SDC-91-00008

SDC
SOLENOIDAL DETECTOR NOTES

A CONSERVATIVE APPROACH TO OFF-LINE COMPUTING

K. Amako

November 30, 1990

SDC-91-00008

Computing/Software for SDC

- A Conservative Approach to Off-Line Computing -

Katsuya Amako
KEK

Nov. 30, 1990

The aim of this note is to present a base for discussions to design architectures for hardware and software of SDC off-line computing/software. The structure of this note is, first, we raise a question which, we believe, is basic to design and try to answer to it in conservative approach. Here, conservative approach means that we do not jump at technologies which are not firmed at present stage but advocated to be realized, for example, in 5~10 years. What we kept in mind is that, even we employed conservative approach in present designing, the system is flexible enough that we can integrate a new technology which may be available in future without changing its global structure. Subjects discussed in this note are very primitive and limited, and only the note will be meaningful is to trigger serious discussions.

I. Introduction

To design and realize hardware and software systems of off-line computing for the SDC experiment is a difficult but challenging task. Major difficulty comes from the fact that we must design systems which will not be obsolete at the time we start the experiment in nearly 10 years later. This means that we must somehow foresee hardware and software technologies available in distant future. Practically, however, this is an impossible task because computing technology is one of the most rapidly changing area and nobody can correctly predict what are available in 10 years later (even 5 years later). In spite of this fact, we must complete the design which is not obsolete in distant future. Other difficulty comes from the fact that the scale of the software system we must accomplish is so huge that it takes a long period of time and it is only realized by merging software activities by large number of physicists (over several hundreds) around the world. Also we should keep in

mind is that the system will be used until SSC ceases its physics activities. This means that we must design a software system which will be used more than 20 years. Establishment of a methodology to develop a software system of this kind is one of ultimate goals for software engineering. Recent progress in software engineering provides various methodologies. However, there is not established method which is appropriate for our specific application. Here, we must find our own approach to solve the problem.

The first milestone of our activities is to design architecture of hardware and software of SDC computing and prepare a proposal in 1 year later. To attain the goal, the first of all, we need to establish global designs of both hardware and software architectures which serve as a common base for further details. This, we believe, is very important at present stage because people have their own experiences in past and have different opinions in designing and realizing our hardware/software systems. If we start discussions of more details of the systems (ex. Should we use parallel/vector computing? Which code management system should we user? etc) without such common base, our efforts may collapse. The aim of this note is to present a base for discussions to construct such a common base. Of course we do not believe that items discussed in this note are exhaustive nor complete. We only discussed about very primitive and limited questions from which we can start to. We hope this note bring up hot discussions and criticisms.

Major questions we try to attack in the note are:

- a global scheme of hardware
  - >> where do we put CPU resources ?
  - >> where do we put storage resources ?
  - >> what are required to network ?
- a global scheme of software
  - >> how to develop a software system ?
  - >> what are basic requirements to be satisfied by the system?
  - >> how to select tools to use for software development ?

Major items considered are:

- On-line Data Acquisition
  (Only relevant to the off-line stage)
- DST Production
- Detector Simulation
- Physics Analysis

II. On-Line Data Acquisition

- Only relevant to the off-line stage -

[II.a] Hardware - Data Storage for Raw Data

From a conservative viewpoint, we should store permanently all raw data which are triggered at the data acquisition level. An obvious reason to do this is that our experiment surveys a new regime of the energy scale, therefore, we should be free from any biases as much as possible especially at the beginning of the experiment. The following are questions related to data storage devices for raw data and possible answers from a conservative viewpoint.

Q1. What is the required writing speed ?

A. According to our EoI, the trigger rate and the data size at the output stage of the 3rd level trigger are

event rate = 10 ~ 100 Hz,
data size = 1 MByte.

Here, the 3rd level trigger is assumed to be the final trigger at the on-line stage. By definition, this trigger requires high level information from major detector elements (e.x. complete tracking information, ...) to filter out background events. This trigger is designed to be realized in the on-line farm. As experienced in many high energy physics experiments, it is usually impossible to prepare the 3rd level trigger from the very beginning of the experiment, because it requires detail studies of the performances of major detector components using real data. Realistically we should assume that, also in our experiment, the 3rd level trigger algorithm is first tuned at the off-line level, and then it is moved into the on-line level. Therefore, from a conservative viewpoint, the storage device has to handle the trigger rate of the 2nd level trigger of

event rate = 100 ~ 1000 Hz,
data size = 1 MByte,
bandwidth = 100 MByte/sec ~ 1GByte/sec.

Q2. What are possible devices to satisfy the above requirement?

A. Because storing raw data is for archival purpose, we need not use a random access device. A sequential device is enough for the present purpose. One of candidates which will be available in near future is SONY DIR-1000 [*]. This has the following feature.

- Digital VTR,
- Recording Speed = 32 MByte/sec,
- Capacity = 100GByte/tape.

To handle the event rate at the 2nd level trigger above, we need

# of tape drives = 3 ~ 30,

# of tapes = 90 ~ 900 tapes/day.

Other possibility is [**]

- Optical Tape (Creo Products)
- Recording Speed = 3 MByte/sec
- Capacity = 1TByte/tape

To handle the event rate at the 2nd level trigger, we need

# of tape drives = 30~ 300,

# of tapes = 9 ~ 90 tapes/day.

It is clear from the above calculations that we cannot endure the trigger rate of 1000Hz at the 2nd level. The event rate at the 2nd level output should be less than 100Hz at least at the beginning of the experiment. Applying the same argument, even after we successfully install the 3rd level trigger at the on-line stage, the final recording rate of events should be less than 100 Hz. Of course we anticipate criticism that this is too conservative: a media of much higher speed/density is definitely available at the time we start experiment, and using this we can allow a much higher trigger rate. However, we should always keep in mind that the amount of data from the on-line stage becomes larger, the CPU load at the off-line stage becomes heavier, and this may introduce a delay to obtain physics results. From the viewpoint of efficiency to get physics results, smaller size of raw data is better.

We definitely need an auto-loading system for these tapes.

[*] H. Fujii, talk at the electronics/DAQ subgroup meeting in the SDC collaboration meeting at LBL, Nov 5-7, 1990.

[**] D. Baden and R. Grossman, "A Model for Computing at the SSC", SSCL-288.

Q3. Do we need to distribute these tapes among our collaboration institutes?

A. Because storing raw data is for archival purpose, we need not distribute these tapes among our collaboration institutes. These should be kept in the site where DST productions are taken place.

[II.b] Software

[II.b.1] 3rd Level Trigger

As described in the answer to Q1 of [II.a], we expect that a filter algorithm used in the 3rd level trigger is first tuned at the off-line level, and then move it into the on-line level. Therefore, we cannot separate code developments of programs used in the on-line farm and the off-line stage.

Q1. How to develop programs which are transparent in both the on-line farm and the off-line stage ?

A. To realize this, we need to follow the general principles listed below for development of program packages which will be used in both the 3rd level on-line farm and the off-line stage.

1. To apply the same coding rules.
2. To use the same data structure.
3. To use the same parameter database (ex. geometry parameters,...).
4. To use the same utility tools (ex. graphic language, histogramming packages, ...).

[II.b.2] On-Line Monitor Programs

On-line monitor programs to check performance of detector components and analysis packages of these detectors at the off-line stage (3rd level trigger) have close relations. This is because these packages access to same raw data, and are expected to be developed by the same detector sub-group. Although it is not must but desirable to keep program packages used in the on-line monitor transparent to the off-line stage. To realize this, we should follow the same principles as list in the answer to the previous question.

III. DST Production

In a macroscopical viewpoint, a DST production program in collider experiments can be divided into the following two steps.

Step 1 : First, we execute detail analyses, which consume too much CPU time to fit into the on-line stage, of data from detector components (ex. fine-tracking, vertex finding, fine calorimeter-clustering, etc). Information produced by these analyses is used as a base for event selections at the next step.

Step2 : Then, we apply various kind of physics filters to select events. These filters are optimized to specific physics processes (ex. select events which have electrons, large missing Et, etc), and we produce multiple series of DSTs at the end of this step. Each series of DSTs has enriched samples of events which serves as a base for further studies of some specific themes of physics.

In principle we can produce final DSTs directly from raw data by executing above two steps in one program. However, if we do this, we may waste a huge amount of CPU time to execute the 1st step. This is because we need to do the step-1 again if a new physics idea comes up and we want to do the DST production for it. From a conservative view point, it is desirable to separate the DST production into two stages as follows.

1st-DST Production

In this production, we execute the step 1 above. This produces 1st-DSTs which have raw data and results of detail analyses of data from major detector components. Before writing data to 1st-DSTs, obvious background events will be thrown away by applying a filter program.

2nd-DST Production

Applying the filter programs in the step 2 to 1st-DSTs, we produce series of 2nd-DSTs. Each series of DSTs has enriched samples of events which serves as bases for studies of some specific physics themes. Most physicists access to these data to do their physics analyses.

[III.a] Hardware : 1st-DST Production

[III.a.1] CPU Resource

The following are questions related to the CPU resource for the 1st DST production and possible answers from a conservative viewpoint.

Q1. Where do we execute mass productions of 1st DSTs ?

A. We should execute mass productions of 1st DSTs at a particular site (obvious selection of the site is SSCL, therefore let's assume it is SSCL in the following discussion). We should NOT do any productions of 1st DSTs outside SSCL. The major reasons for these are

1) We do not want to transfer a huge amount of raw data between collaboration institutes.
2) This makes the bookkeeping of DSTs much simpler.
3) We need not worry about a portability of the production program.
4) We need not worry about inconsistency of program versions used in productions at different sites.
5) This makes management of the parameter data base for productions much simpler.

Q2. Don't we need any CPU resources outside SSCL for 1st DST production ?

A. We don't need CPU resources outside SSCL for mass productions. However, although small comparing to above, we need local CPU resources at collaboration institutes (especially outside the U.S.) to develop software components for the 1st DST production. There may exist an extreme opinion that we use an extra-high-speed network link (ex. GBit-link) in future and do our all software activities on centralized CPUs at SSCL. In this case, we can save manpower to maintain computing resource outside SSCL, and this is the the most cost effective way to manage the resources in our collaboration. However, even if we can use a reliable and cheap GBit-link in future, the physical delay (propagation delay) of response to an end-user who are far away from SSCL (especially outside the U.S.) may cause a serious deficiency for developing his program. No bothering delay of response from a computer is must in program development. Extending this argument to the extreme end, there may exist an opinion that we need to install a certain amount of

CPU resource at every individual institutes in our collaboration. In this case, the overall cost of off-line computing becomes huge. A possible resolution to these arguments is to have a remote computer center at each major laboratory/institute in our collaboration so that institutes in small scale can access to a near by center with a high-speed-network-link.

[III.a.2] Data Storage

Q1. Do we need to distribute 1st DSTs among collaboration institutes?

A. We should not distribute 1st DSTs among collaboration institutes. Any activities of physics analyses in our collaboration should be started from 2nd DSTs.

Q2. What kind of devices do we use to store data from the 1st DST production ?

A. If we will have a huge progress in the technology, it is desirable to store 1st DSTs in random access devices. However, sequential devices are still OK for the present purpose. A possible media is one used for the storage of raw data.

[III.a.3] Network

Q1. Are there any requirements to network for executing 1st DST productions ?

A. Before starting the production, we need to tune up detector parameters. Tuning parameters for each detector element is expected to be done by the corresponding detector subgroup. Persons who have a charge to maintain parameters need to access to raw data (or calibration data) stored in SSCL and to process them using CPUs in SSCL. It is desirable that these persons can control and update a database for the detector parameters at SSCL from collaboration institutes through network. The network speed of the order of T1 link is enough for this purpose.

[III.b] Software : 1st-DST Production

Although productions of 1st DSTs are executed at the particular site (SSCL), components of the 1st DST production program will be developed at many institutes around the world. The most difficult but challenging point to complete the 1st DST production program is how to unify these activities and merge into one program. As described in introduction, to establish a methodology to develop a software system with large number of people is one of ultimate goals for software engineering and it provides number of methods (ex. relatively popular one is SA/SD). However, as is already experienced by some high-energy experiment groups, it is an illusion that we can completely solve the problem by just employing an innovative software tool. Major difficulty in developing a software system with many people is coursed by human-relation problem which cannot be solved by just using a new software tool. Therefore, what is the most important is that, first of all, we must establish an organization for software development of our group. Whether we can establish a well organization or not is a critical factor for a success of our experiment. This does NOT mean that we should keep away from tools software engineering provide. Instead, we should try to use and evaluate them under a well organized team.

Q1. How to organize a off-line software development team?

A. A possible scheme of the organization for the off-line software team is

```
Team Managers  ----------------+--- On-Line 3rd Level Trigger Coordinators
  (2 persons)                  |                 L------ Development Teams
                               +--- 1st DST Production Coordinators
                               |                 L------ Development Teams
                               +--- 2nd DST Production Coordinators
                               |                 L------ Development Teams
                               +--- Detector Simulator Coordinators
                               |                 L------ Development Teams
                               +--- Analysis Tool Development Coordinators
                               |                 L------ Development Teams
                               +--- Parameter Database Coordinators
                               |                 L------ Development Teams
                               +--- Network Coordinators
```

```
|                    L------ Development Teams
+--- Library Tool Development Coordinators
|                    L------ Development Teams
+---  ......
:
```

Under the coordinators in each category in right side of the above structure, there are code development teams. See for example in the answer to Q2 below.

Q2. How to organize a development team for the 1st DST production program?

A. A possible scheme of the organization is shown in the below. We assume that each component of the 1st DST production program will be developed by the corresponding detector subgroup.

```
1st DST Production  ------------+--- Overall System Development Team
Coordinators (2 persons)        |
                                +--- Barrel Calorimeter Coordinator
                                |                L----- Development Team
                                +--- End-Cap Calorimeter Coordinator
                                |                L----- Development Team
                                +--- Central Tracker Coordinator
                                |                L----- Development Team
                                +--- ......................
                                :
```

Q3. How to chose tools to develop software ?

A. Selection of a tool for developing programs always introduces a hot discussion and sometime it leads into a serious disagreement. The list of tools which may induce such hot discussions is

- language selection (Fortran77, Fortran90, C, C++, ...)
- code management tool (Patchy, CMZ, CMS, SCCS,....)
- program development tool (SA/SD, ...)
- graphic tool (GKS, PHIGS, PEX,...)

:

Difficulty in selections of these tool is that we usually cannot provide an appropriate measure to compare possible choices. Usually what an advocator of a new tool argues is that it is proved to be very effective in his own test case. Because it is not obvious that it is also true in our specific application, we may fall into endless disputes. If we are able to prepare our own appropriate measure, we can judge pros and cons of selections in qualitative and quantitative manner. A candidate of such measure is a miniature version of our overall software system which is able to execute analysis routines of major detector components (outer tracker, barrel calorimeter,...) using simulated data generated by a detector simulator. We propose to construct it as soon as possible. The size of the program should be kept in a reasonable size so that we can try to rewrite it using a new tool. A good candidate of such program is SDC-Shell [*]. To construct components of this program, it is desirable to use tools which are familiar to most of us (ex. Fortran77, Patchy, no SA/SD, GKS,...). This makes a comparison easier. We should not hesitate to try apply new technologies to this measure. We should spend a certain amount of manpower to investigate new tools for 2~3 years of the beginning stage of the program development . After these investigations, we should freeze technical choices and start to develop a system for final usage.

[*] A program which is under development by Shell Working Group chaired by Y. Kunori.

Q4. How to make standard rules for code developments?

A. We believe that rules should be established by organizing a special task force which is composed of actual program writers in our collaboration. The selection of members should be done by coordinators listed in the answer to Q1 in [III.b]. Rules to be established are

1. Coding rules
2. Data Structure inside a program
3. File I/O format for data output
4. .....

Q5. Do we need to worry about portability of the 1st DST production program ?

A. We need NOT worry about portability of the 1st DST production program on various kind of machines because we use a dedicated computer farm for the

production. This enable us to tune the program as fast as possible on the dedicated farm. The process speed of the DST production is one of major factors to determine the speed to obtain final physics results. HOWEVER, we need to write a component of the program which is expected to be used in later stage (ex . track finding routine,...) by following the common coding rules.

[III.c] Hardware : 2nd-DST Production

[III.c.1] CPU Resource

The following are questions related to the CPU resources for the 2nd DST production and possible answers from a conservative viewpoint.

Q1. Where do we execute mass productions of 2nd DSTs ?

A. We apply the same principle employed in the 1st DST production, i.e., we should execute mass productions of 2nd DSTs at a specific site (obvious selection of the site is SSCL). We should NOT do any productions of 2nd DSTs outside that site. An outcome of this principle is that we need not transfer 1st DSTs outside SSCL. Although the amount of 1st DSTs is smaller comparing to raw data, it is still too large to transfer them between collaboration institutes.

In 2nd DST production, there are multiple paths (ex. production of electron enriched DSTs, production of missing Et DSTs, etc). We also expect that, motivated by new physics ideas, new paths and their executions will be proposed by physics subgroups among our collaboration institutes. To satisfy these demands, 2nd DST production should be able to execute from outside SSCL. It is obvious that we need a well defined rules and coordination to handle these demands.

Q2. Don't we need any CPU resources outside SSCL for 2nd DST productions ?

A. We don't need any CPU resources for mass productions of 2nd DSTs. However, although small comparing to above, we need a local CPU power at collaboration institutes (especially outside the U.S.) to develop and test filters for

the 2nd DST production. Also remote computer centers of the major laboratories in the collaboration should have a certain amount of CPU power for this purpose. For the reason why we do not use CPU resources at SSCL, see the answer to Q2 in [III.a.1].

[III.c.2] Data Storage

Q1. Do we need to distribute 2nd DSTs among collaboration institutes?

A. We should distribute a full set of 2nd DSTs to the remote computer centers in our collaboration. We also need to distribute the DSTs to other institutes. In this case, we need not distribute whole sets but send data which are related to a specific physics an institute is interested in. Of course there is a choice that we do NOT distribute 2nd DSTs. In this case, activities of physics analyses outside SSCL have to depend totally on network. Because of the argument in the answer to Q2 of [III.a.1], we do not employ this approach.

Q2. What kind of devices do we use for the storage of 2nd DSTs ?

A. We expect a serious bottle-neck in accessing data if these data are stored in sequential devices. Possible courses of a such bottle-neck are

- shortage of mounting devices,
- multiple users want to access to the same data.

Therefore, there is no question to use random access devices for storing 2nd DSTs at SSCL and other remote computer centers. Especially, at SSCL, we need a random access devices which is large enough to store whole 2nd DSTs. A possible choice of the devices for this purpose is RAID[*].

[*] See for example, D. Baden and R. Grossman, "A Model for Computing at the SSC", SSCL-288.

[III.c.3] Network

Q1. Are there any requirements to network for executing 2nd DST productions ?

A. As already mentioned in the answer to Q1 in [III.c.1], it is important that we can initiate and control productions of 2nd DSTs from outside SSCL. To realize this, we need to access to the production farm at SSCL from collaboration institutes through network. The speed of T1 link is enough for this purpose.

Q2. Do we need a high speed network to distribute 2nd DSTs among our collaboration institutes?

A. It is NOT necessary to use a high-speed link for distribution of 2nd-DSTs to institutes outside SSCL. To transfer large amount of data, we should use a tape media which is used for storage of raw data and 1st DSTs. Because the capacity of such media is so large that (see, the answer to Q2 of [II.a]) , comparing to use network, it is much faster and reliable to ship them by, for example, air cargo.

[III.d] Software : 2nd-DST Production

Q1. How to organize a software team for 2nd DST production?

A. A possible scheme of the organization is shown in the below. The basic structure is same as in the 1st DST production except the following point. In the organization for the 1st DST production program, each development team is expected to be established from the corresponding detector sub-group. In the 2nd DST case, there is not an obvious group from which each development team is organized. How to organize every development teams for 2nd DST production is critical issue in overall organization of the SDC group, and it is not a subject the computing/software group need to handle.

```
2nd DST Production  -----------+--- Overall System Development Team
Coordinators (2 persons)       |
                               +--- Electron Events Coordinator
                               |                 L ----- Development Team
                               +--- Muon Events Coordinator
```

```
|                    L ----- Development Team
+--- Missing Et Events Coordinator
|                    L ----- Development Team
+-- ......................
:
:
```

Q2. How to chose tools to develop software?

A. The same answer in Q3 of [III.b] is applicable.

Q3. How to make standard rules for code developments?

A. The same answer in Q4 of [III.b] is applicable.

Q4. Do we need to worry about portability of the 2nd DST production program ?

A. The same answer in Q5 of [III.b] is applicable.

Q5. Do we need to worry about portability of data in 2nd DSTs?

A. Because 2nd DSTs are accessed by most physicists in collaboration institutes over the world, it is essential to guarantee readability of data in 2nd-DSTs in various kind of computers

IV. Detector Simulation

[IV.a] Hardware

[IV.a.1] CPU Resource

The following are questions related to the CPU resource for executing detector simulations, and possible answers from a conservative viewpoint.

Q1. Should we concentrate CPU resources at SSCL to produce detector simulation data?

A. Comparing to the CPU resources for 1st/2nd DST productions, it is not obvious whether we should or should not concentrate resources at SSCL to produce detector simulations data.

(Pros to the concentration)
- It is easy to maintain consistently the version of the detector simulation program which is used in mass production of data. If productions are done at multiple institutes, we always need to worry about which version is used and what conditions they are generated.
- The bookkeeping of generated events is easy. If productions are done at multiple institutes, we always need to bookkeep these because we do not want to do duplicate efforts. Of course this is technically possible but it is not simple.
- To obtain final simulated events, we need to pass generated data to the 1st/2nd DST production programs. If we execute jobs of detector simulation outside SSCL, we need to transfer data to SSCL because we can only execute mass productions of 1st/2nd DST at there.

(Cons to the concentration)
- In macroscopic view, there are two kinds of demands to do detector simulation:
    1) routine productions of simulation data (ex. for luminosity calculation, for standard physics, etc).
    2) study for new physics ideas which will prompt from various physics activities among our collaboration institutes.

It is a reasonable choice that we execute simulations for activities related to 1) at SSCL. However, If we also have to execute all detector simulations for 2) at SSCL, we may encounter a difficulty. This is because demands in 2) occur randomly and, to execute simulations for these, we need a coordination to assign their priorities - if we have an infinite amount of CPU resource in SSCL, we need not worry about this, but in reality, it is always limited. Here, problem is that a priority assignment is not an obvious task, and it may introduce a danger that a new important physics idea may get a low priority. The major institutes in the collaboration should have enough computer resources which enable them to pursuit their physics interests independently.

A possible answer to the question about CPU resources for detector simulations is
- put enough CPU resources in SSCL to perform routine productions,
- put CPU resources which are expected to be enough to handle demands to do detector simulation for new physics ideas at major collaboration institutes (remote computer centers).

[IV.a.2] Data Storage

Q1. Do we need to distribute simulation data among collaboration institutes?

A. We should distribute data produced in SSCL and other collaboration institutes by requests.

Q2. What kind of devices do we use for the storage of simulation data ?

A. We should employ the same devices as used for 2nd DSTs.

[IV.a.3] Network

Q1. Are there any requirements to network for executing productions of detector simulation data ?

A. There is no special requirement.

[IV.b] Software

Q1. How to organize a software team for the detector simulator?

A. We employ the same structure as used in the 1st DST production, i.e., each component of the detector simulation program will be developed by the corresponding detector subgroup.

```
Detector Simulator ------------ +--- Overall System Development Team
Coordinators (2 persons)        |
                                +--- Barrel Calorimeter Coordinator
                                |                  L ----- Development Team
                                +--- End-Cap Calorimeter Coordinator
                                |                  L ----- Development Team
                                +--- ......................
                                :
                                :
```

Q2. Do we use GEANT for the standard SDC detector simulator?

A. For simulation activities related to EoI/LoI/Proposal, it is reasonable to use GEANT. However, it is desirable to develop a dedicated detector simulator when the design of our detector is fixed. Major reasons are

- GEANT is a general purpose simulator and has functions which we don't need for our specific application.
- GEANT is slow to trace particles especially when we define a multi-layer geometry in it. This may cause a serious problem when we want to do mass productions of simulation data.

To construct a detector simulator which is optimized to our specific purpose, there are two selections:

1. To develop a new original detector simulator.
2. To change GEANT so that it is optimized to our specific purpose.

At the present stage it is not obvious which selection we should adopt.

V. Physics Analysis

[V.a] Hardware

[V.a.1] CPU Resource

Q1. Where do we put CPU resources for physics analyses?

A. It is obvious that every institutes in our collaboration should have enough local CPU resources to do their own physics analyses. There may exist an extreme opinion that we should use the centralized CPU resources at SSCL for all analysis activities. The answer to this objection is mentioned in the answer to Q2 of [III.a.1]. Imagine if you use PAW on a machine far a way from your terminal through network.

Q2. There will be demands to do further reductions of data starting from 2nd-DSTs - production of 3rd DSTs , 4th.....). Which CPU resources do we use for these demands?

A. We should use the CPU power at SSCL and near by remote computer center .

[V.a.2] Data Storage

Q1. Do we need to distribute 3rd/4th/... DSTs among collaboration institutes?

A. We should distribute them by requests.

Q2. What kind of devices do we use for the storage of 3rd/4th/... DSTs ?

A. We should employ the same devices as used for 2nd DSTs.

[V.a.3] Network

Q1. Are there any requirements to network for doing physics analyses?

A. There is no special requirement.

[V.b] Software

In principle, we need NOT set any rules nor boundaries to develop physics analysis programs which use data in 2nd DSTs. Each physicist has a right to develop his own software in his way from 2nd DSTs.

Q1. Should the SDC computing/software group provide tools for each physicist to do his physics analysis?

A. Although each physicist can do his physics in his own way, following standard tools should be supplied to help his activity.

- A standard program by which a user can play physics analyses (ex. a tool like PAW).
- Common libraries (high level graphic interface, statistical library, etc).

We need to guarantee portability of these tools on various kind of computers.