger systems as the hardware becomes available to ensure that any bottlenecks will be found and (hopefully) fixed before the beginning of Run II.

9 Effort Estimates

The effort estimates which are available so far are very preliminary and are not the result of careful study of the tasks which need to be performed before the farms are running and ready for Run II raw data. Nevertheless, they are a good starting point and allow us to make some initial estimates and compare those estimates to what is expected to be available during the next 2 years.

1. The requirements for CDF Run II on the farms needs to be done. This document is certainly part of that. It will take 1-2 FTE's approximately 3-6 months to fully accomplish this.

2. The prototype activity will begin in October and continue through the middle of 1999. This should occupy 1-2 FTE's.

3. The control software for running the CDF farms needs to be specified (part of #1 above) and written. This is a joint activity, which also involves software and effort from the Computing Division Farms Group. It is expected that this will take 2 FTE's from CDF and 2 FTE's from the Computing Division. The steps involved are:

   (a) specify files, executables and calibrations
   (b) submit jobs
   (c) prepare for next set of jobs
   (d) collect output
   (e) merge output
   (f) write output to mass storage

(g) error recovery

4. Once software is written and systems are running a period of debugging and tuning will be required. This should also occupy 1-2 FTE's.

5. Finally, running the farms after they are in steady-state operation will take 2 FTE's.

It should be noted that in the above effort estimates many of the jobs are separate and can be worked on by the same people at different times in the development of the farms. Therefore, the total amount of effort required is much less than the sum of the estimates made above.

10 Milestones

1. Oct 98
   (a) Production Executable (10/98)
   (b) Data Handling "Proof-of-Principle" (10/98)
   (c) Physics Analysis Software Recommendation (www.fnal.gov/projects/runii/runIIdocs/runIIsteering/pasrec/pasrec_response)
   (d) Review by Offline Godparents (10/7-8) (CDF note 4801)
   (e) Review by FNAL PAC (10/23-24)
2. Nov 98
   (a) Assemble DH prototype system
   (b) Farms Prototype (11/16/98)
      i. Running Oct Exe
      ii. Batch System/Interaction w/ CDF resource mgmt
   (c) Joint Workshop with CD/D0
   (d) Calibration DB Proposals and Prototyping
   (e) Geometry Data Base Proposals and Prototyping
3. Dec 98
   (a) Review of Analysis Hardware by ad hoc Tech. Com.

(b) Farms writing to tape robot
(c) Initital Spec for Farms Production Manager, Catalog
(d) Treatment of luminosity
(e) Revised data model proposal
(f) Data streams proposal (joint with TDSWG)

4. Jan 99
   (a) First "Full" Data handling, Farms prototypes
   (b) Remote Access Workshop
   (c) Collaboration Approval of DH Plan
   (d) Calibration DB Design
   (e) Geometry DB Design
   (f) Review by von Ruden committee (Jnt Projects and CDF)
5. Feb 99
   (a) Begin Full Prototype Integration
      i. All systems
      ii. Interactions with Online,Trigger,Detector Groups
6. Mar 99
   (a) Baselined Reconstruction
   (b) Serial Media P.O.
   (c) First Analysis System P.O
7. April 99
   (a) First Farms P.O.
8. May 99
   (a) Full prototype integration complete
   (b) Production Executable has
      i. Real geometry DB

    ii. Real calibration hooks
    iii. New version of Data Model
  (c) Data Handling System has
    i. Analysis system, Farms, Disk Cache, Tape Store, RIP
    ii. Full Resource Management
    iii. Full Production Management
    iv. Full Interactions with CDF Data Catalog, Trigger DB
  (d) Simulation
    i. CDF II with real geo DB, calibrations, DM

9. June 99
  (a) Engaged physics groups
  (b) Test Download Run I Data Set for Production
  (c) Create Large MC data samples on Farms
  (d) Begin Mock Analysis Exercises
  (e) Optimize the full prototype
  (f) Begin serious work on Validation Procedures for
    i. Code Management
    ii. Production

10. Sept 99
  (a) Run from CDF Data Logger at Full Speed
  (b) Mock Data Challenge

11. Oct 99
  (a) First Run II Hardware at FCC

12. Dec 99
  (a) Fully Functional Run II System

References

[1] Frank Rinaldo, Stephen Wolbers, Computers in Physics 7:184-190,1993.

[2] Stephen Wolbers, The Fermilab Computing Farms in 1997, Fermilab-TM-2044, 1998.

[3] Mark Breitung, et al., Processing Farms Plans for CDF and D0 for Run II, proceedings of CHEP98, Chicago IL.