GEM TN-91-00014

GEM Computing Meeting

SSCL

September 4, 1991

Abstract:
Agenda, minutes, and available transparencies of meeting.

Computing for Simulation and Analysis Meeting

SSCL, Main Building (Bldg. 4) Room B110 (CCD conf.room)
(9:00 a.m. Sept. 4th)

The meeting was informal, with talks and time for discussion of the topics raised.

9:15 a.m. Space at Detector and Central Facility -- G. Yost
SSCL Organization for Computing and projections -- L. Cormell
9:45 a.m. Discussion of above issues and:
Projection of use of PDSF for FY92 (the planned expansion of this facility may be postponed or scaled back to direct funds elsewhere). Needs for space, manpower etc. in the detector area, and in the central facility.
10:15a.m. BREAK
10:30a.m. Approaches to computing at collider detectors:
L3 approaches to computing -- J. Branson, R. Mount
11:30a.m. D0 approach -- R. Raja (FNAL)
12:30p.m. LUNCH
2:30 p.m. Parallel farms (Hypercube,Touchstone) L. Gladney/Lockyer (Penn)
3:30 p.m. SLD approach -- Tony Johnson

Minutes

1. G.P. Yost (SSCL) emphasised the need for specific and speedy input to the specifications for the IR and the West Campus, in terms of space for GEM computing; this is rapidly solidifying and we may have inadequate space unless we develop a plan. (Yost, Cormell, McFarlane and Mount will work on this.)
2. Larry Cormell (SSCL) described the SSCL organization for support of computing, particularly the Physics Research Division's efforts. He also presented projections of computing power at SSCL (in PDSF).
3. Richard Mount (CalTEch) described the L3 approach to computing
4. Rajendran Raja (FNAL) described the D0 approach
5. Larry Gladney (Penn) described the effort at Penn/Lincoln to use Intel multi-processor architectures, particularly for on-line computing. The Hypercube and Touchstone toroidal mesh will be tested. I/O possibilities were described (basically be putting I/O nodes in the mesh).
6. Tony Johnson (Boston) described the SLD system, using Jazelle and IDA to provide a highly interactive system with graphic user interfaces, in data collection and analysis.

Our thanks are due to R. Raja and L. Gladney, who are not GEM members, for their presentations.

One point that was emphasised by several speakers was that communication between modules of a system should be by data structures (Zebra or Jazelle) rather than COMMON blocks. Defining the data structures early on greatly helps the creation of the system.

The group defined two immediate tasks: preparation of a space/facility specification, and preparation of a contribution to the LOI.

K. McFarlane.

GEM Computing Meeting 9/4/91

Name	Institution	Email
George Yost	SSCL	SSCVX1::YOST
Richard Mount	CALTECH	MOUNT@LEPICS (BITNET)
NURAL AKCHURIN	UNIV OF IOWA	NURALA@FNAL
BRENT MOORE	U. OF MISSISSIPPI	MOORE@DON.PHY.OLEMISS.EDU
BRIAN SCIPIONI	SSCL	brian@helium.ssc.gov
Shelly Fisher	IBM	MFisher1@DalvmIC1.VNET.IBM.COM
R. J. BLANKENI	IBM/HOUSTON	
LARRY CORMELL	SSCL	CORMELL@SSCVX1
TONY JOHNSON	BOSTON U.	TONYJ@SLACVX.SLAC.STANFORD.EDU
Rajendran Raja	Fermilab	RAJA@FNAL

COMPUTING SPACE NEEDS

2 MAIN VENUES: GP Yost

THE IP

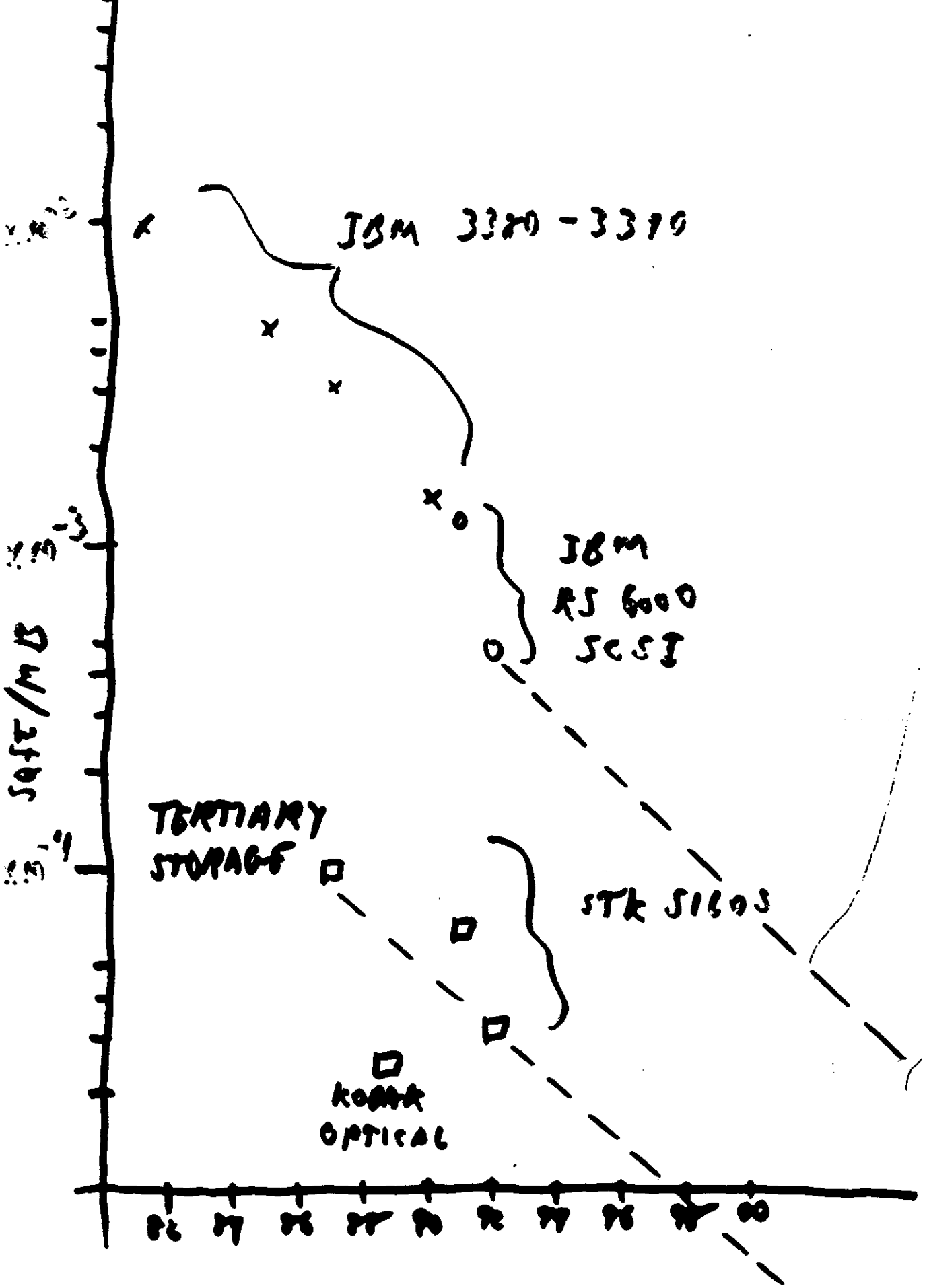
THE WEST CAMPUS

HOW MUCH SPACE?

HOW CLOSE TO THE SR (CABLE LENGTH)?

WHAT PHYSICAL REQUIREMENTS (A/C, RAISED FLOORS)

NEED INPUT INTO PLANNING PROCESS BY END OF SEPT


IBM 3380 - 3390
IBM
RS 6000
SCSI
TERTIARY
STORAGE
STK SILOS
KODAK
OPTICAL
$/MB

ADD SPACE FOR ACCESS,
SERVICE:
factor 2
⇒ 45k ✓

ADD SPACE FOR CONSOLES,
WORK AREA, USER AREA,
OPERATIONS, SYSTEM
MANAGEMENT, EXPANSION,
NETWORK/COMMUNICATION
Another factor 2

⇒ 90k ✓

A SAMPLE CALCULATION

ASSUME 1997 IS WHEN WE BUY STUFF

DISK SPACE 1×10^{-4} sqft/MB

TERTIARY STORAGE 1×10^{-5} sqft/MB

ASSUME 10 TB magnetic STORAGE (secondary)

2 PB (2×10^{15} Bytes) tertiary

$\Rightarrow 10 \times 10^{6} \times 10^{-4} = 1000$ sq ft secondary

$2 \times 10^{9} \times 10^{-5} = 2 \times 10^{4}$ sq ft tertiary

ASSUME 10^{6} MIPS compute servers at IPI for off-line

If 500 MIPS/cpu, 8 cpus/10 sq ft

$\Rightarrow$ 400 MIPS/sqft

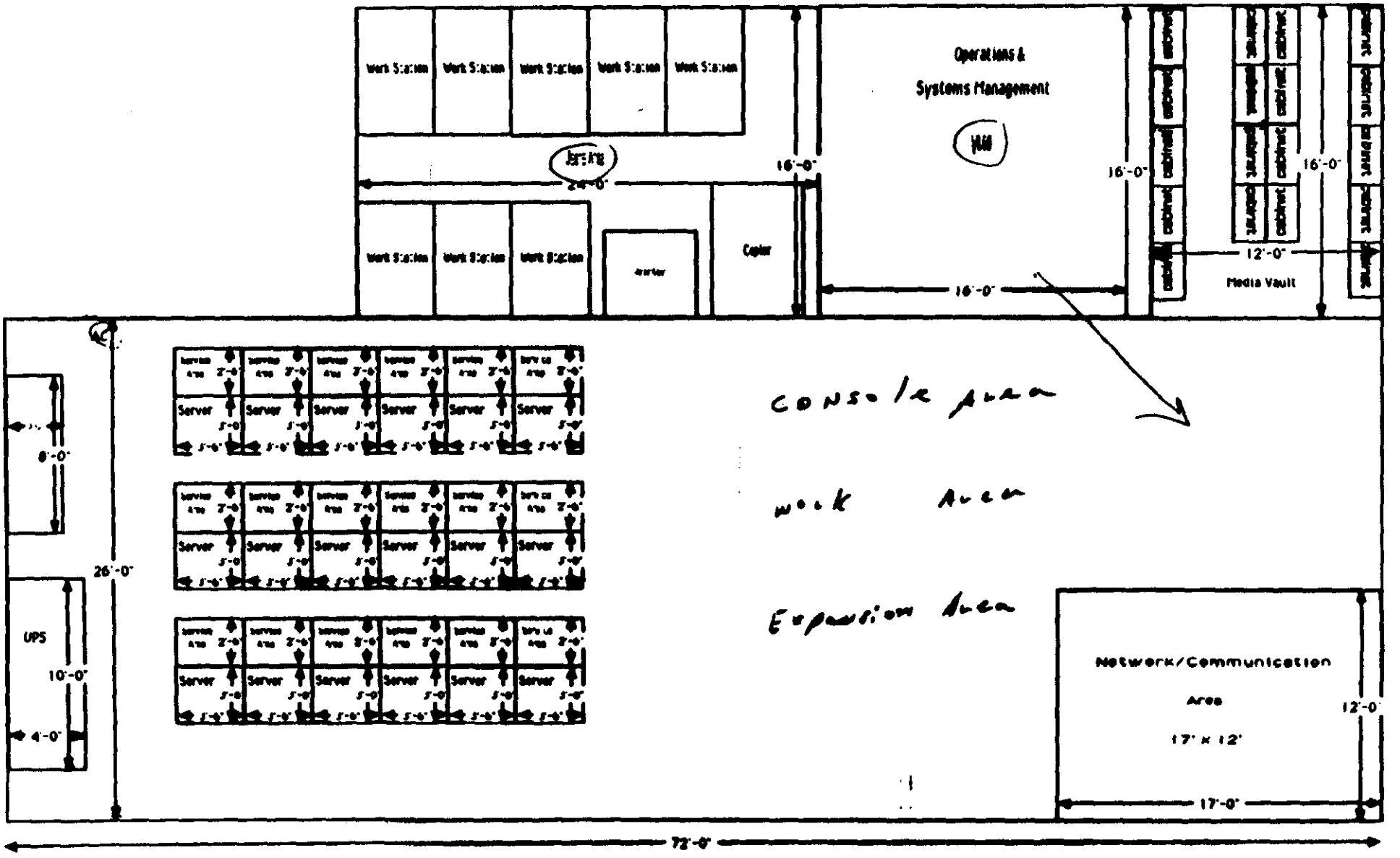
$\Rightarrow 10^{6}/400 = 2500$ sq ft cpu

TOTAL:

1000	BOXES ONLY
20000	
2500	
23500 sq ft	

Satellite

Nicell Ramsey

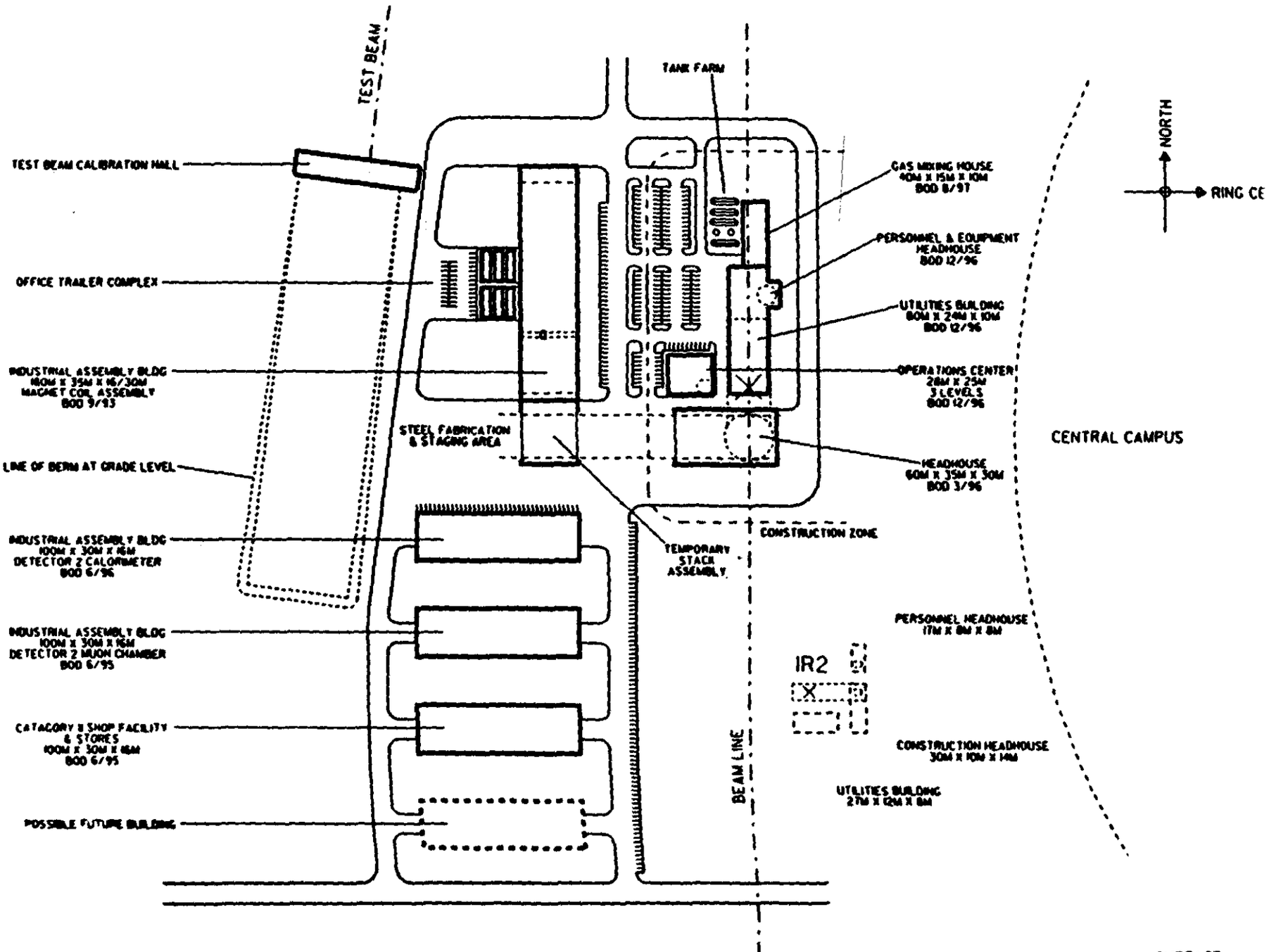


console area

work Area

Expansion Area

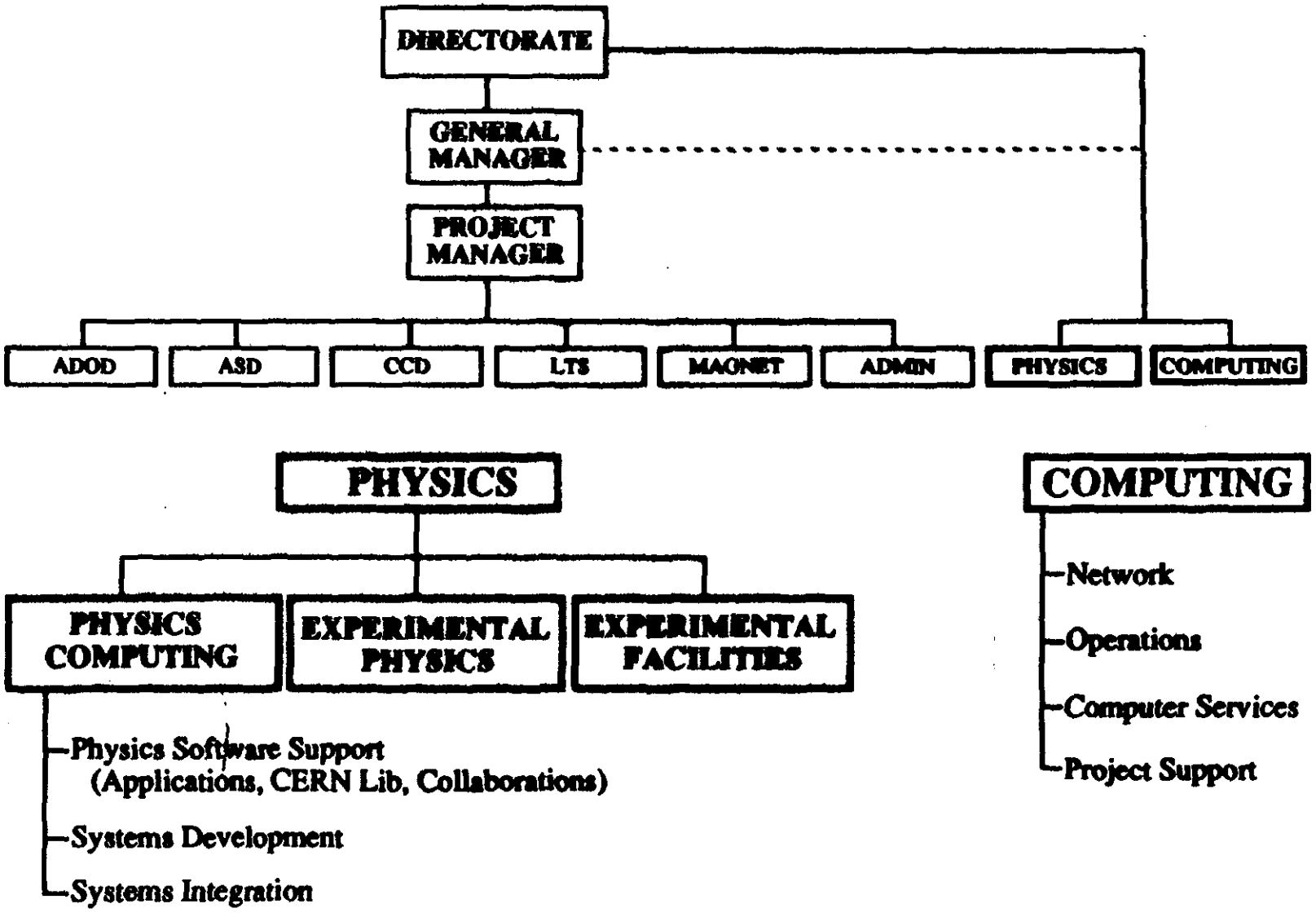
~3000 ft²



SURFACE
DETECT
06/
TVP/KH -

SSCL Computing Organization and Projections

L. R. Cornell



File: dcornell

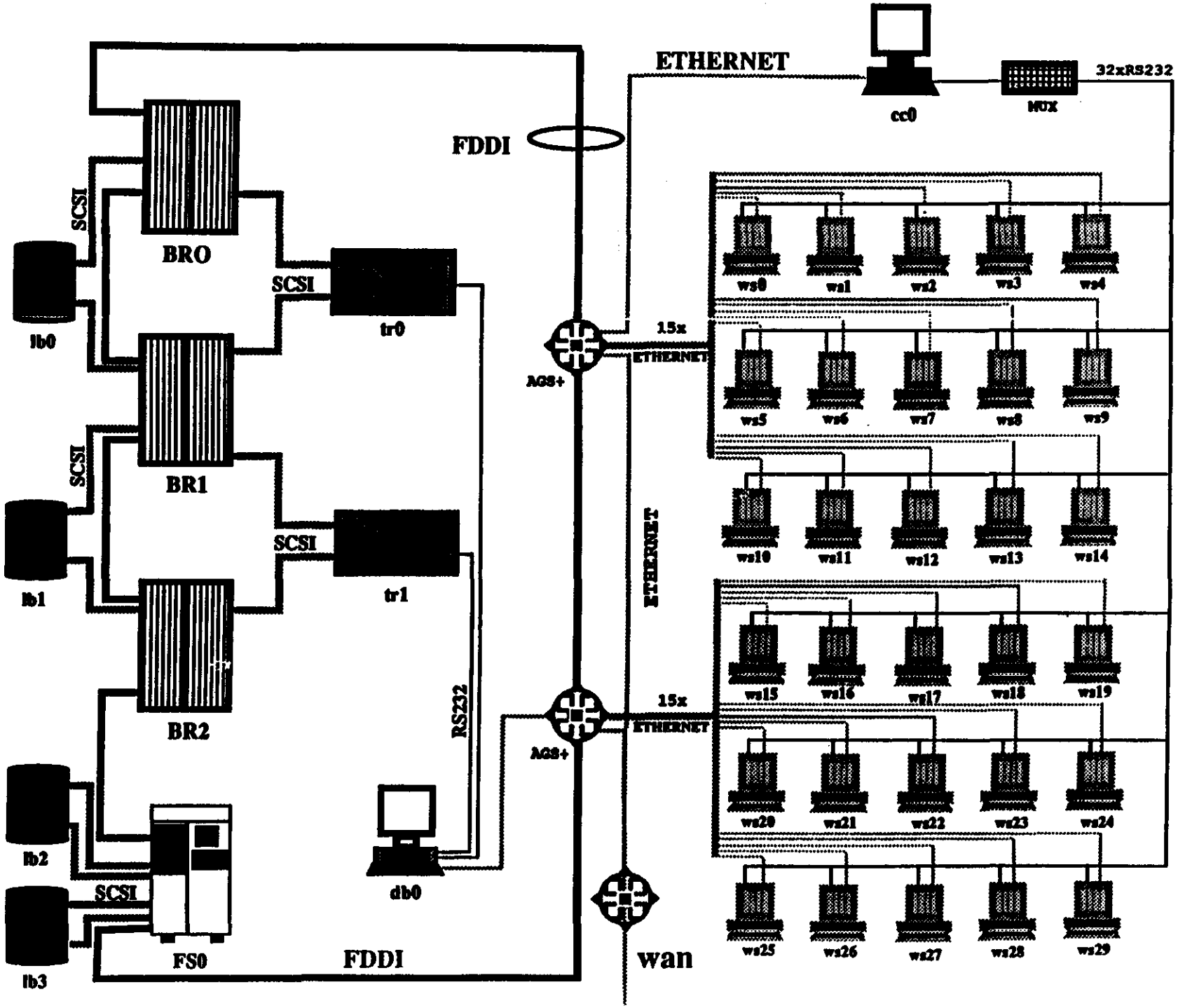
COMPUTING GROUP

WHAT CAN WE DO FOR GEM?

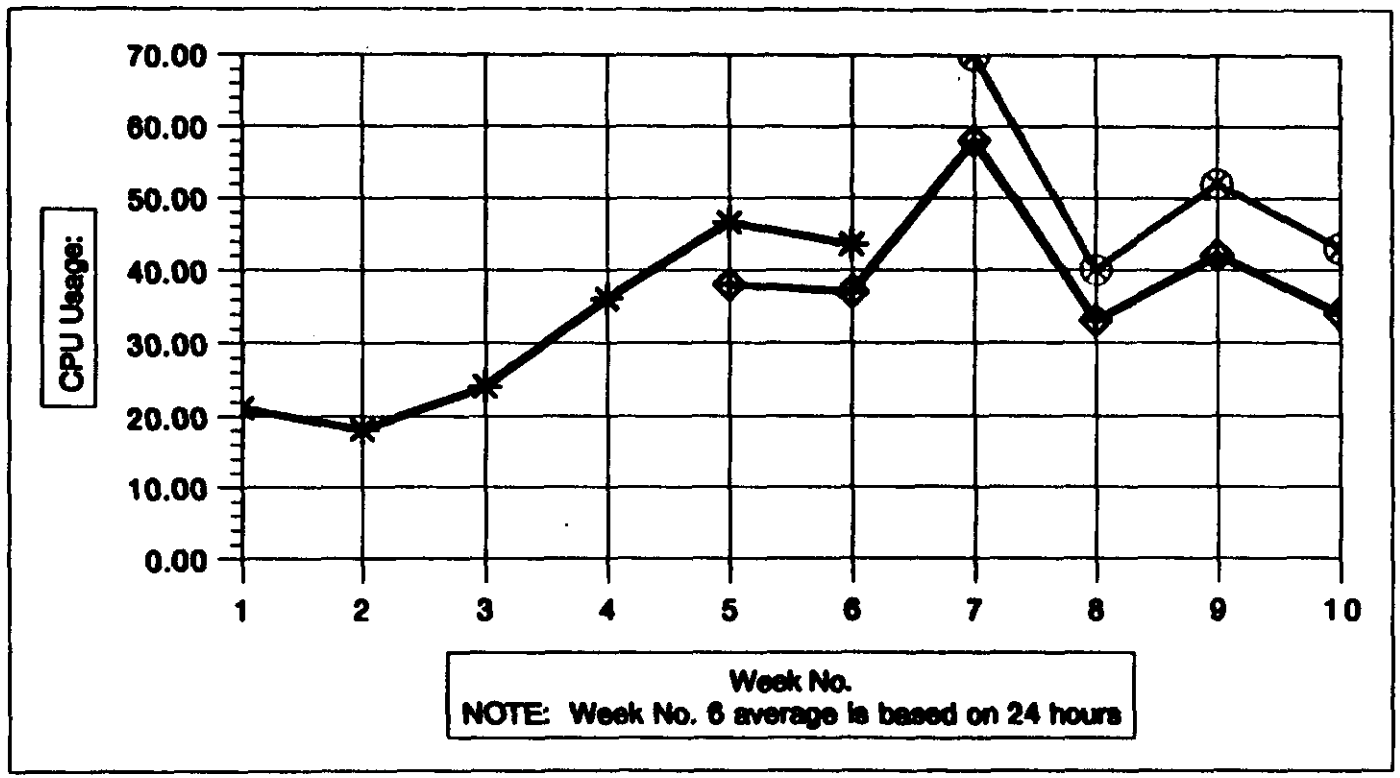
- SOFTWARE SUPPORT: CERNlib, CPS, GEANT, etc., port to various RISC/UNIX platforms, library, documentation support
- SYSTEMS DEVELOPMENT: Improved methods (thruput) for parallel processing. R&D in tape storage, data base techniques.
- MANPOWER RESOURCE: Can provide dedicated support/ development personnel. For example, have hired several physicists/computer scientists this year to support SDC.
- COMPUTING RESOURCES: PDSF - 1000 SSCUPs (~VUPs). Expanding to 4000 SSCUPs by end of year. Will continue to develop systems as required for test beam analysis, on-line and off-line analysis in future.
- USER SERVICES: UNIX manuals, guides, training. PDSF user manual. Will provide computer user area soon.

PHYSICS DETECTOR SIMULATION FACILITY NETWORK

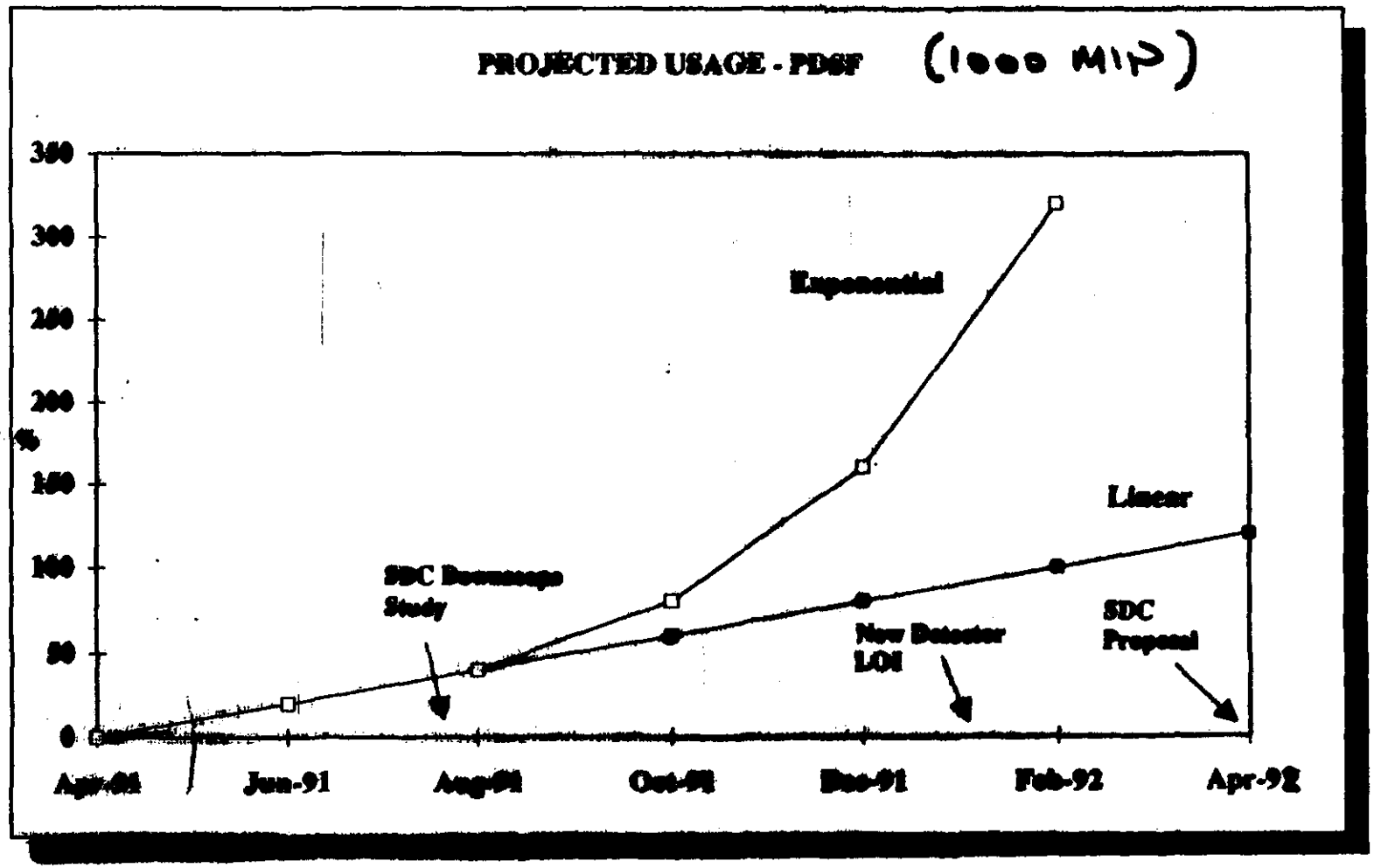
T. SONG 2/14/91

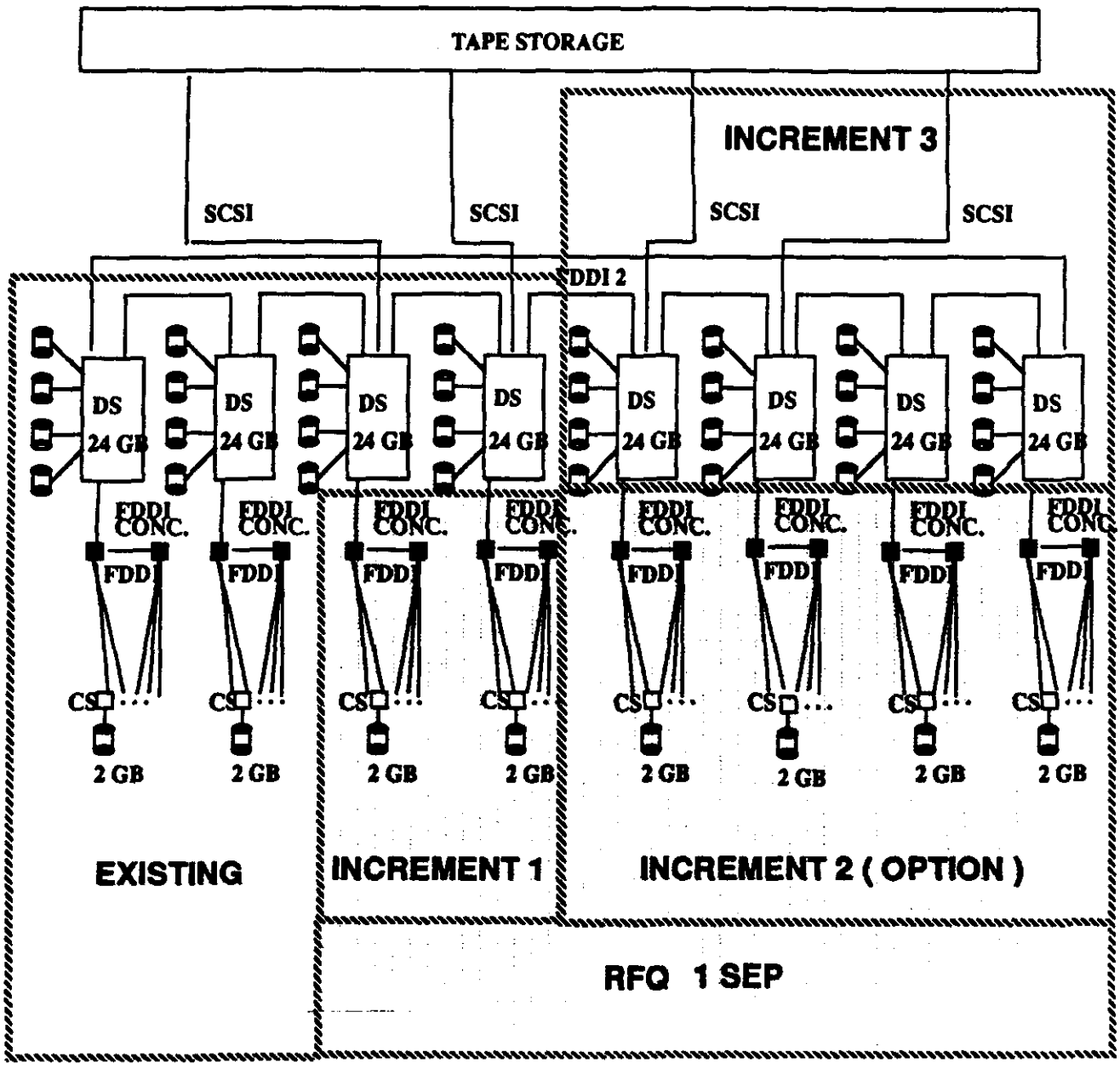


PDSF Average Utilization
06/20-08/28

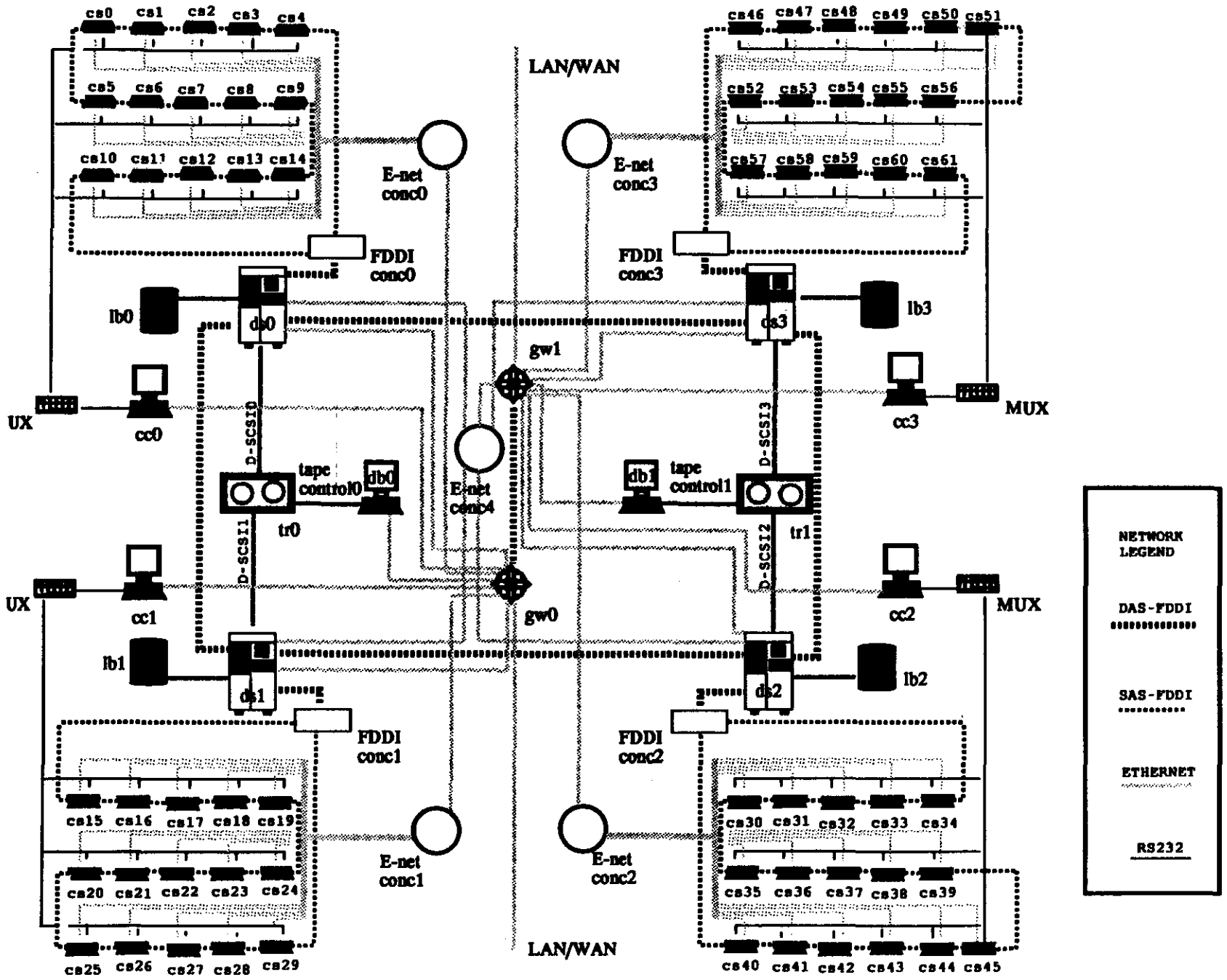


1 - Prime-time statistics (5 days by 10 hours).
2 - Full statistics (7 days by 24 hours).
3 - Full statistics (7 days by 24 hours). Excluding fs0


PROJECTED USAGE - PDSF
(1000 MIPS)
350
300
250
200
%
150
100
50
0
Exponential
Linear
SDC Downscope Study
New Detector LOI
SDC Proposal
Apr-91
Jun-91
Aug-91
Oct-91
Dec-91
Feb-92
Apr-92


TAPE STORAGE
INCREMENT 3
SCSI
SCSI
SCSI
SCSI
DDI 2
DS
24 GB
FDDI
CONC.
FDDI
CS
2 GB
EXISTING
INCREMENT 1
INCREMENT 2 (OPTION)
RFQ 1 SEP

PLANNED PDSF WITHOUT OPTION



OFF-LINE COMPUTING COSTING MODEL

ASSUMPTIONS

- Approximately $75M (FY90) Available
- Assume 60% for Hardware -- 40% for Maintenance, Operations, etc.
- Spread Costs to Ramp up and Peak in FY98
- Computing Model Assumes DCE: Ranches of MP Compute Servers, File Servers, Workstations, and Associated Disks, Network, Tape Robots, Peripherals
- Attempt to Model Decrease in $/MIP and Increase in MIPS/CPU (Make Similar Assumptions for RAM, Disks, and Tapes)

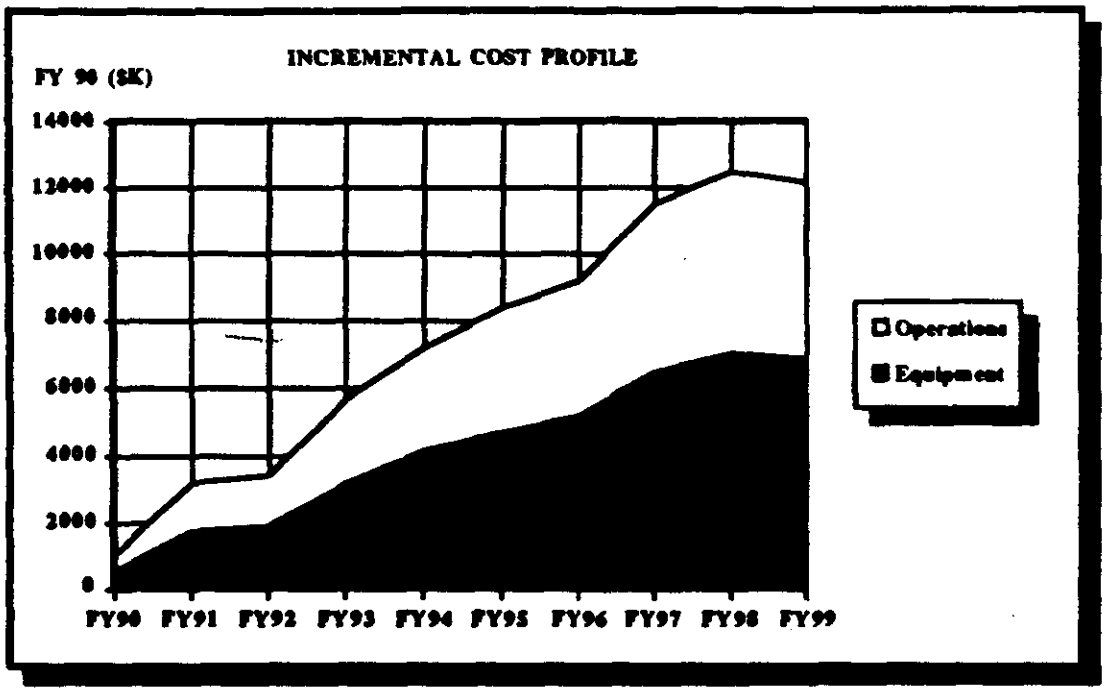


Figure E.2 Off-Line Computing Costs – Incremental.

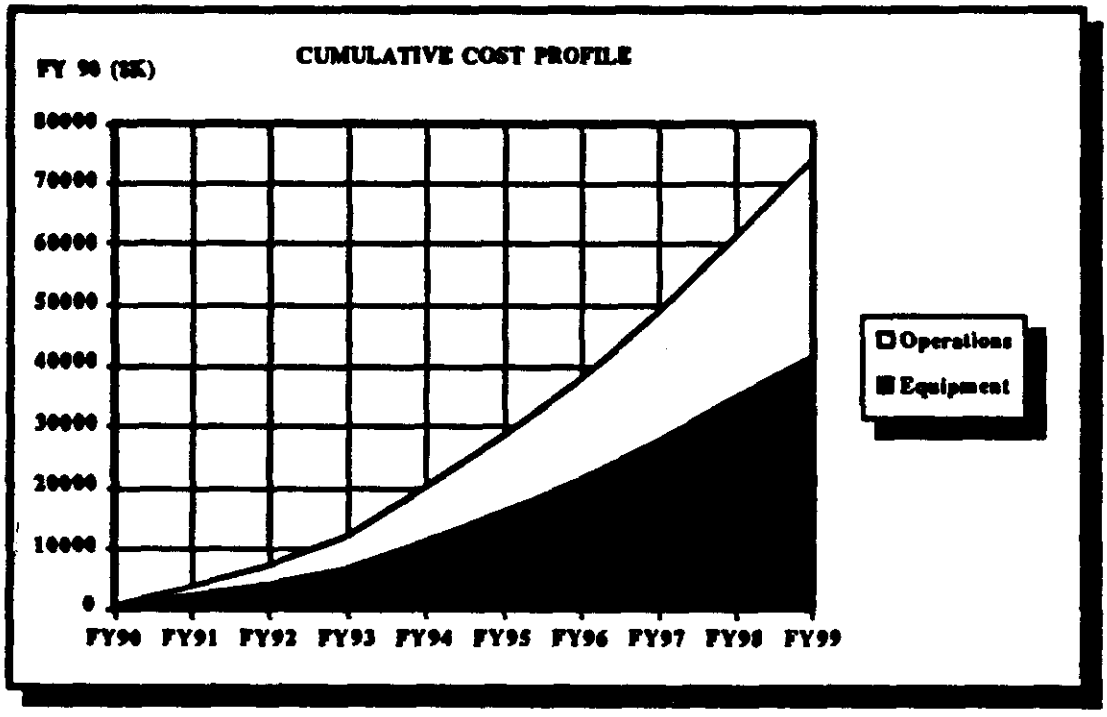
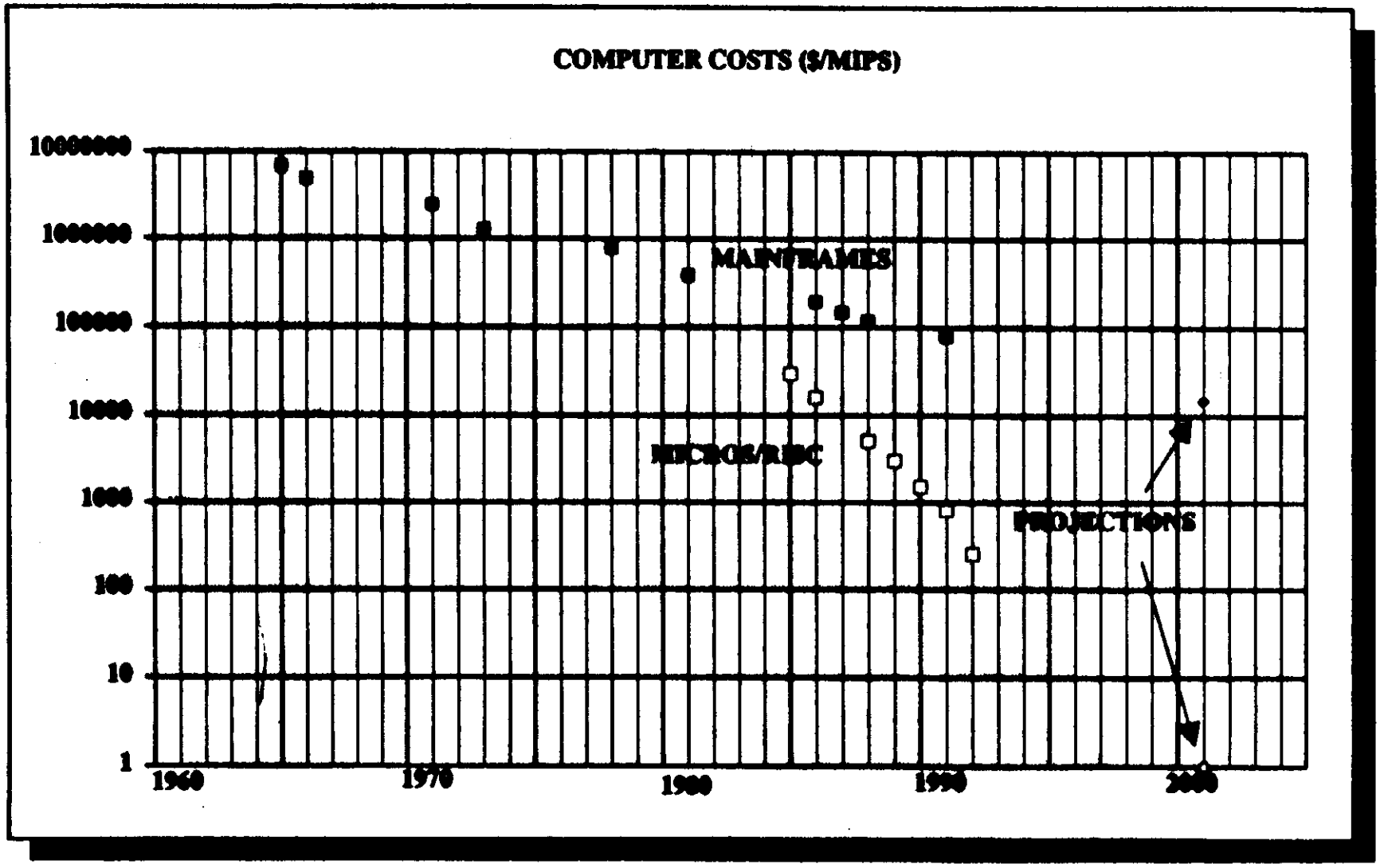


Figure E.3 Off-Line Computing Costs – Cumulative.



7/15/91 1:36 PM -LRC

Table E.3 Total Available Off-Line Computing Resources

DEVICE		FY90	FY91	FY92	FY93	FY94	FY95	FY96	FY97	FY98	FY99
Compute Servers	QTY	0	3	7	14	25	39	52	68	83	97
	MIPS	0	720	2064	5424	12464	24784	43904	76160	125600	198560
File Servers	QTY	1	2	3	6	9	13	17	23	28	32
	MIPS	80	200	424	984	2157	4131	7317	12693	20933	33093
Workstations	QTY	10	34	66	122	210	312	416	544	664	776
	MIPS	200	920	2264	5624	12664	24784	43904	76160	125600	198560
Magnetic Disks	GB	16	65	144	289	559	905	1351	1917	2601	3314
Optical Disks	GB	0	0	0	70	290	570	1050	1650	2240	3020
Tape Storage	TB	0	2	5	12	26	50	80	130	197	282

UNOFFICIAL

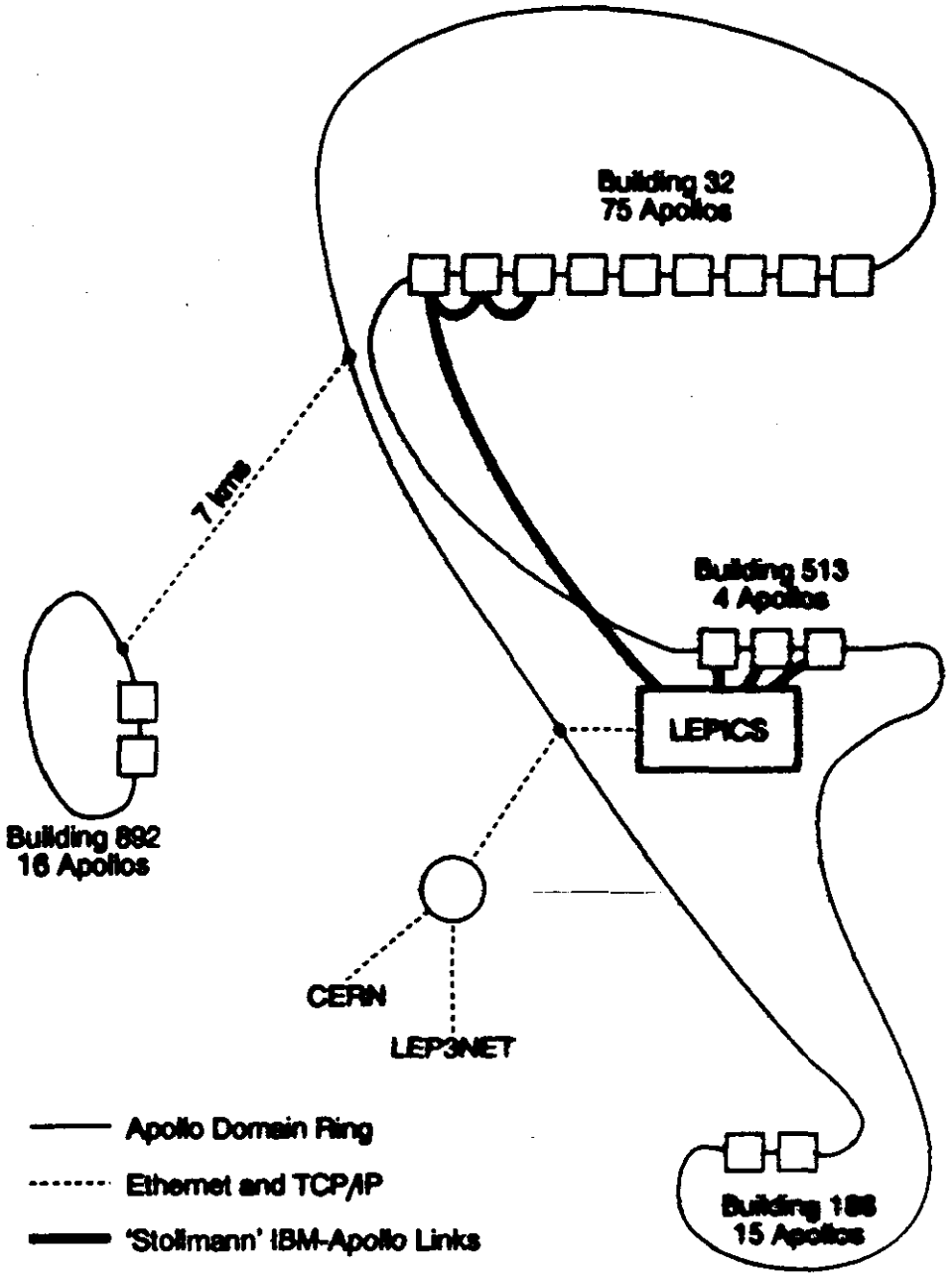
L3 COMPUTING

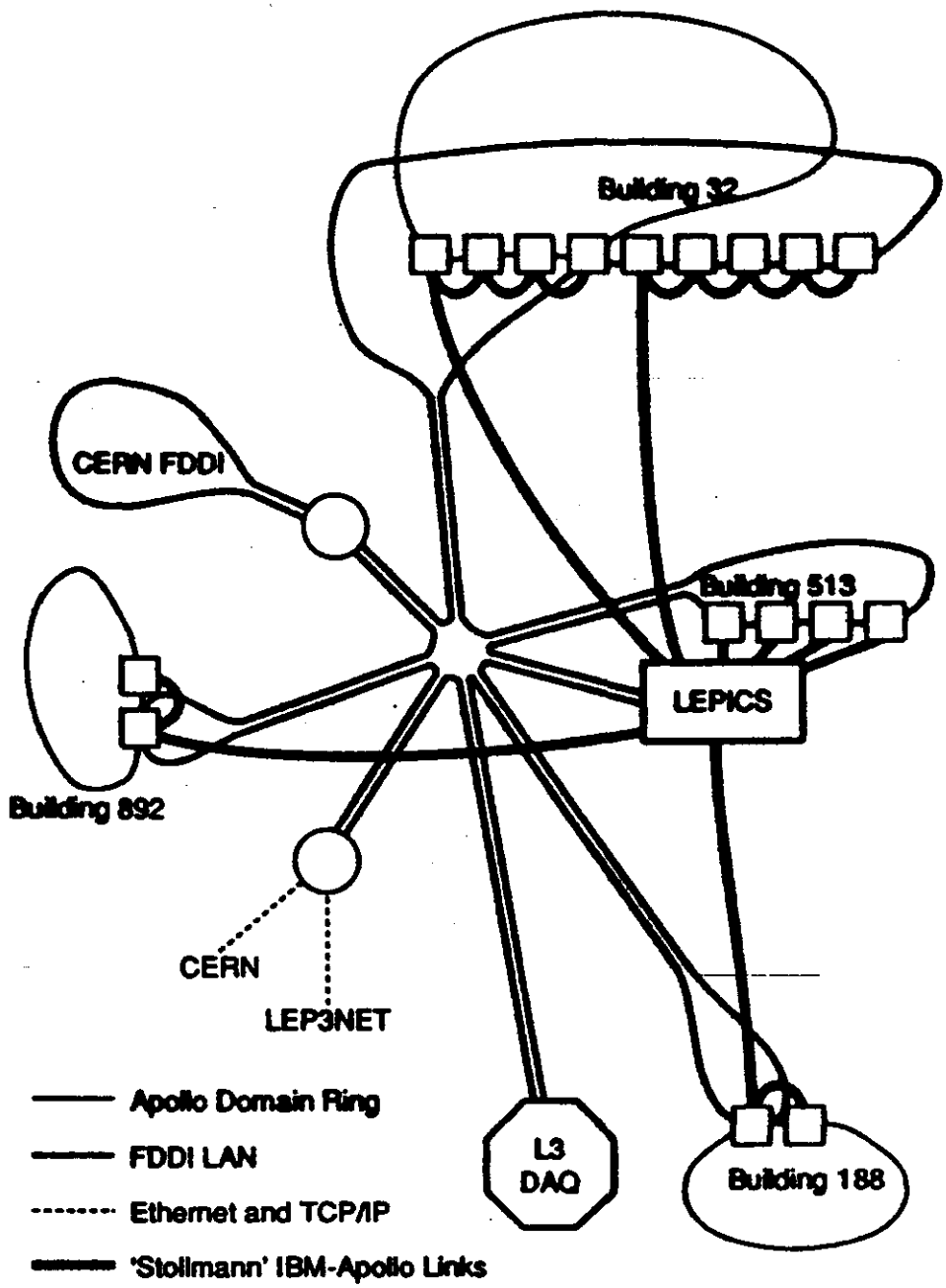
GEM COMPUTING MEETING

SEPTEMBER 4, 1991

RICHARD P. MOUNT

CALTECH


Building 32
75 Apollos
7 kms
Building 513
4 Apollos
LEPICS
Building 892
16 Apollos
CERN
LEP3NET
Building 188
15 Apollos
Apollo Domain Ring
Ethernet and TCP/IP
'Stollmann' IBM-Apollo Links



//l3cit3/localuser/mount/citann91/figs /computing_cern_new_fig

L3 Apollo Network at CERN

- 100 nodes:

  22 DN10000s (36 CPUs).

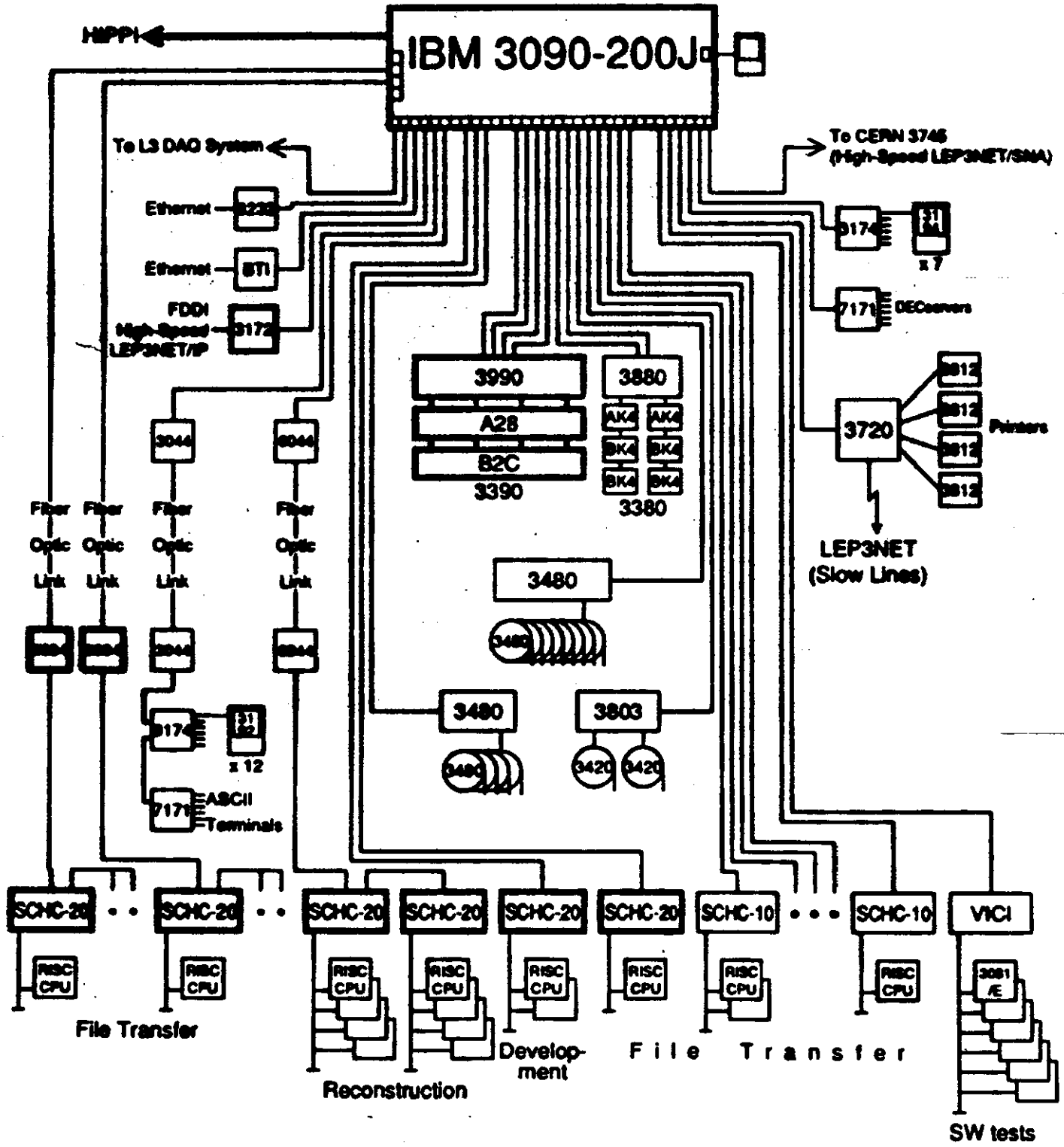
- Over 60 GB disk.

- Optimum topology: one ring.

  (Each user needs access to data stored on all other nodes in the ring.)

- Massive (factor > 100) insufficiency of Domain ring for file access.

- FDDI also expected to be insufficient.


IBM 3090-200J
HIPPI
To L3 DAQ System
Ethernet
Ethernet
FDDI
High-Speed
LEP3NET/IP
To CERN 3745
(High-Speed LEP3NET/SNA)
x 7
DECservers
3990
3880
A28
B2C
3390
3380
3720
Printers
LEP3NET
(Slow Lines)
3480
3803
Fiber Optic Link
x 12
ASCII
Terminals
SCHC-20
SCHC-10
VICI
RISC CPU
File Transfer
Reconstruction
Develop-
ment
File Transfer
SW tests

Statistics Display for Node 2C08E //13dsp10k1

Process Name	
Null Process	0.0%
Null Process	0.0%
Null Process	0.0%
Null Process	0.0%
OS processes	0.1%
init	0.0%
(rpcsvcgs)	94.1%
(rpcsvcgs)	95.7%
(rpcsvcgs)	95.0%
(rpcsvcgs)	95.7%
(qlibvlib.exe)	1.9%
(sh)	0.0%
(sh)	0.0%
(ngsdaemon)	0.0%
(qlibvlib.exe)	0.2%
(sh)	0.0%
(sh)	0.0%
(ngsdaemon)	0.0%
(qlibvlib.exe)	0.0%
(sh)	0.0%
(sh)	0.0%
(ngsdaemon)	0.0%
(pd daemon)	0.0%
server proxyd.n	0.0%
(stoll FxFer)	0.0%
(ngtdaemon)	0.0%
(ngsdaemon)	0.0%
(lpsdaemon)	0.0%
nbu helper	0.0%
(qwlisd)	0.0%
nsimon	0.0%
(qcon)	0.0%
(lbd	0.0%
nsmsd	0.0%
(rwmlmsd)	0.0%
(lmsfd)	0.0%
(lsad)	0.2%

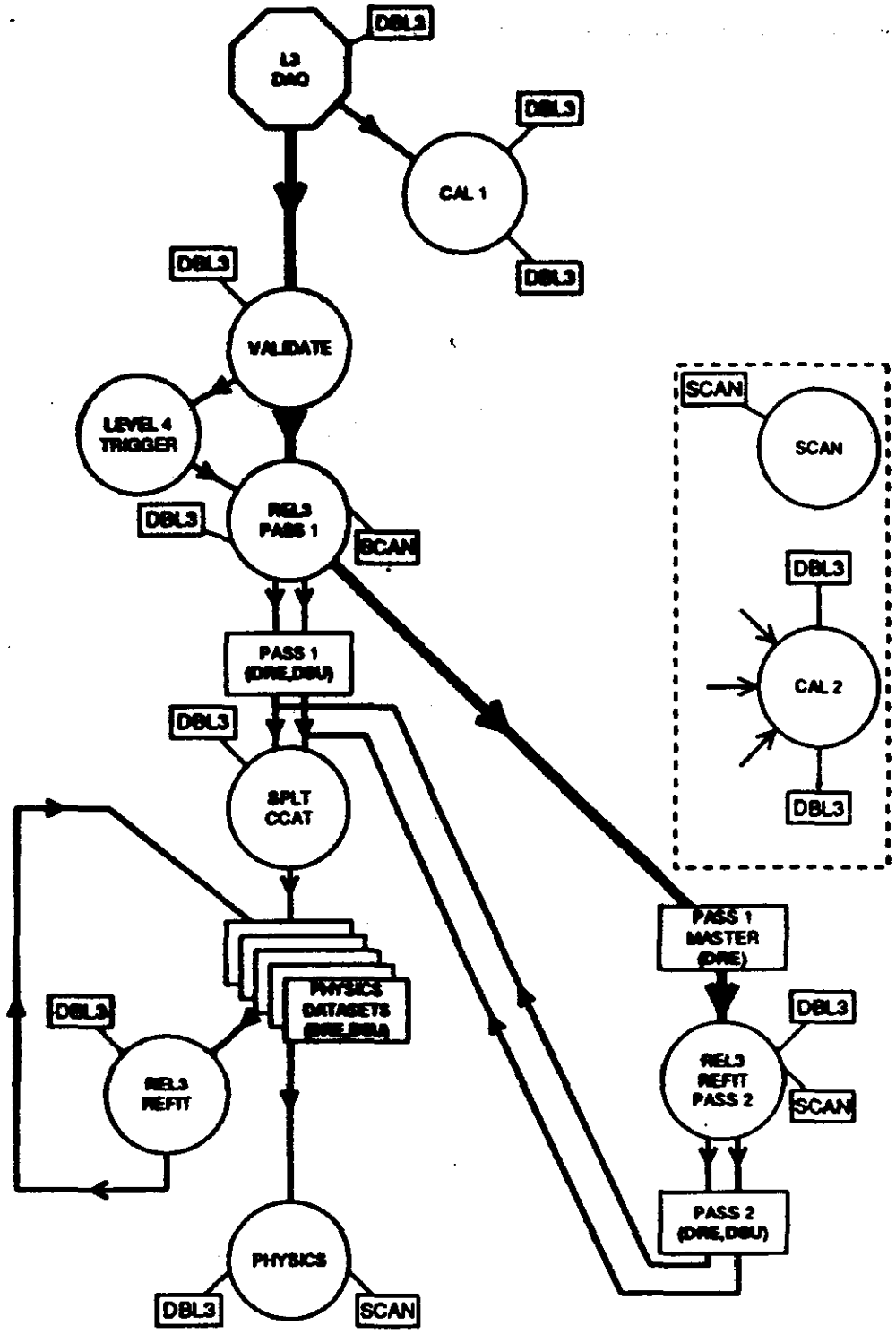
IO Type	
Network	94
Winchester	0

Network IO Type	
Paging Requests	0
Paging Service	0
Other Traffic	94

Command:

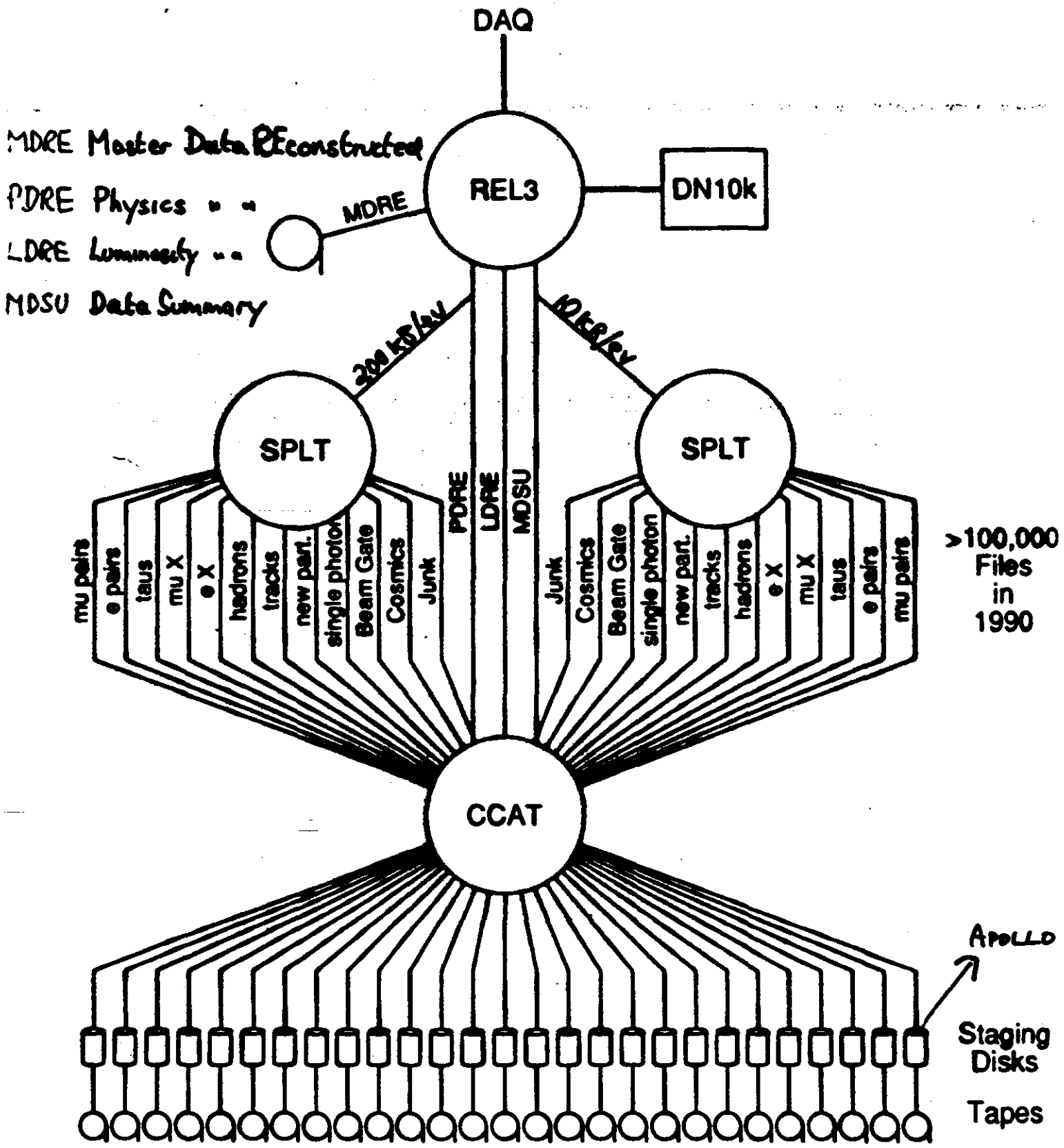
Summary of Requirements of the L3 Data Analysis Task

- An integrated local and wide area computing system.
- Everybody should have good access to 40 Terabytes of data.
- 500 to 1000 MIPS are needed, much of it with high-speed access to the data.
- The user interface (especially graphics) should be the best offered by today's workstations.



//13cit3/localuser/mount/citann91/figs /analysis_analdpsnew_fig

Data Handling



R. P. Mount, August 1990

Joint Projects

L3/IBM	Parallel Processing
L3/Apollo	DN10000 to IBM Channel Interface
L3/IBM	HPPI: to workstations and long-distance
L3/HP	FDDI: high throughput

L3 Off-line Computing at CERN
Investment and Running Costs

INVESTMENT: Hardware

Workstations	$ 3.2 M
Mainframe (CPU/Memory/Channels)	$ 1.4 M
Mainframe Peripherals	$ 1.2 M
Miscellaneous	$ 0.2 M

INVESTMENT: Manpower

Workstation-Mainframe Communications	3-4 man years
Parallel Processing	3-4 man years
Data Management	2-3 man years

L3 SOFTWARE

CODE MANAGEMENT:	PATCHY (DBL3)
	CMZ
SIMULATION:	EGL3
	SIL3 (GEANT + GEISHA)
RECONSTRUCTION, SCANNING, REFITTING, PHYSICS ANALYSIS	REL3 (GEANT + ...)
PHYSICS ANALYSIS:	PAW
CALIBRATION DATABASE:	DBL3
PRODUCTION LOG:	DBL3
DATA MANAGEMENT	SPLT/CCAT/FATMEN/L3STAGE...

L3 DATA STRUCTURES

(ZEBRA)

- FLEXIBLE, LARGELY SELF-DESCRIBING, STRUCTURE (ENCODED LINKS)

- ACCESS ALWAYS THROUGH 'MNEMONICS'.

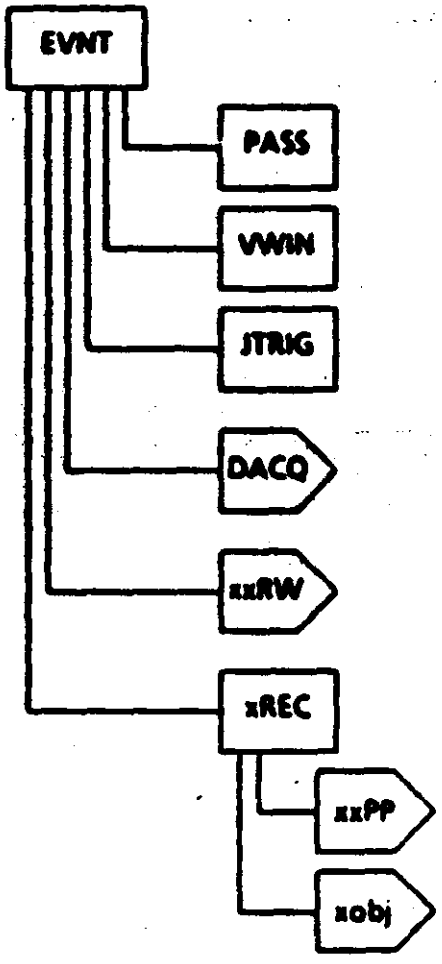
- RECONSTRUCTION ≡ DETERMINE WHICH OBJECTS ARE CONSTITUENTS OF HIGHER LEVEL OBJECTS.

  CREATE REFERENCE LINKS TO SHOW RELATIONSHIP

- DST FORMAT IS INVISIBLE.

  (REL3-type DATA STRUCTURES ARE RECREATED WHEN READING A DST).

REL3 Data Structures: Top Level Structure

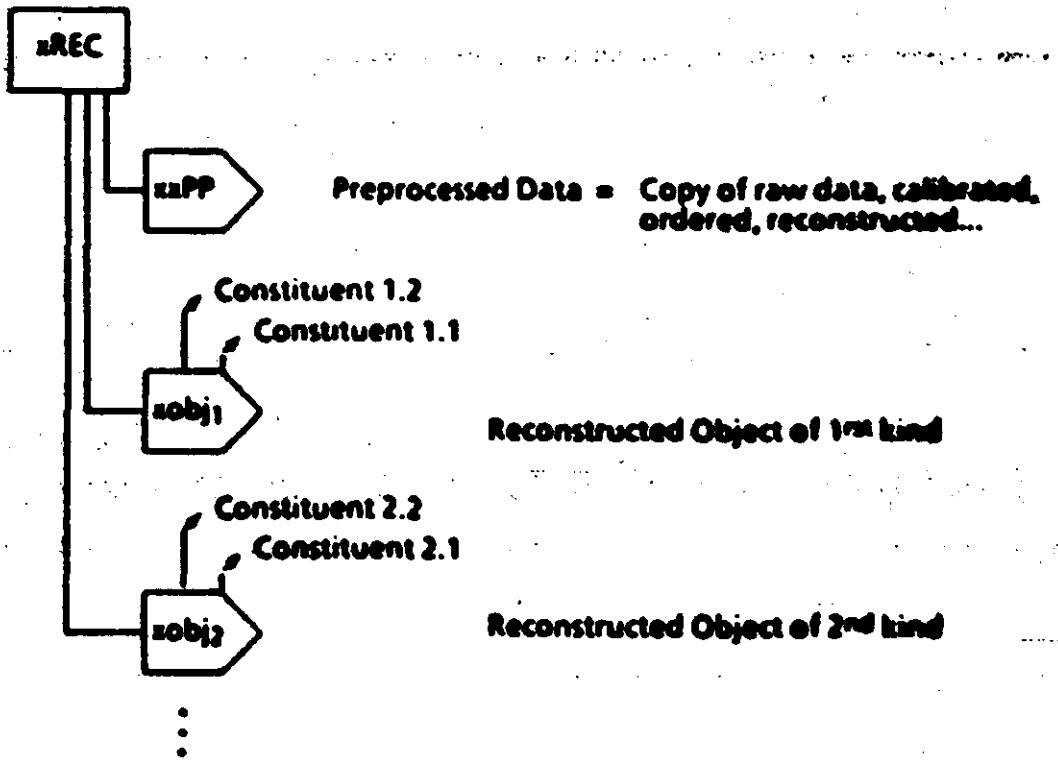


x = T
E
H
M
F
S

xx = TE/TZ
EC
HB/HC/HM
MP/MZ
FB/FW
SC

xobj = TTRK/TVRT
EGCL/EBMP
HGCL
MUTK
...

REL3 Data Structures: Reconstruction Fan-Out xREC



```
C REL3  V  1.62/01   910723 18:00   GEANT3/L3 Reconstruction programs
+PATCH, REL3HEAD, IF=$DOC.
+DECK, REL3HEAD.

Current version number : 1.60

Coordinators for this Pam : J.Branson, F.Bruyant
Deputy                    : S.Banerjee, N.Colino


This is a 'Steering' Pam file for the preparation of the following
programs, through the corresponding Pilot Patches *[program name],

  for library :

  RELI  creation of L3 Reconstruction library, with all options selected

  for tests and production :

  REGEL3  event Reconstruction and display in the L3 detector; in the
          production version used for creating GEOM and SETS structures
          for Reconstruction

  for production :

  VLDT    event Validation pass
  RECO    event Reconstruction
  RFIT    event Refit

  for tests :

  RNGEL3  same as REGEL3 without code for Initialization structures
  RDGEL3  same as RNGEL3 without code for Reconstruction (display only)


Extra material required :

  Pam files DBL3, INL3, UTL3, VWL3, XSL3, ECL3, FLL3, HCL3, JTL3, MUL3,
            SCL3, TEL3, AXL3, AAL3 and ABL3;
            + GEANT341 (L3 version of GEANT 3.14);

  Latest bug correction pam file L3CORR (later -> //DBL3/CORPAM )

  CERN libraries KERNLIB and PACKLIB
    (including the latest versions for ZEBRA, FFREAD, HBOOK, KUIP, PAW,
     MIGZ, as well as the graphics packages)

  For Host/Attached Processor operation:
    Host : APUTYLIB + L3APOLIB (Apollos) or L3EMULIB (Emulators)
           N.B.: APUTYLIB contains ZEBRA 3.65, FFREAD 3.08 and
                 HARRAY+HTITLE from HBOOK 4.06
    Child: Apollos: apoinit.bin, dbchv304.lib, apogeant.lib,apopack.lib
           Emulators: DBCHV307,L3EGEANT,L3EPACKL,L3EFORTL


List of Patches :
----------------

  REL3HEAD Pam header and short history of successive versions
  REL3DOC  General documentation
  REL3BNKS Description of the banks
  *RELI    Pilot patch for creating L3 Reconstruction library
  *RANAL   User Pilot for a cut down analysis work
  *RDGEL3  As *RNGEL3, but code for the event reconstruction is removed
  *VLDT    Production Pilot for Validation pass
  *RECO    Production Pilot for Pass 1 reconstruction
  *RFIT    Production Pilot for Refit pass
  *RNGEL3  As *REGEL3, but code for the initialization phase is removed
  *REGEL3  Pilot patch for event reconstruction and display
  RECDES   Patchy sequences and CDE'S
  REGEL3   High level steering routines for event Reconstruction
  REBODY   Fanout and auxiliary routines for event Reconstruction
  RESETO   Material for graphic interface preparation
```

RESIMU Interface routines for reconstruction in simulation context
RESPIN Special input format
REUTIL Utility routines for reconstruction
RTRDOC Documentattion for fast junk rejection algorithm
RTRIGR Fast junk rejection code
REXINT Action routines for interactive commands
REUSER User entry points in reconstruction
RXL3BNKS Documnetation for DSU banks
RXCDSU Code for converting to DSU format
RXRDSU Code for converting from DSU format
RXUTIL Utility routines for packing and unpacking
REP3   Routines specific to P3 operation mode

PATCHY Options :

$DOC - To flag the (non-executable) documentation Patches

APOLLO, IBM or VAX for the computer in use
$HIGZ for the interface with the graphics package
$INIT to activate the initialization code

$OUTLIB to ignore routines at library creation time
$REGEL3 to select the Reconstruction menus for interactive usage
$VALID to select code specific for event Validation
$PASS1 to select code specific for event Reconstruction
$REFIT to select code specific for event Refit
$SHOW to select code specific for job with no reconstruction
but with the possibility of preparing & displaying the
graphics objects
$SCAN to select additional code needed for Scan program
$VALHIS to select code specific for validation histograms
(default with $VALID)
$RDSU to enable decoding of DSU tapes
$WDSU to enable encoding of DSU format and write on DSU file
$ANAL to antiselect detector analysis, event selection, .. codes
$PPXP to allow expansion of PP banks, even when detector
reconstruction code is turned off

$BATCH - Mandatory for batch jobs
$DEBUG - Control of debug print-out (+ DEBUGG in GEANTDEV)
$SIMU - for reconstruction and reading of simulated data

$AXL3 - to trigger 'A cross' L3 detector combined reconstruction
+ particle identification
(default in *REGEL3, *RNGEL3 and *RDGEL3)
$AAL3 - to trigger Group Beta event selection + analysis
(default in *REGEL3, *RNGEL3 and *RDGEL3)
$ABL3 - to trigger Group B event selection + analysis
(default in *REGEL3, *RNGEL3 and *RDGEL3)
$ECAL - to trigger Electromagnetic CALorimeter reconstruction
(default in *REGEL3, *RNGEL3 and *RDGEL3)
$FLUM - to trigger Forward LUMinosity monitor reconstructtion
(default in *REGEL3, *RNGEL3 and *RDGEL3)
$HCAL - to trigger Hadron CALorimeter reconstruction
(default in *REGEL3, *RNGEL3 and *RDGEL3)
$JTRG - to trigger TRiGger data reconstruction
(default in *REGEL3, *RNGEL3 and *RDGEL3)
$MUCH - to trigger MUon CHambers reconstruction
(default in *REGEL3, *RNGEL3 and *RDGEL3)
$SCNT - to trigger SCiNTillation counter reconstruction
(default in *REGEL3, *RNGEL3 and *RDGEL3)
$TECH - to trigger Time Expansion CHamber reconstruction
(default in *REGEL3, *RNGEL3 and *RDGEL3)
$VWIN - for graphics preparation in relation with VWL3 pam file

$P3HOST - for the prepariation of the host part of the P3 version
$P3CHILD - for the prepariation of the child part of the P3 version
$AP3081E - to select the Emulators as attached processors
$APDN10K - to select the Apollos as attached processors

+DECK, RESTORY.

```
| SELECT|Redefine the SELECT data |  IRUN   | Run number                |
|       |card                     |  IEVB   | First event number        |
|       |                         |  IRNG   | Range of events           |
+-------+-------------------------+---------+---------------------------+
```

+DECK, REL3FLOW.

Reconstruction Program simplified flow chart

(*) means standard CERN code (not L3 code)

```
M A I N        GEANT3-L3  Main Program
 |
 |
 |->G Z E B R A (*)  initialization of Zebra system
 |
 |
 |->U G I N I T      user initialization routine for GEANT3-L3
 |   |               reconstruction
 |   |
 |   |->GINIT  (*)       GEANT initialization routine
 |   |->UTLUDF           assignment of standard logical units
 |   |->REGEDF           assignment of standard tracking cuts
 |   |->REGZIN           initialization of GEANT/ZEBRA data structures
 |   |->UTZINI           initialization of L3/ZEBRA data structures
 |   |->UZINIT           initialization of the UZ package
 |   |->REUZIN           initialization of UZ counters
 |   |
 |   |->REDATA           defines Reconstruction specific user data cards
 |   |   |
 |   |   |->TEDATA           TEC chamber
 |   |   |->ECDATA           Electromagnetic Calorimeter,
 |   |   |->......           ........,
 |   |   |->......           ........,
 |   |   |->JTRDAT           Trigger
 |   |   |->AXDATA           Across-L3
 |   |   |->REDATU           User specific
 |   |
 |   |->UTDATA           reads GEANT standard and user data cards
 |   |   |
 |   |   |->GFFGO  (*)       reads GEANT standard data cards
 |   |   |->UTRUID           decodes RUID data card
 |   |   |->UTIOPA           decodes IOPA data card
 |   |   |->UTSELE           decodes SELE data card
 |   |
 |   |->REDRZF———        decodes DRZF data card
 |   |
 |   |->REVERS           identifies current REL3 Pam file
 |   |->GBHSTA (*)       books GEANT standard histograms
 |   |                   (data card HSTA)
 |   |->REZINI           initialization of reconstruction specific
 |   |   |                ZEBRA data structure features
 |   |   |->TEZINI           TEC chamber
 |   |   |->ECZINI           Electromagnetic Calorimeter,
 |   |   |->......           ........,
 |   |   |->......           ........,
 |   |   |->JTRZIN           Trigger
 |   |   |->AXZINI           Across-L3
 |   |
 |   |->IGINIT (*)       initialization of Graphics package
 |   |->GDINIT (*)       initialization of GEANT3 drawing package
 |   |->LAINIT           initialization of Scan program
 |   |
 |   |->UTDBHD           initializes data base
 |   |->REDBHD           Reconstruction specific data base initial.
 |   |   |
 |   |   |->TEDBHD           TEC chamber
 |   |   |->ECDBHD           Electromagnetic Calorimeter,
 |   |   |->......           ........,
 |   |   |->......           ........,
 |   |   |->JTRDBH           Trigger
 |   |   |->AXDBHD           Across-L3
```

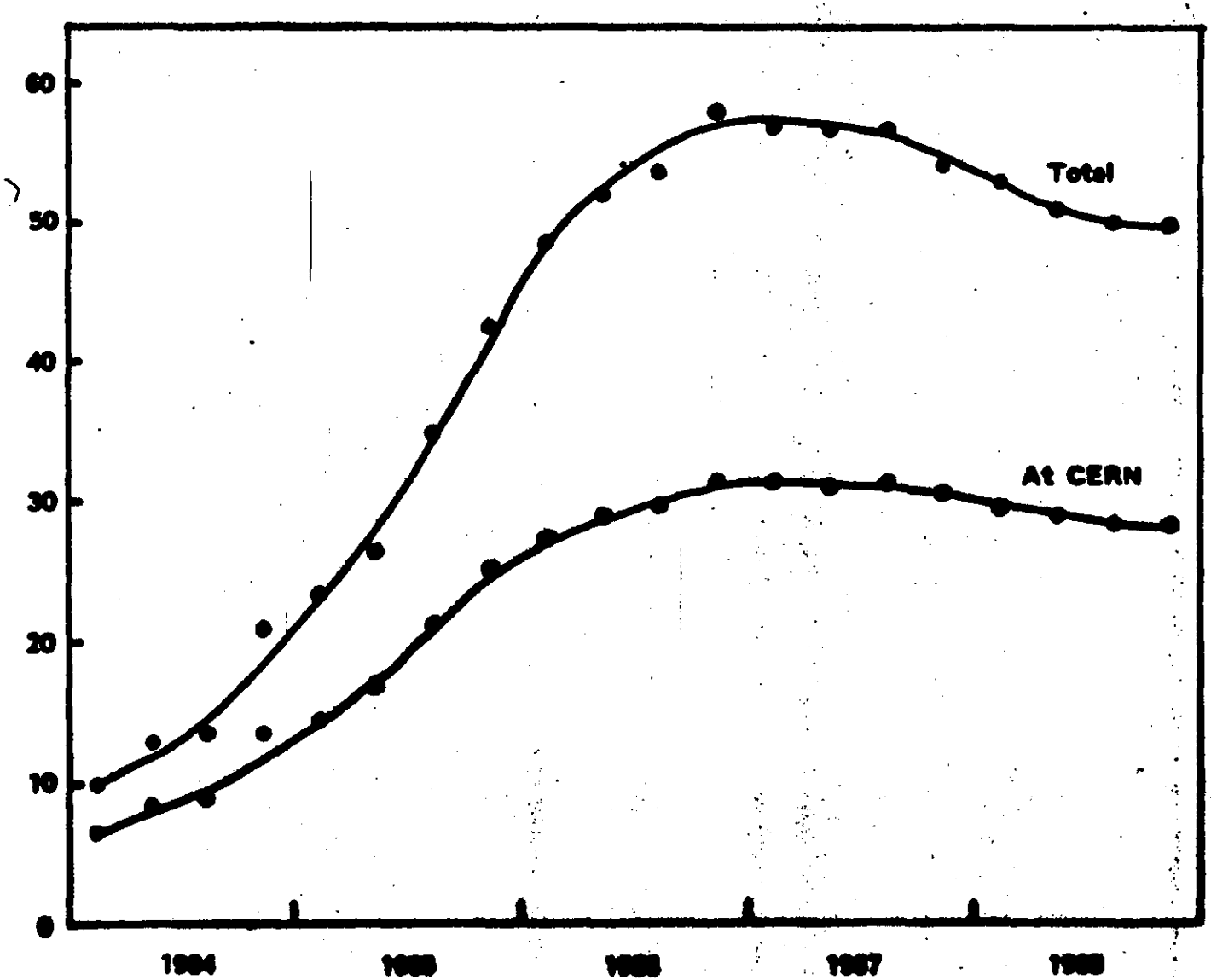
```
|    |                                                                  |
|    |->checks time left                                                |
|    |                                                                  |
|    |------------------------------------------------------------------
|    |
|
|
|->U G L A S T      termination routine
|    |->ZEND            termination control
|    |    |->UTEND (1)
|    |    |    |->VWLAST            terminates VWL3 package
|    |    |    |->IGEND  (*)        Closes HIGZ file
|    |    |
|    |    |->RELAST            terminates reconstruction software
|    |    |    |
|    |    |    |->TEEND             for TEC chamber,
|    |    |    |->.....             .......,
|    |    |    |->.....             .......,
|    |    |    |->SCEND             Scintillator
|    |    |    |->JTREND            Trigger
|    |    |    |->AXEND             Across-L3 reconstruction
|    |    |    |->URZEND         Closes the physics data sets
|    |    |
|    |    |->RUEND             user termination
|    |    |->UTEND (2)
|    |    |    |->GLAST  (*)        GEANT standard termination
|    |    |    |->UTGCLO            Closes FZ files
|    |    |    |->UZEND             terminates UZ package
|    |    |    |->DBEND             closes data base files
|    |    |    |->HPDIR  (*)        prints histograms
|    |
|
stop
```

```
+PATCH, REL3BNKS, IF=$DOC.
+DECK, REL3BNKS.
```

```
+---------+       +------+
|  EVNT   |-(-)--| DACQ |    Special support for all simulation data
+---------+-(-)+ +------+
 |   |||||      |  ||||||   +------+
 |   |||||      |  |||||+---| HEAD |   JHEAD  structure of GEANT3
 |   |||||      |  |||||    +------+
 |   |||||      |  ||||+----| VERT |   JVERTX structure of GEANT3
 |   |||||      |  ||||     +------+
 |   |||||      |  |||+-----| KINE |   JKINE  structure of GEANT3
 |   |||||      |  |||------+------+
 |   |||||      |  ||+------| HITS |   JHITS  structure of GEANT3
 |   |||||      |  ||       +------+
 |   |||||      |  |+-------| JXYZ |   JXYZ   structure of GEANT3
 |   |||||      |  |        +------+
 |   |||||      |  +--------| DIGI |   JDIGI  structure of GEANT3
 |   |||||      |           +------+
 |   |||||      | +------\
 |   |||||      +-| DACQ  >   Special support for real data from DAQ
 |   |||||        +------/    (linear structure EVDQ,MUCH,...,JTRG)
 |...|||||      +------\
 |   ||||+-----| PASS  >   Information of a given pass of program
 |   ||||      +------/
 |   ||||           |   All xOBJ's
 |   ||||           |   //....//
 |   ||||           |   +------+
 |   ||||           +---| LOBR |    List of reconstructed objects
 |   ||||               +------+
 |   ||||      +------\
 |   |||+------| xyRW  >   Raw data banks for all detectors
 |   |||       +------/
 |   |||            |     +------+
 |   |||            +-(-)-| xyDI |   Track Information for MC data
 |   |||                  +------+
 |   |||
 |   |||       +------+
```

L3 OFFLINE SOFTWARE BASE
MANPOWER NEEDS

(1984 ESTIMATE)



L3 SOFTWARE

(PERSONAL) VERDICT

- L3 ACHIEVED WELL ENGINEERED SOFTWARE WITHOUT CASE TOOLS OR EVEN BUBBLES.

  (WHICH DOES NOT MEAN THAT CASE TOOLS ETC. ARE A BAD IDEA!)

- ZEBRA, CMZ/PATCHY, GEANT, DBL3 — SERVED L3 VERY WELL

  ∴ ARE SERIOUS CANDIDATES FOR EVOLUTION INTO SOFTWARE FOR SSC EXPERIMENTS.

- INTERACTIVE GRAPHICS (INSTRUMENTATION OF THE CODE) IS A KEY TO CONFIDENCE IN PHYSICS RESULTS

- A STRONG, CONSENSUS SEEKING, MODERATELY DEMOCRATIC, SOFTWARE MANAGEMENT IS VITAL.

DØ

R. Raja (FNAL)

An Overview

Rajendran Raja 3 Sept 91

- Structured Analysis and System design (SASD

from real

- Data Acquisition System (rates, Level II triggering, Tapes)

from real C

- Alarms and Monitoring (Hardware database, Alarms, Parameter page)

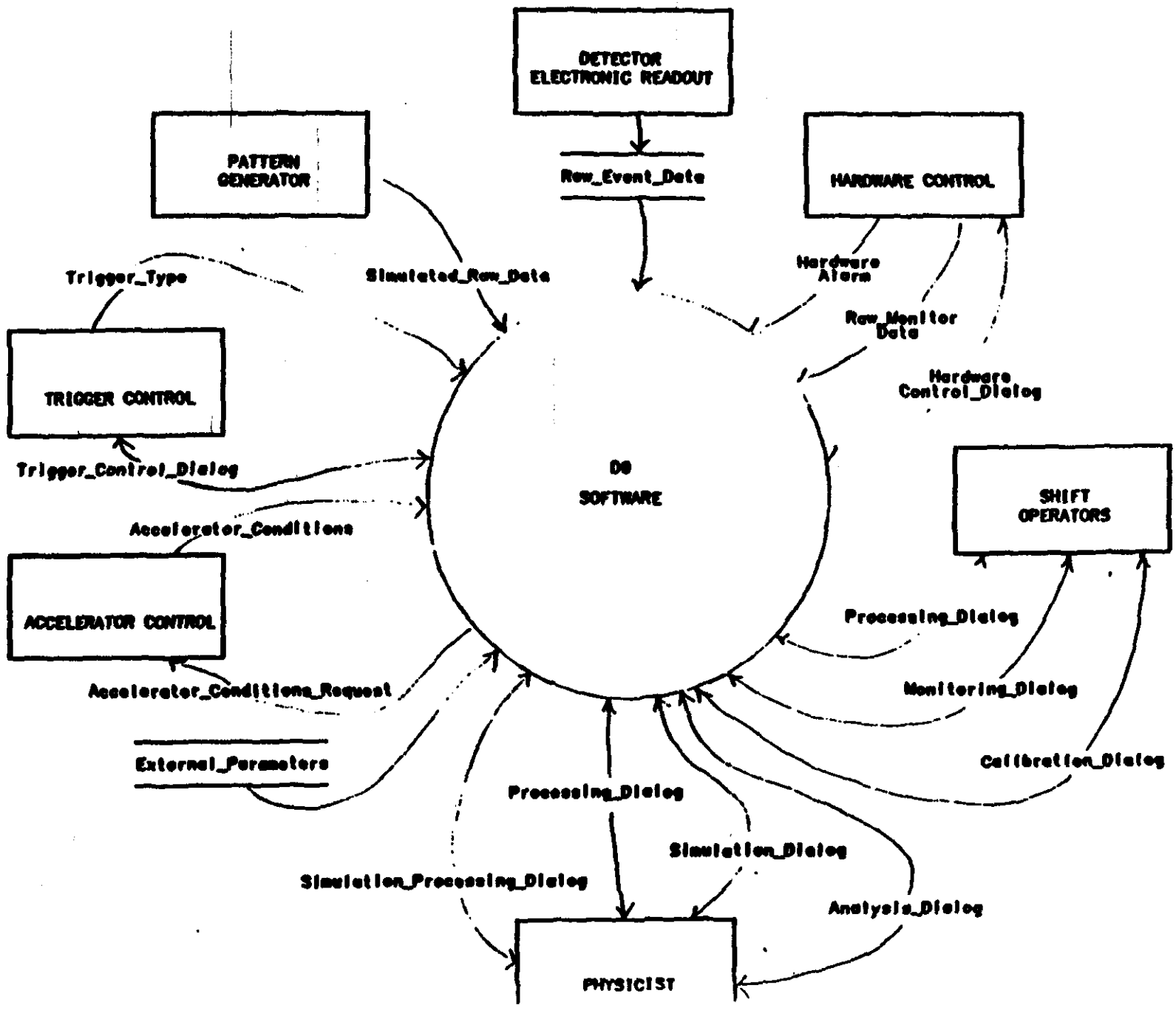
other 77*

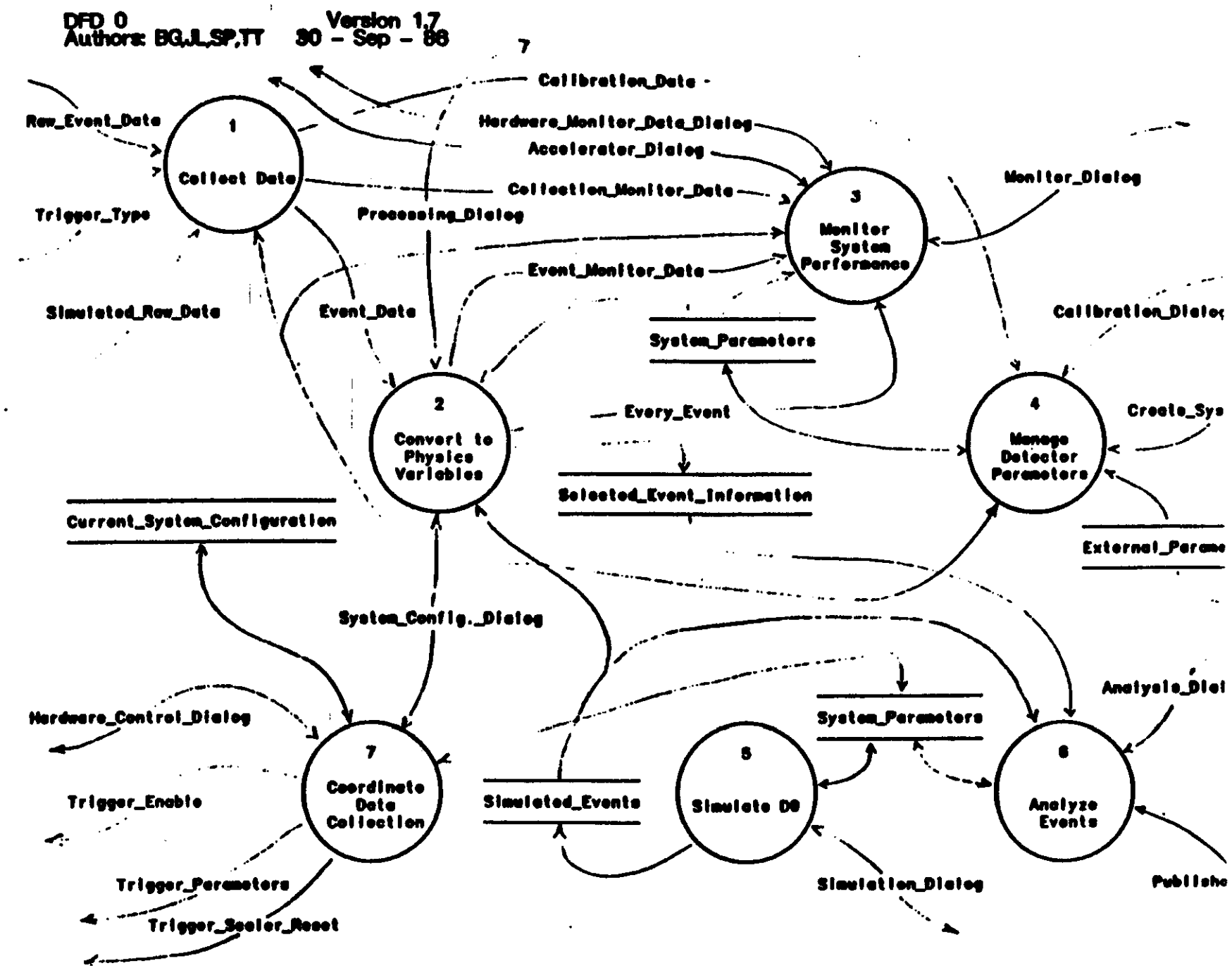
- Monte Carlo Simulation (D0geant, Showerlibrary, datadriven geometry)

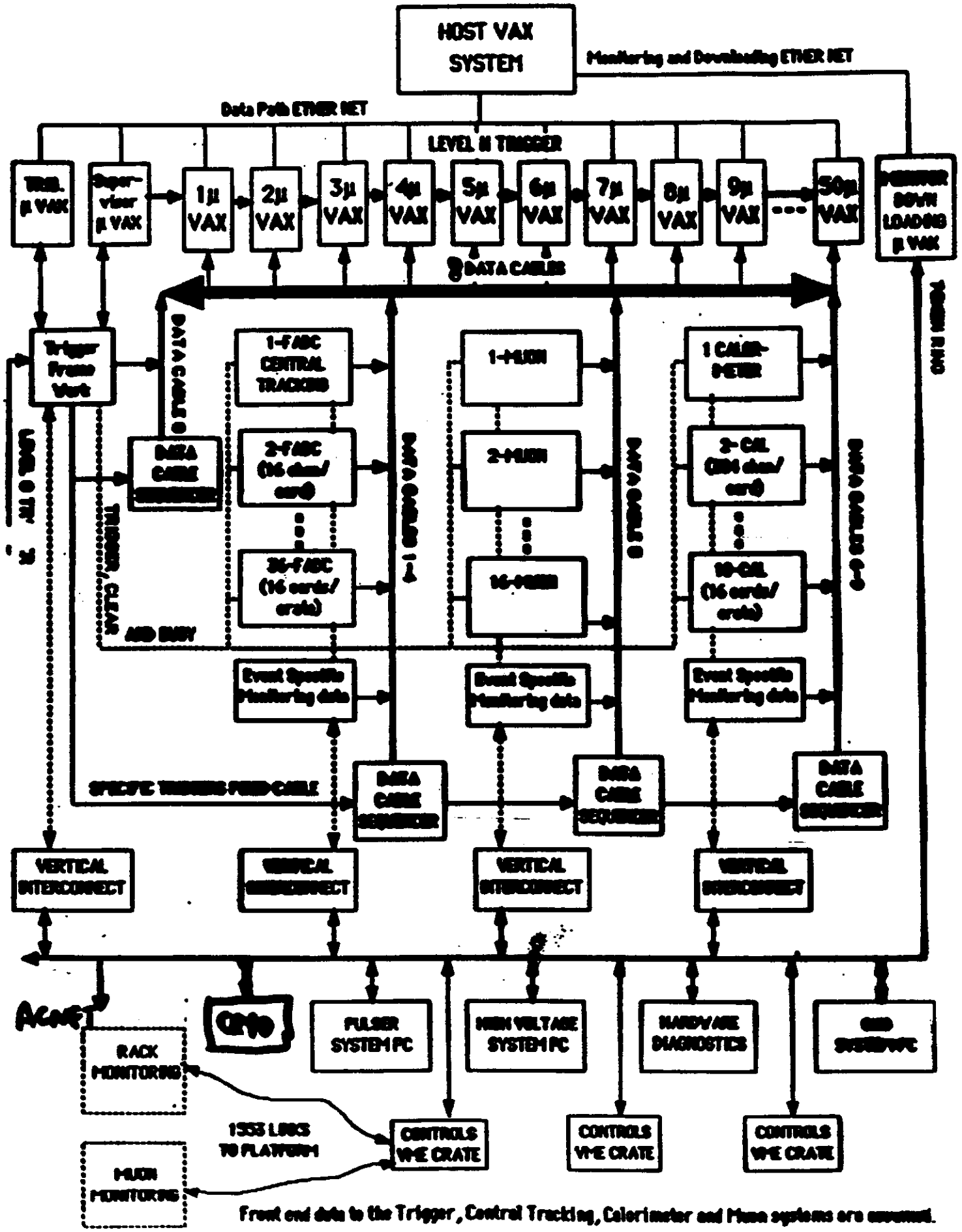
other 77*

- Offline analysis (algorithms, Risc farms, fileserver)
- Graphics DI 3000 + Emulator 3D Event display
- Library Management and release procedures (CMS/MMS, beta releases, test releases... production releases)

D0 SOFTWARE CONTEXT DIAGRAM (OVERVIEW) VERSION 2.1
Author: JTL 06 - Sept - 88




DFD 0
Authors: BG,JL,SP,TT
Version 1.7
30 - Sep - 88
7
Calibration_Data
Raw_Event_Data
1
Collect Data
Hardware_Monitor_Data_Dialog
Accelerator_Dialog
Collection_Monitor_Data
3
Monitor System Performance
Monitor_Dialog
Trigger_Type
Processing_Dialog
Event_Monitor_Data
Simulated_Raw_Data
Event_Data
System_Parameters
Calibration_Dialog
2
Convert to Physics Variables
Every_Event
4
Manage Detector Parameters
Create_Sys
Selected_Event_Information
Current_System_Configuration
External_Parame
System_Config._Dialog
Analysis_Dial
Hardware_Control_Dialog
7
Coordinate Data Collection
System_Parameters
5
Simulate DB
6
Analyze Events
Trigger_Enable
Simulated_Events
Trigger_Parameters
Trigger_Scaler_Reset
Simulation_Dialog
Publishe



Front end data to the Trigger, Central Tracking, Calorimeter and Muon systems are essential.

D-Zero Online/DAQ System

Required Functions

1. Detector installation
2. Debug detector
3. Control operation of detector
4. Calibrate detector
5. Monitor elements and system as a whole
6. Record physics events
7. Furnish real time physics feedback
8. Maintain continuous detector history

Quantitative Performance Characteristics

1. At nominal luminosity (low) 50-100K interaction/sec
2. Level 1 trigger decision in 3.5 MicroSec between cross
3. Level 1.5 trigger decision in ~ 20 MicroSec then abort
4. Level 1 yield ~200 events/sec
5. ~100K electronics channels
6. Event size 200-400KBytes
7. Each of 8 cables run at 40MBytes/sec
8. Level 2 yield 1-2 events/sec
9. Level 2 nodes have on average 250MSec to reject event
10. Nominal tape writing speed is 6250 BPI in ~6 minutes

Online System Overview

1. Multiple independent users

   Active users

   Passive users

2. Flexible partitioning of detector

3. Multi-stream data recording

4. System Components

   32 independent triggers

   100+ selectable VME digitizing crates

   8 X 40 MB/Sec data path to filter system

   2.5 MB/Sec path from filter to host

   IBM Token Ring front-end download path

   68K management and monitoring at front-ends

   General ETHERnet connectivity

Online System Overview (continued)

5. VAX transparent connectivity

Level 2 Processors (50)

Data bandwidth - High
CPU cycles availability - High
Memory and disk access - Low
Flexibility - Low
Interactivity Low

Host Processor (1 + 1)

Data bandwidth - Modest
CPU cycles availability - Low
Memory and disk access - High
Flexibility - Modest
Interactivity Modest

Workstations (16)

Data bandwidth - Low
CPU cycles availability - Modest
Memory and disk access - Modest
Flexibility - High
Interactivity High

Migrate applications to match capabilities

Event Filtering (Concept)

Establish a set of algorithmic tools

Missing Pt › X
N jets › E
Muon Pt › X
Etc.

A filter is a sequential script of such tools

Apply filter according to trigger (32) bits

Label events by a filter (128) bit mask

Pass unbiased sample as check

Routing by trigger bit to particular node
and "tool" can generalize to other functions.

Event Monitoring
Calibration

Utility List

1. Zebra - CERN: Memory and data management
2. RZ/DBL3 - LEP3: Limited relational database
3. DI-3000 - Commerical: 3-D graphics system
4. COMPACK - D-Zero: Menu/Screen management system
5. ITC - D-Zero: Multi-link inter-task communication
6. ELNCON - D-Zero: Inter-task communication ELN/VMS
7. DAQ - CDF/CD: Global shared common
8. Rdb - Commercial: Relational database

Functions of COOR

1. **Coordinate Activities**

   Subsystems have consistent instructions

2. **Allocate Resources**

   Prevent independent user conflicts

3. **Distribute Operation Information**

   Configuration and Status

4. **Record run control information**

   Rdb Tape/Run Database

   Zebra Begin Run banks

   Configuration Archive

Data Collection Task
(TAKER)

1. Simple data taking interface

2. Selects from prefined configurations

3. Selects run characteristics

   Fixed/Indefinite length
   Event logging On/Off
   Volume spanning On/Off

4. Begins/Ends runs

5. Monitors progress of this run

   Events Taken
   Events Recorded
   Event Rate

6. Display general data taking activity

Calibration Task

CALIB in level II nodes.

1. Electronics Calibration

2. Involves active control of pulser and digitizing crate

3. Zebra formated output

   Pedestals & Gains

   Element Status Flags

4. Calibrations stored in DBL3 database

5. Event Access

   As "Tool" in Level 2 node

   By filter bit mask in Host global common

   DECnet server extension of above

   From previously recorded data

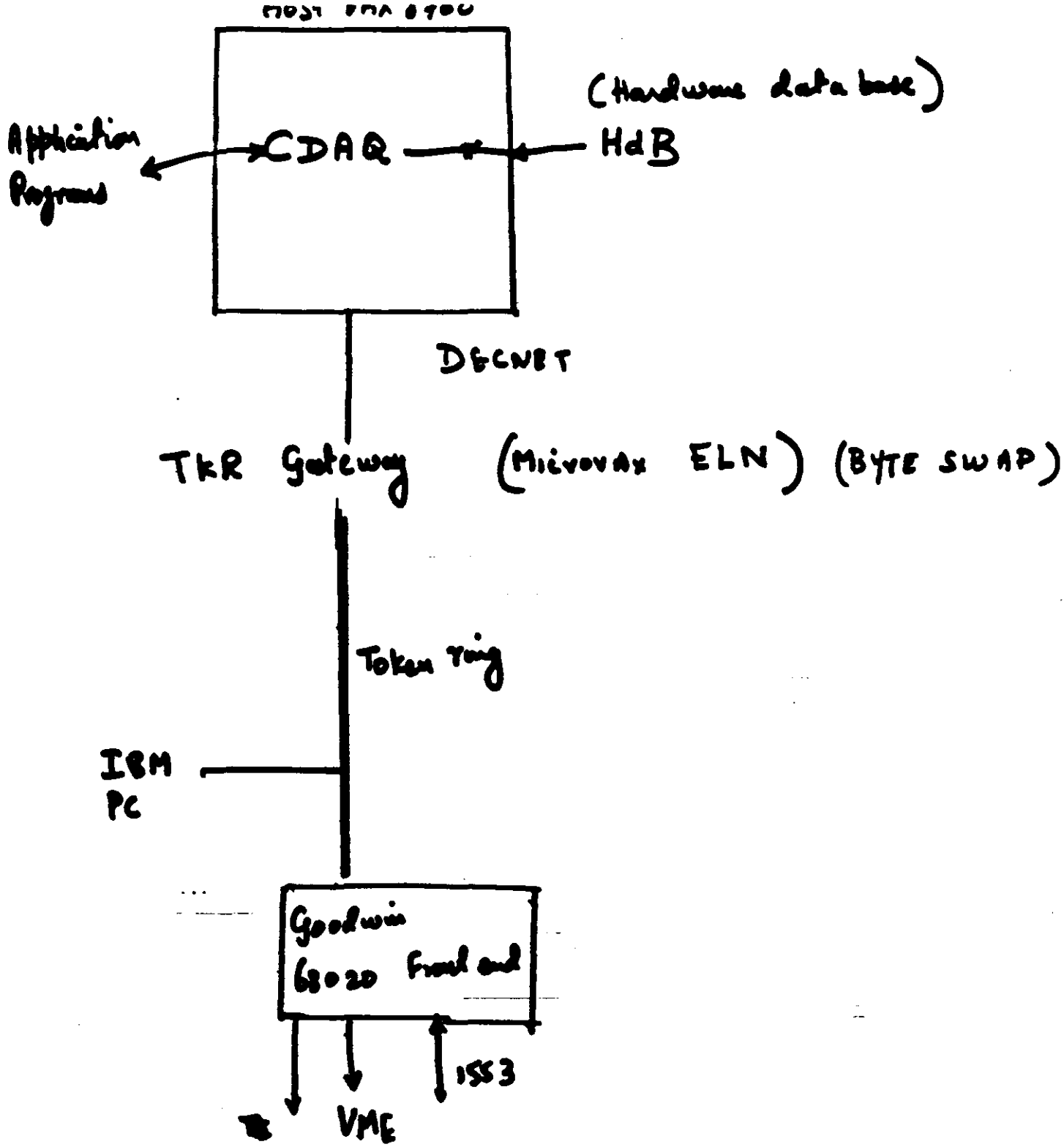
Monitoring/Control

Utilities Supporting Token Ring connected components

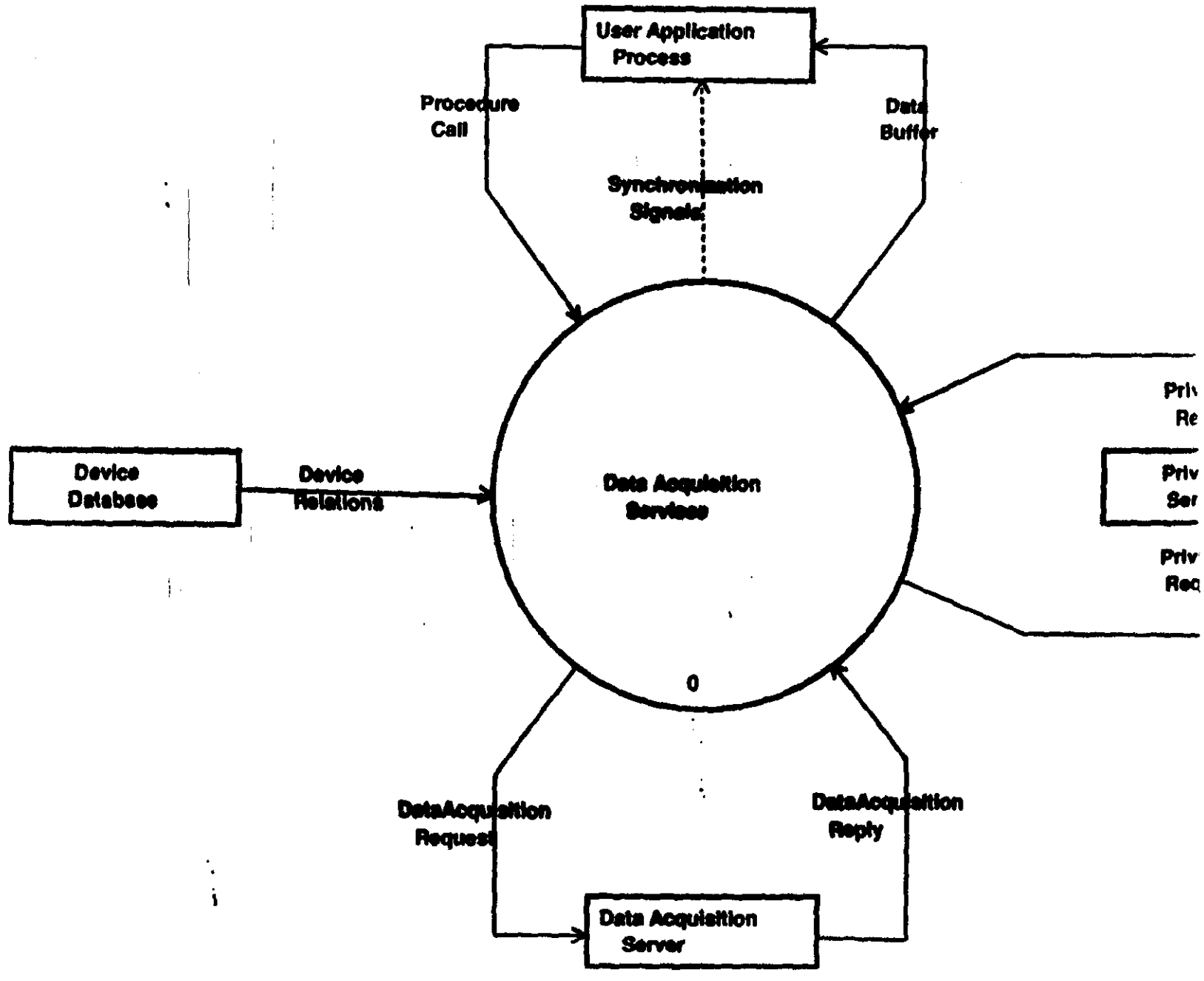
1. Rdb electronics database and support utilities
2. Control path data access services
3. Parameter page

Alarm handling system

1. Alarm dispatcher
2. Alarm display
3. Alarm logger
4. Heartbeat processor


(Hardware data base)
Application Programs
CDAQ
HdB
DECNET
TKR Gateway
(MicroVAX ELN) (BYTE SWAP)
Token ring
IBM PC
Goodwin 6020 Front end
1553
VME

Data_Acquisition_Services



Zero
design notes

HdB → Uses DEC RdB relational data base Product.

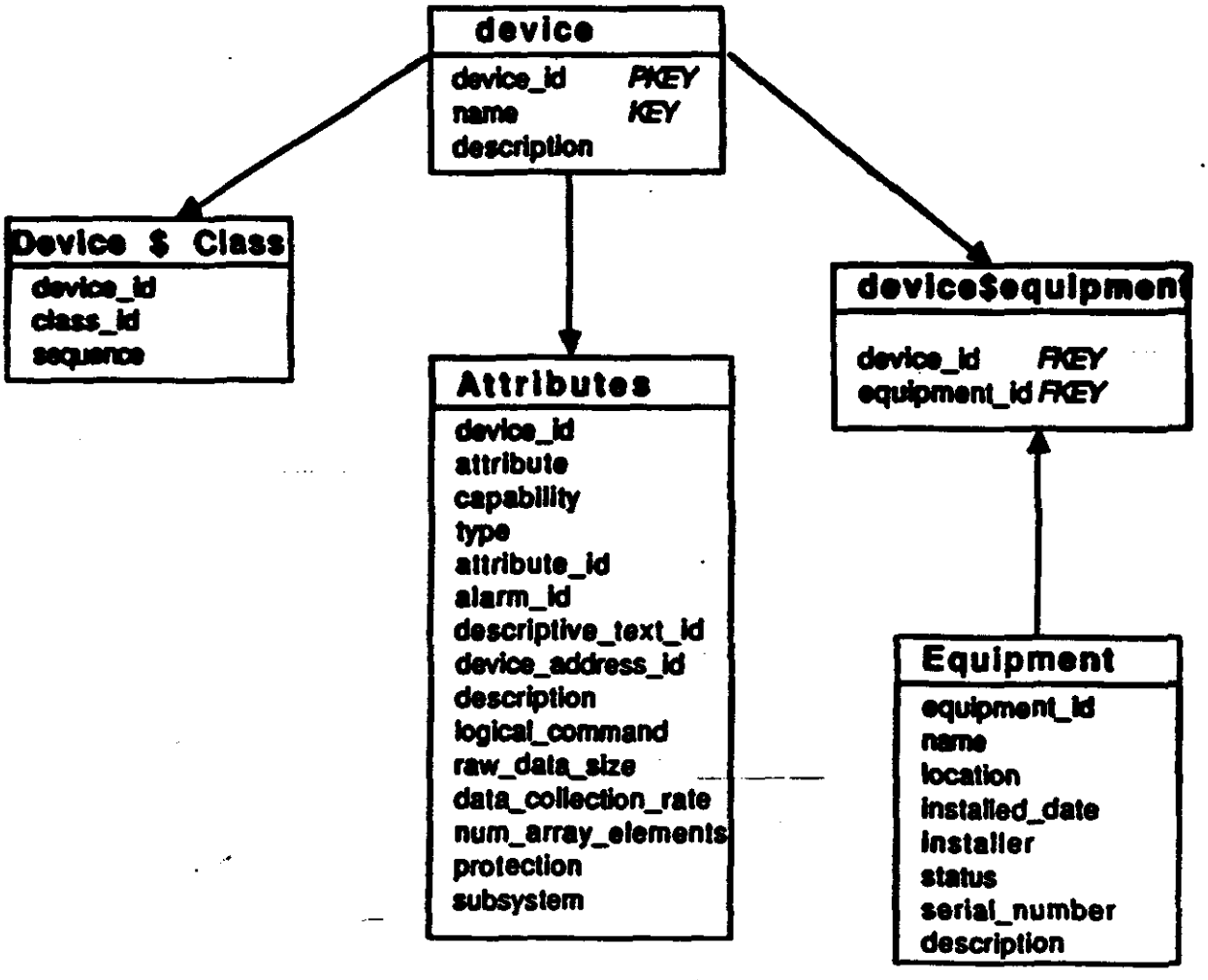
The Hardware Monitor Database

A. Waller
M. Frey

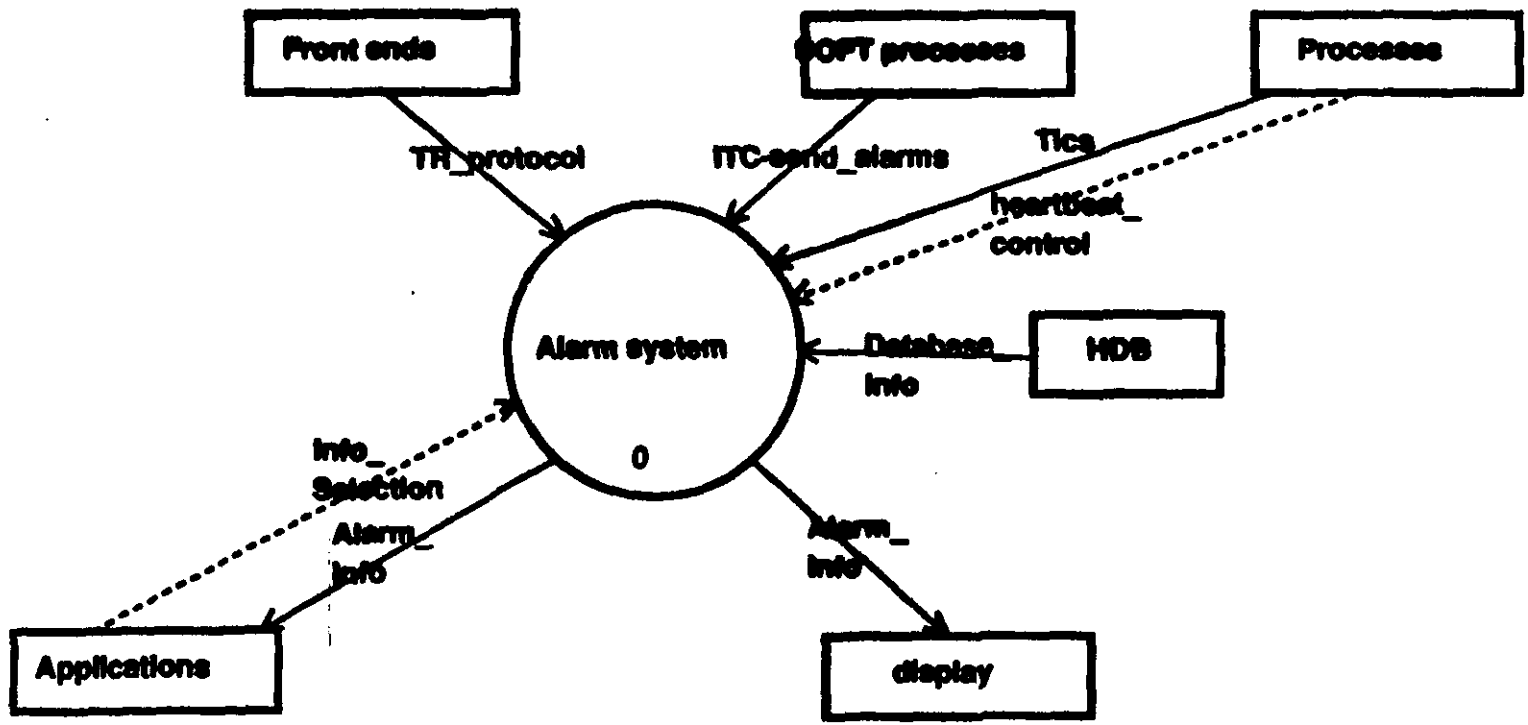
Extensively designed. Uses Browser + Batch Entry to enter data.

Down-load Goodwin tapes using Stress Program

• Working DQ example was not implemented early on. Some mods needed to speed RdB up, optimize it. Stan Krzyzdzinski needs help.


device
device_id PKEY
name KEY
description
Device $ Class
device_id
class_id
sequence
Attributes
device_id
attribute
capability
type
attribute_id
alarm_id
descriptive_text_id
device_address_id
description
logical_command
raw_data_size
data_collection_rate
num_array_elements
protection
subsystem
device$equipment
device_id FKEY
equipment_id FKEY
Equipment
equipment_id
name
location
installed_date
installer
status
serial_number
description

Context-Diagram;7
context for the alarm system



dbEdit Screen Example

```
+-----------------------------------dbEdit-----------------------------------+
| ?=HELP EXIT                                                                |
+----------------------------------------------------------------------------+
| 68K    Download    HDB                                                     |
+--------68K--------------Alarms-------+-------------Analog Alarm------------+
| ?=HELP              | ?=HELP         | ?=HELP NEXT PREV READ SET >PUSH <POP |
+---------------------+----------------+--------------------------------------+
| Alarms              | Analog_Entry   | Node        > 33                     |
| Analog_Tables       | Binary_Entry   | Entry       > 0120                   |
| Binary_Tables       | Comment_Entry  | Priority      00                     |
| Comment_Tables      | Analog_File    | Name          PRCRR1                 |
| Directory           | Binary_File    | Attribute     0120                   |
| Tables              | Comment_File   | Subsystem     00000000               |
| Memory_Access       | Reset_ATarms   | Path          00000000               |
+---------------------+----------------+ HDB_ID        00000000               |
|                                      | Flags         1000000000000000       |
|                                      | Nom//Cnt      20000000               |
|                                      | Tol//Txt      80000000               |
|                                      +--------------------------------------+
+----------------------------------------------------------------------------+
|Opening CDAQ and HDB...                                                     |
|CDAQ and HDB opened successfully                                            |
|CDAQ returns                DQS_SUCCESS                                     |
+-----------------------------------Messages---------------------------------+
```

```
                    all                          ^   0  v   0
b!test_alarm3 .volt b!test_alarm2 .volt b!test_alarm1 .volt

                    subs=1                       ^   0  v   0

                                  detail
 test_alarm3 .volt 20-MAR 16:29:30  another test
 test_alarm2 .volt 20-MAR 16:29:30  HDB not defined on this node
 test_alarm1 .volt 20-MAR 16:29:30  This is a test

^v Sel message          PF1 X:Exit PF2:Help          0    Level : 2 FROZEN
```

Alarm display with detail window

Dial up any device in database.

Get info on device.

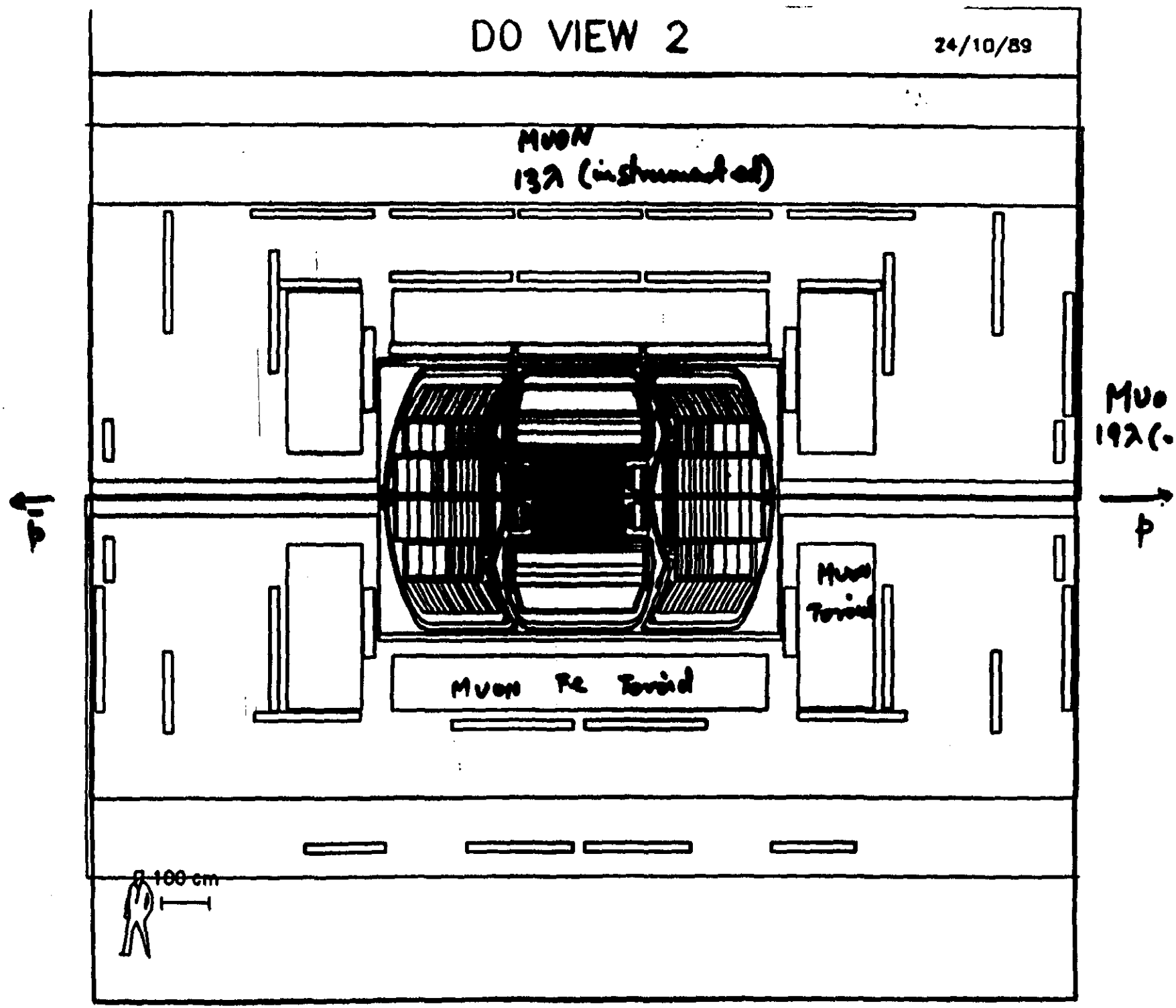
Read it periodically

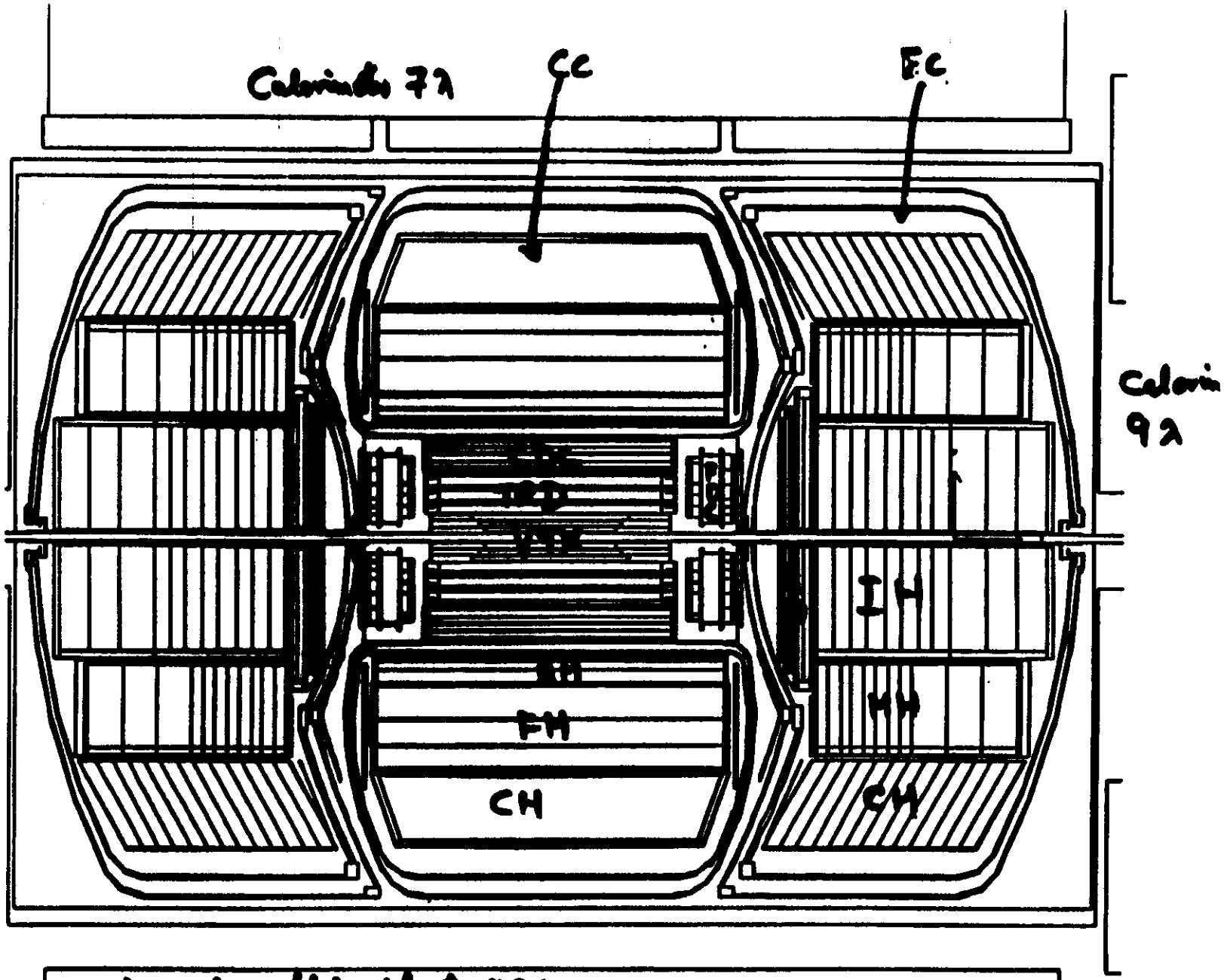
Set devices

Store readings in NTUPLES

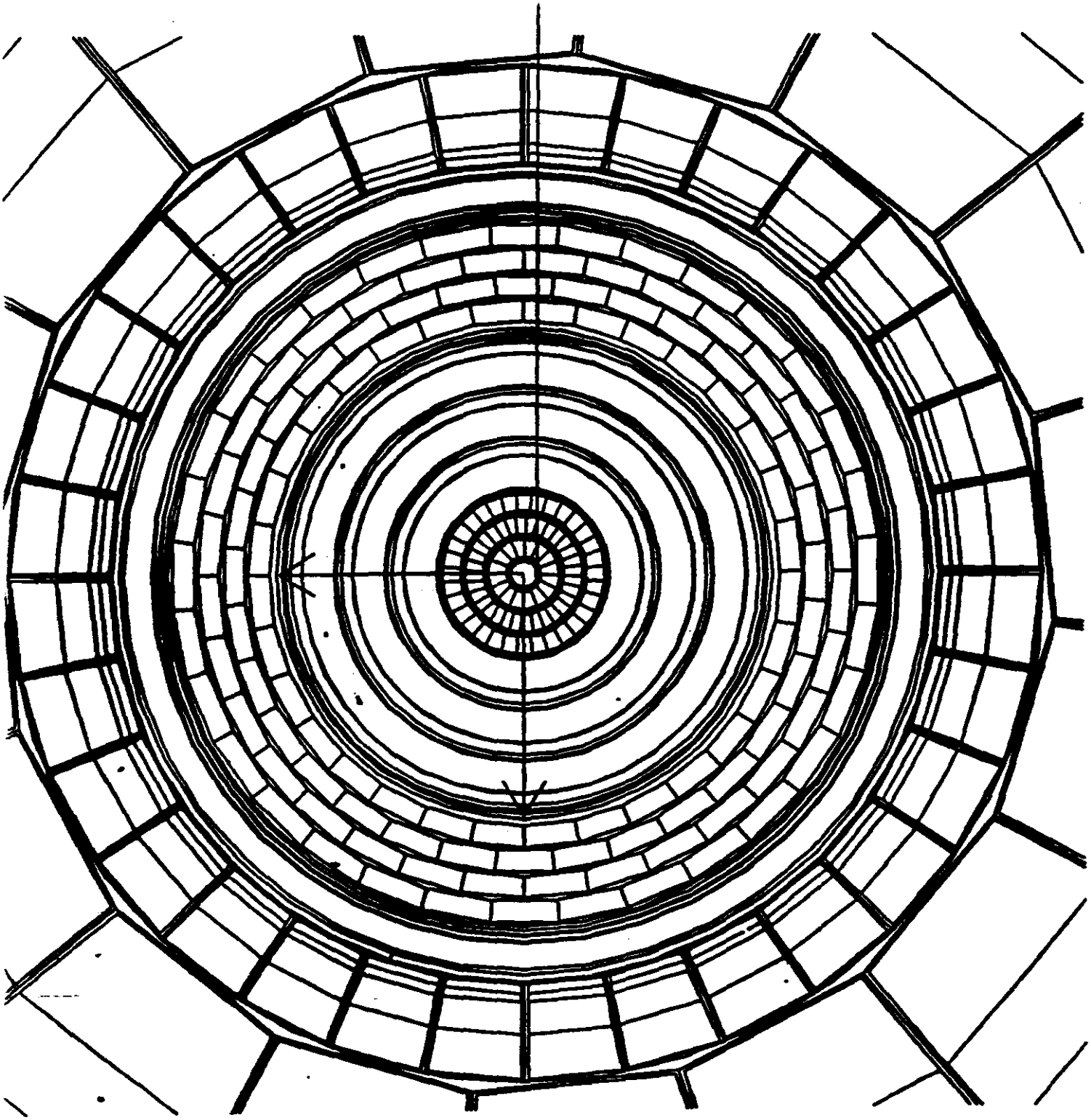
Display as strip chart

Playback NTUPLE.

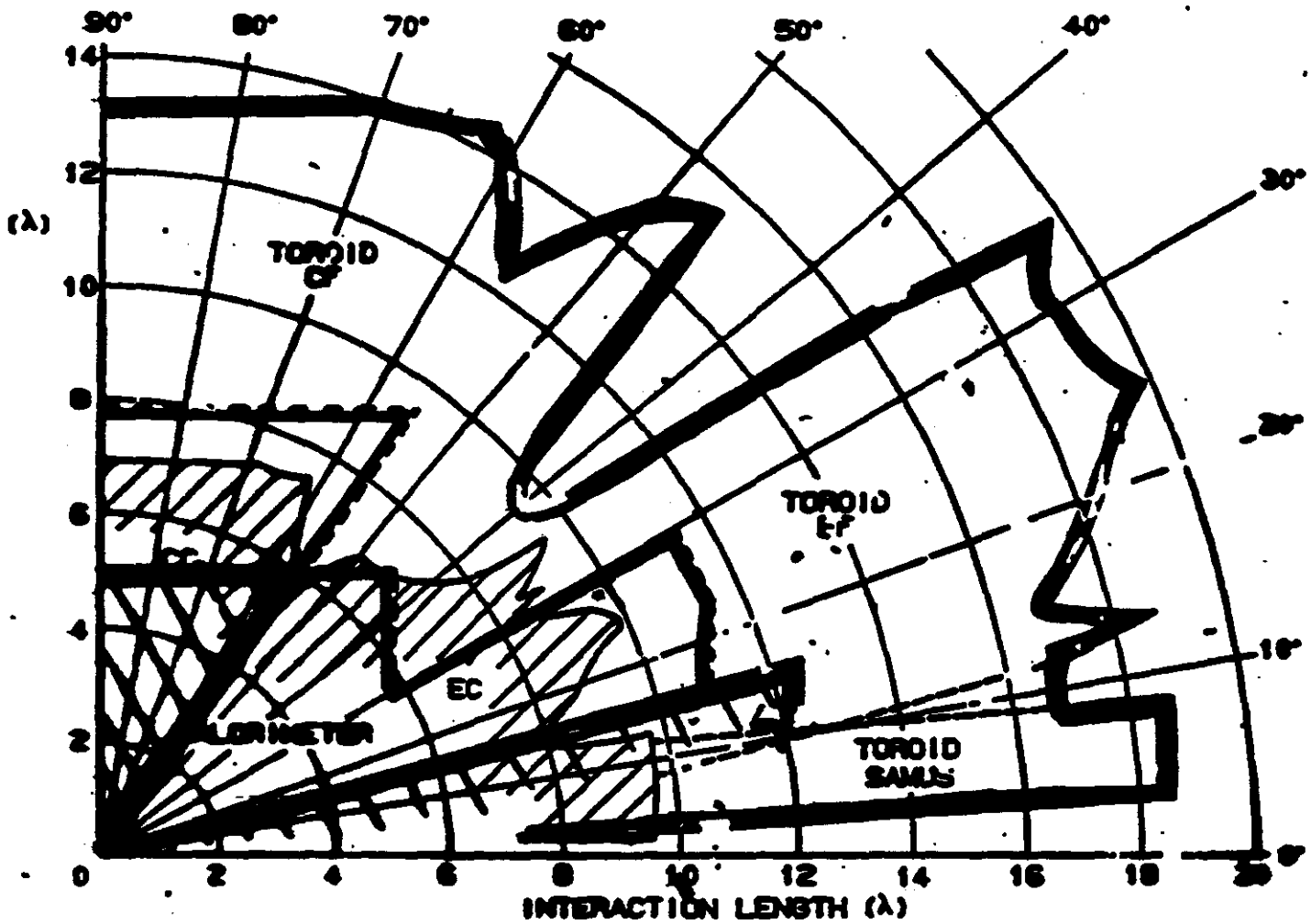

DO VIEW 2
24/10/89
MUON
13λ (instrumented)
MUON
Toroid
MUON Fe Toroid
100 cm
p



Uranium / liquid Argon
Highly Segmented 0.1 x 0.1 Δη x Δφ pads
Stable Calibration
e/π ≈ 1.05 measured > 50,000 channels in Calorim



DØ (instrumented)



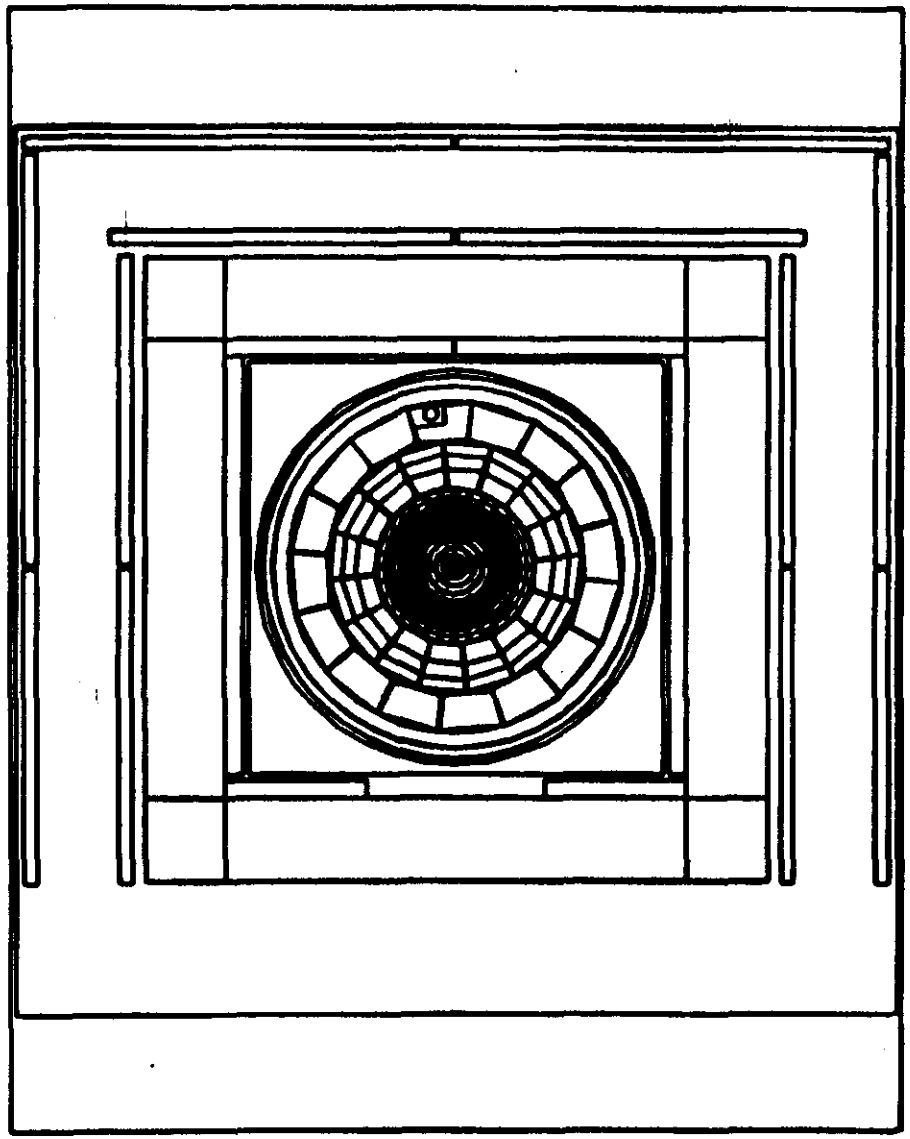
TOTAL INTERACTION LENGTHS OF THE CALORIMETER AND MUON SYSTEM

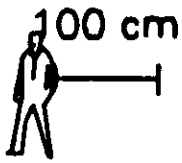
CDF (instrumented)

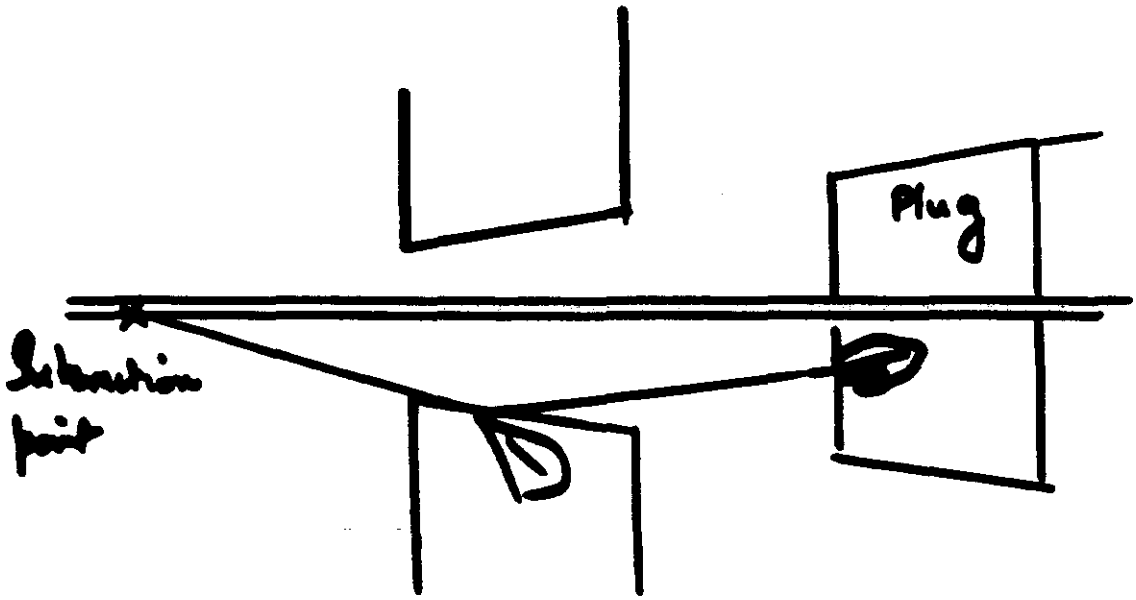
proposed CDF upgrade

DO VIEW 1

24/10/89







Mismeasurement of p_T

- Using the DØ Monte Carlo
  - reduced number of cracks in the design
  - too late to change some design flaws
  - Cryostat problem correctable with inter-cryostat detector
  - SSC detectors can benefit a lot from this experience.

DØ GEANT ⟷ describes DØ Geometry

Data driven

DØ in vertices

Digitizes

GEANT 3.16 with GHEISHA.

DØ planned Hermeticity studies using full showers.

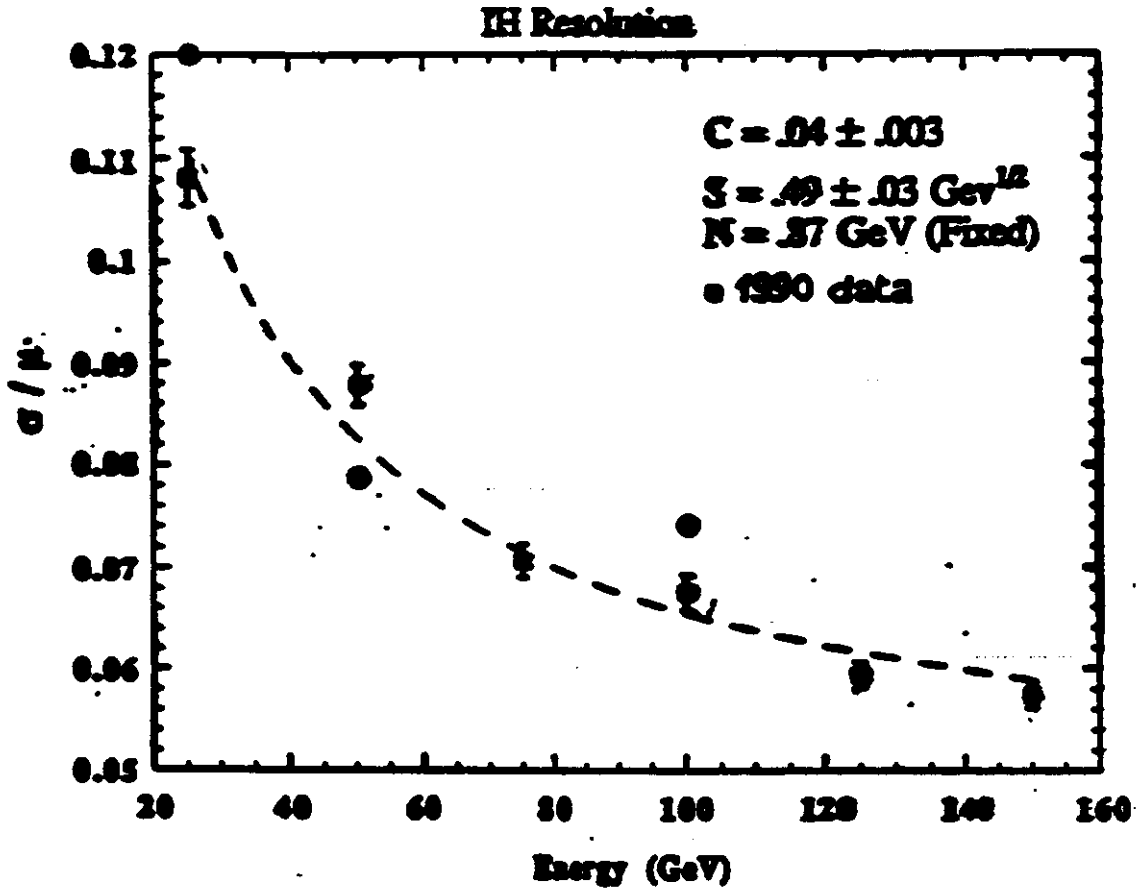
Quality factor. $Q = \iint \frac{\Delta\phi}{\epsilon} \, dy \, d\phi$

1 event = 1.5 hours CPU

100k events Generated. Microvax farms

RISC farms

Showerlibrary DØ invention.

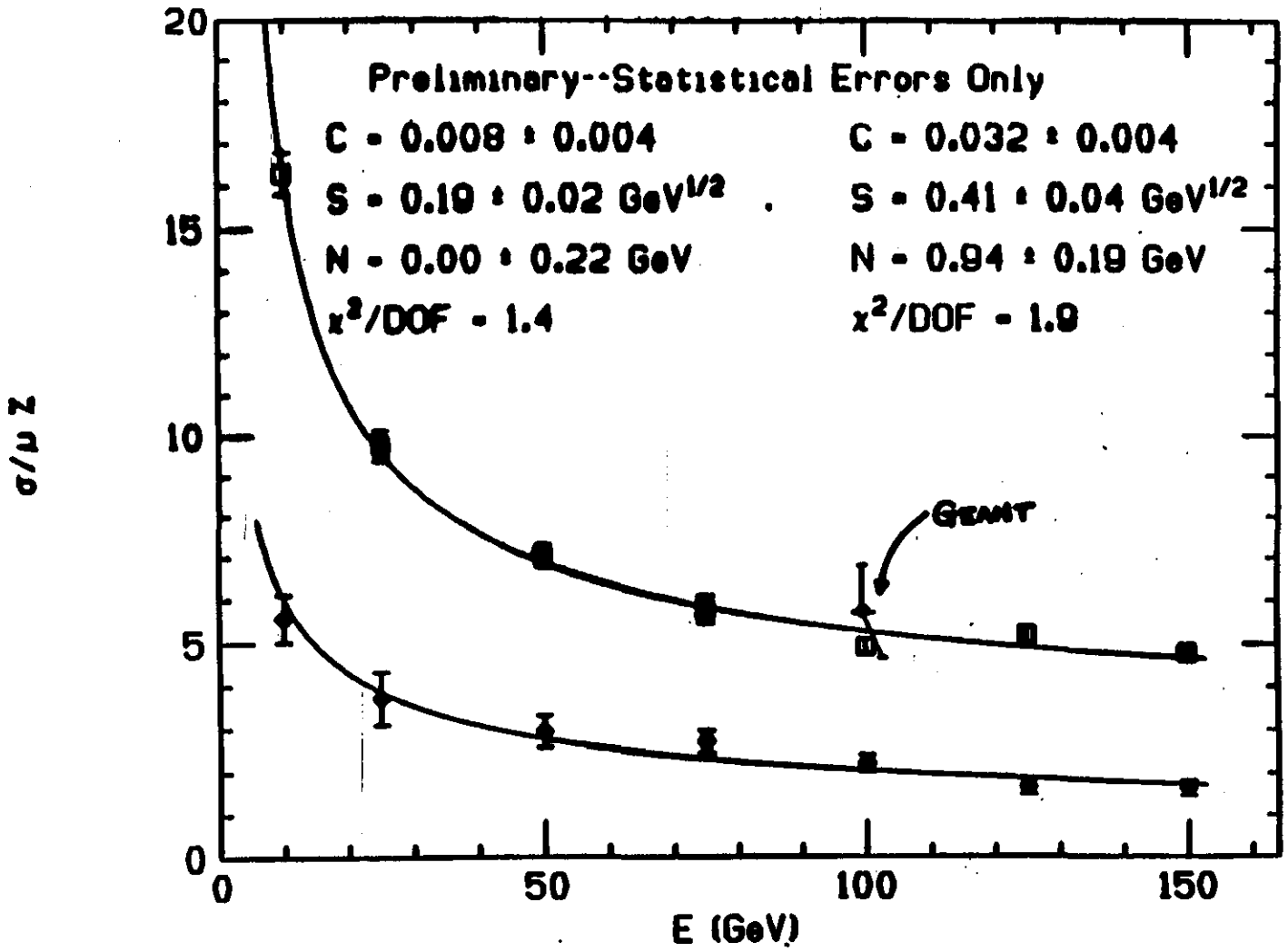


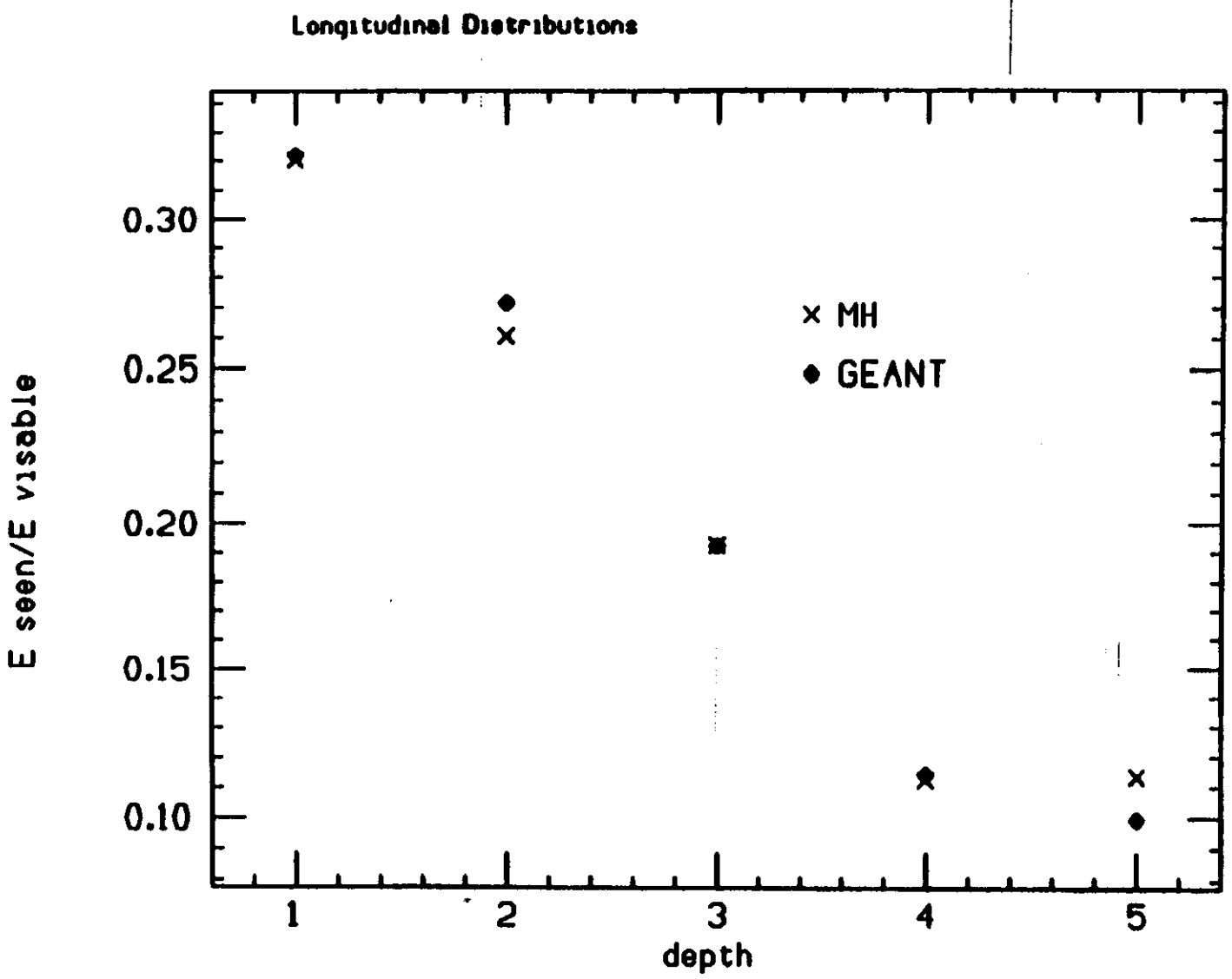
DØGEANT314: • Standard cutoffs
• 100 keV / 10 keV EM

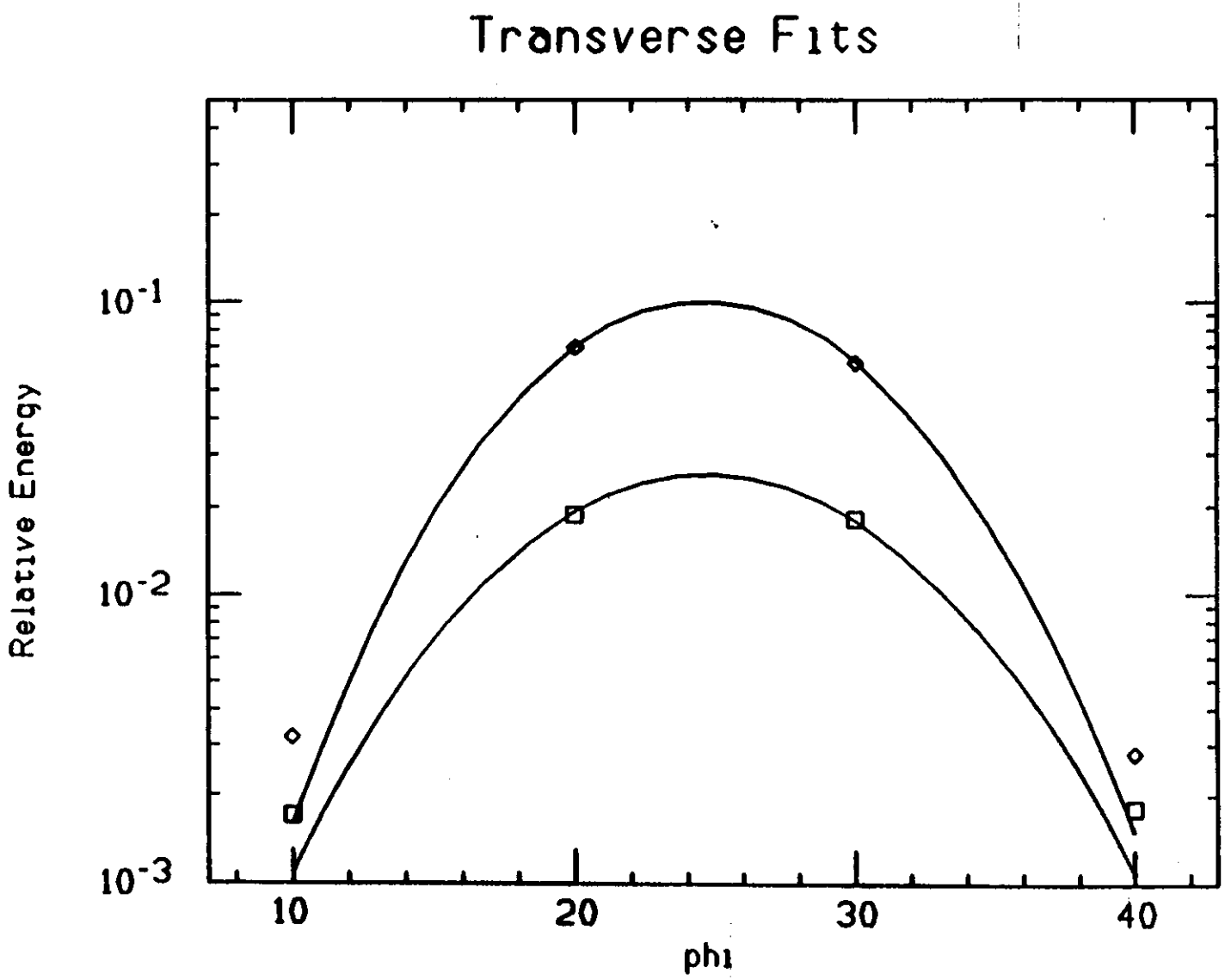
e/π ≈ 1.05 @ 100 GeV

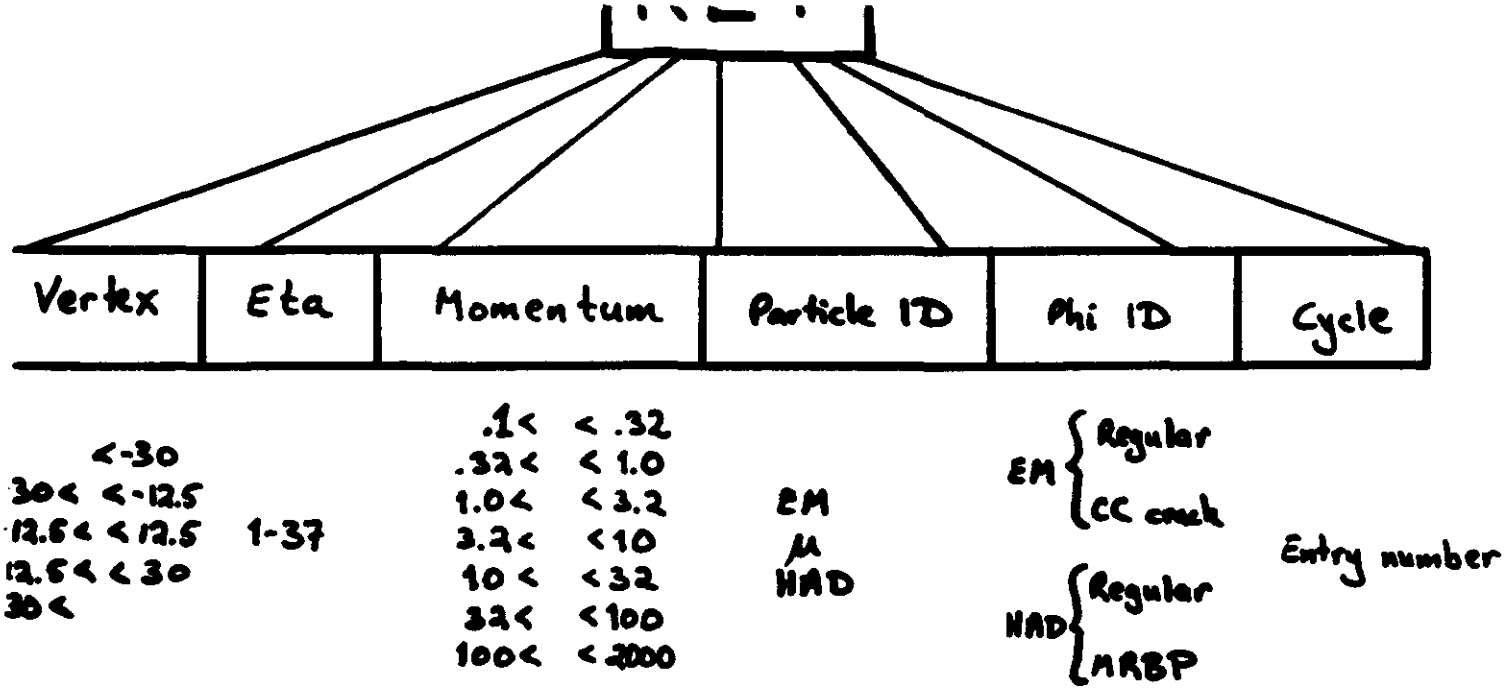
(We are running low cutoffs @ 25, 100, 150 GeV)

MH e & π Resolution




Longitudinal Distributions
0.30
0.25
0.20
0.15
0.10
E seen/E visable
× MH
● GEANT
1
2
3
4
5
depth


Transverse Fits
Relative Energy
10^{-1}
10^{-2}
10^{-3}
10
20
30
40
phi



For every track: WRITE(IOUT) PACK_KEY, NS, (ARRAY(I), I=1, NS)
where ARRAY contains all GCAH information associated with the primary track.

Two runs ⇒ 1 output file ~ 2K blocks

Sort the files on PACK_KEY

Merge 10 files at a time

CONVERT ⇒ keyed access file with index

/FDL ⇒ optimize the index file

Original implementation (J. Womersley) employed the RZ keyed-access storage system of ZEBRA.

Not enough memory
Much too slow in accumulation and access
Limit on number of cycles

Regular Keyed Access Write was tried.
Much too slow in accumulation

Current DØ method based on VAX VMS utilities

SORT / MERGE / CONVERT

based on a key structure.

The Shower Library method is currently employed within the framework of DØGEANT ⇒ Allows one to track selected particles through the full simulation (currently used for muons)

Loop over input particles & determine Key vector.

Fetch number of cycles from auxiliary file and pick random # between 1 and CYCLE_MAX and get that entry

Scale energy of Library entry to that of particle

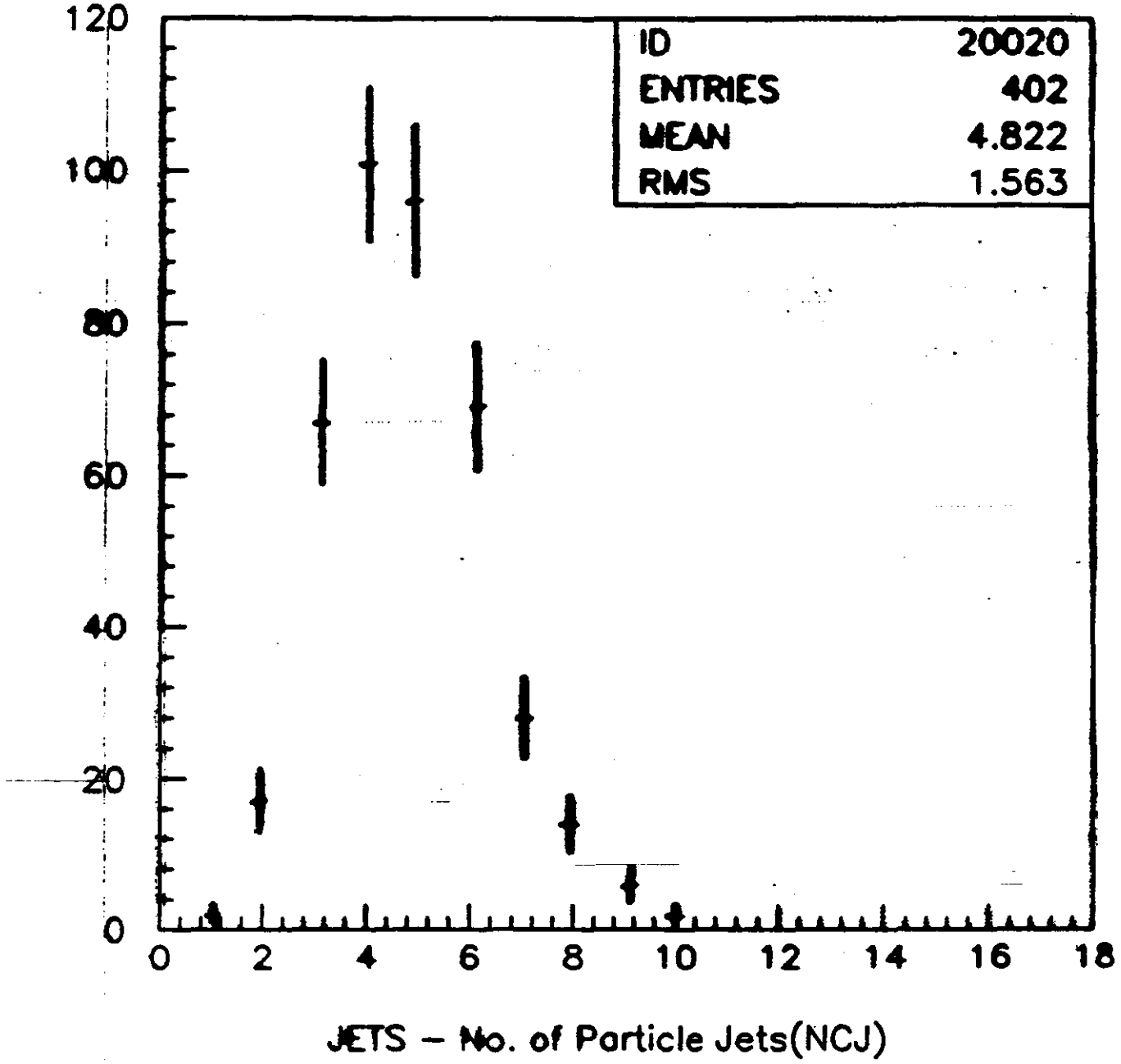
Deposit energies in calorimeter working array

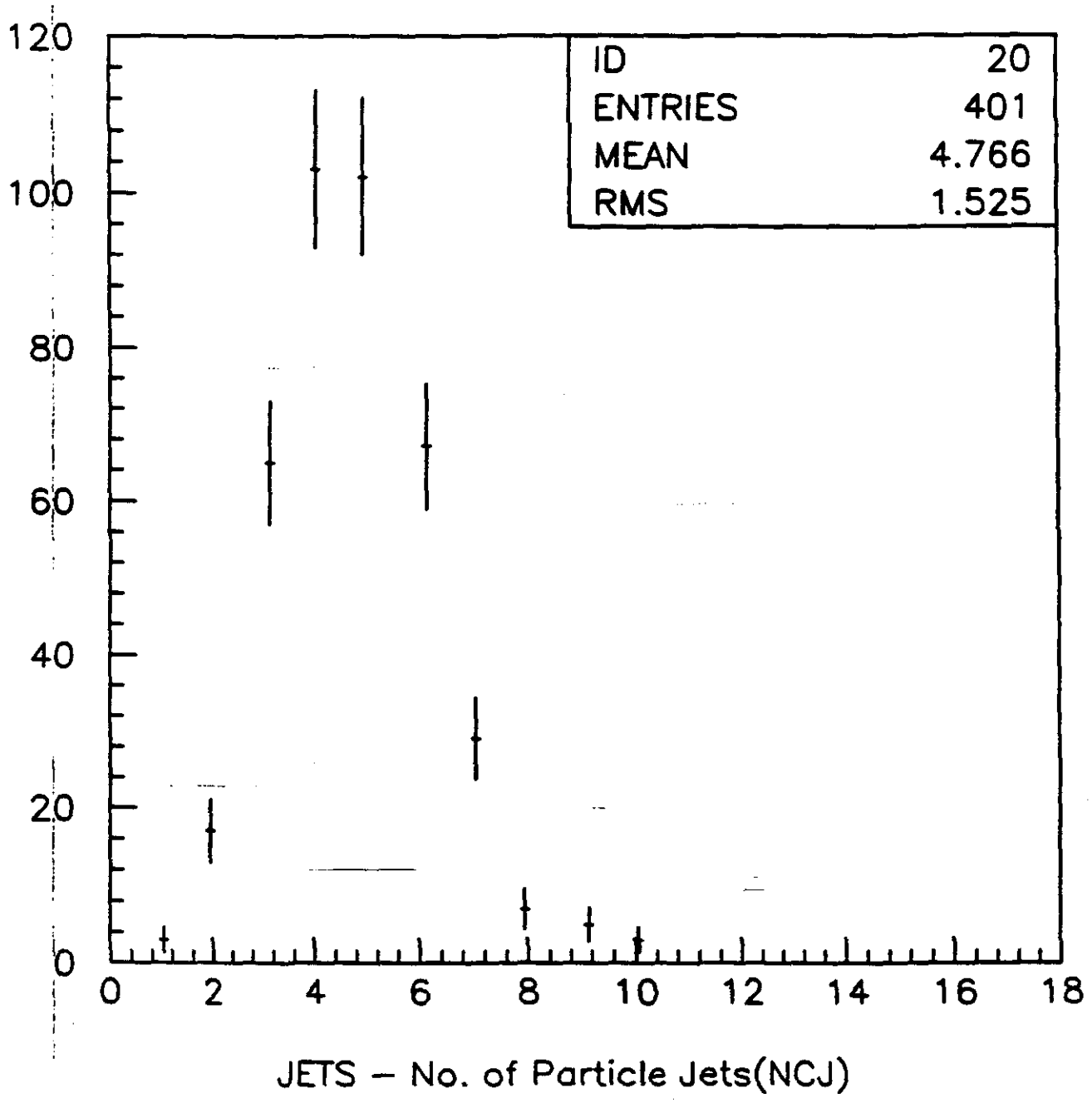
Timing Results

For a sample of two-jet events (Pt 160-240, VAX 3100)

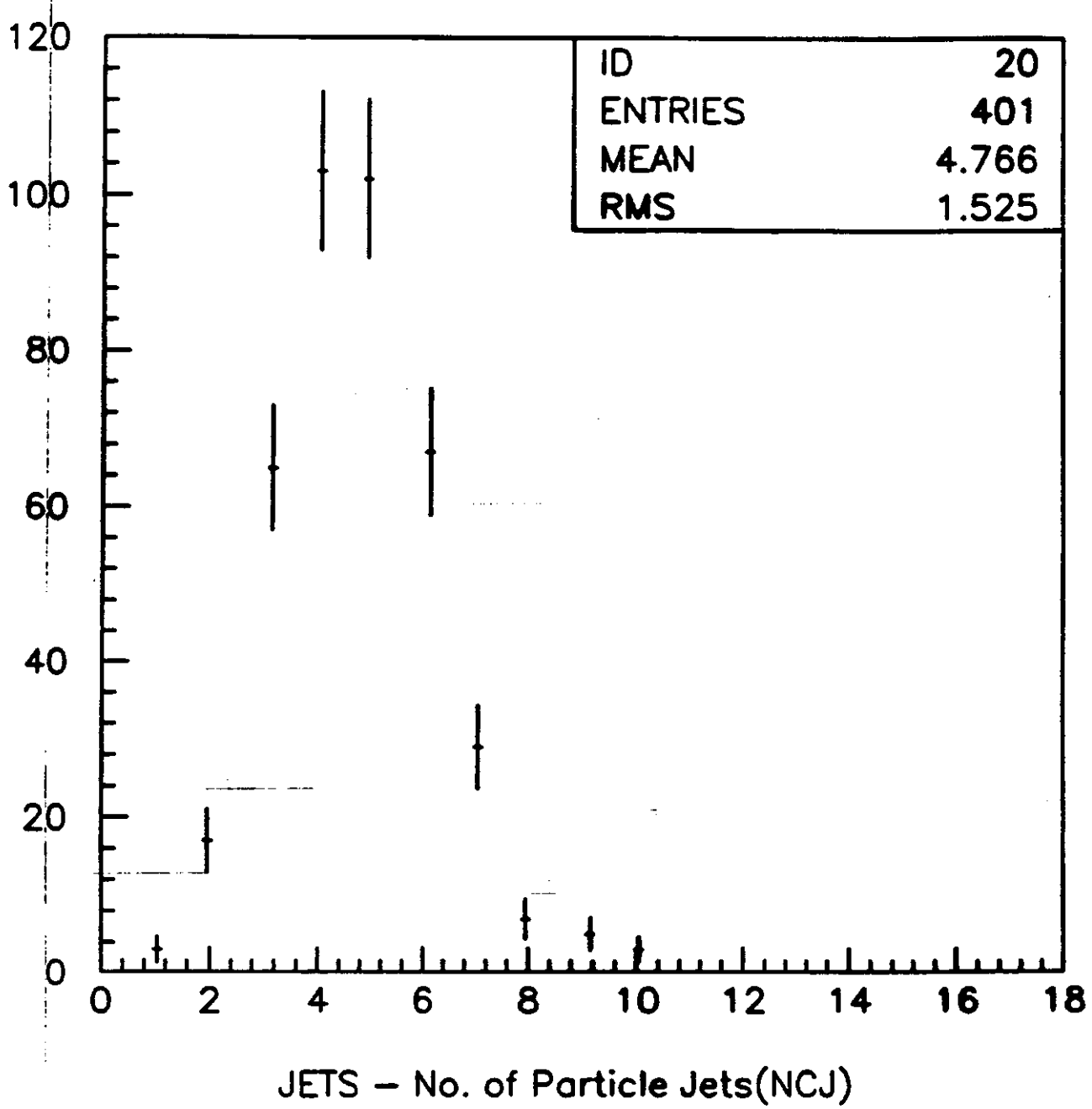
Full GEANT		Shower Library	
CPU time	Elapsed time	CPU time	Elapsed time
3130 s	3590 s	7.2 s	20 s

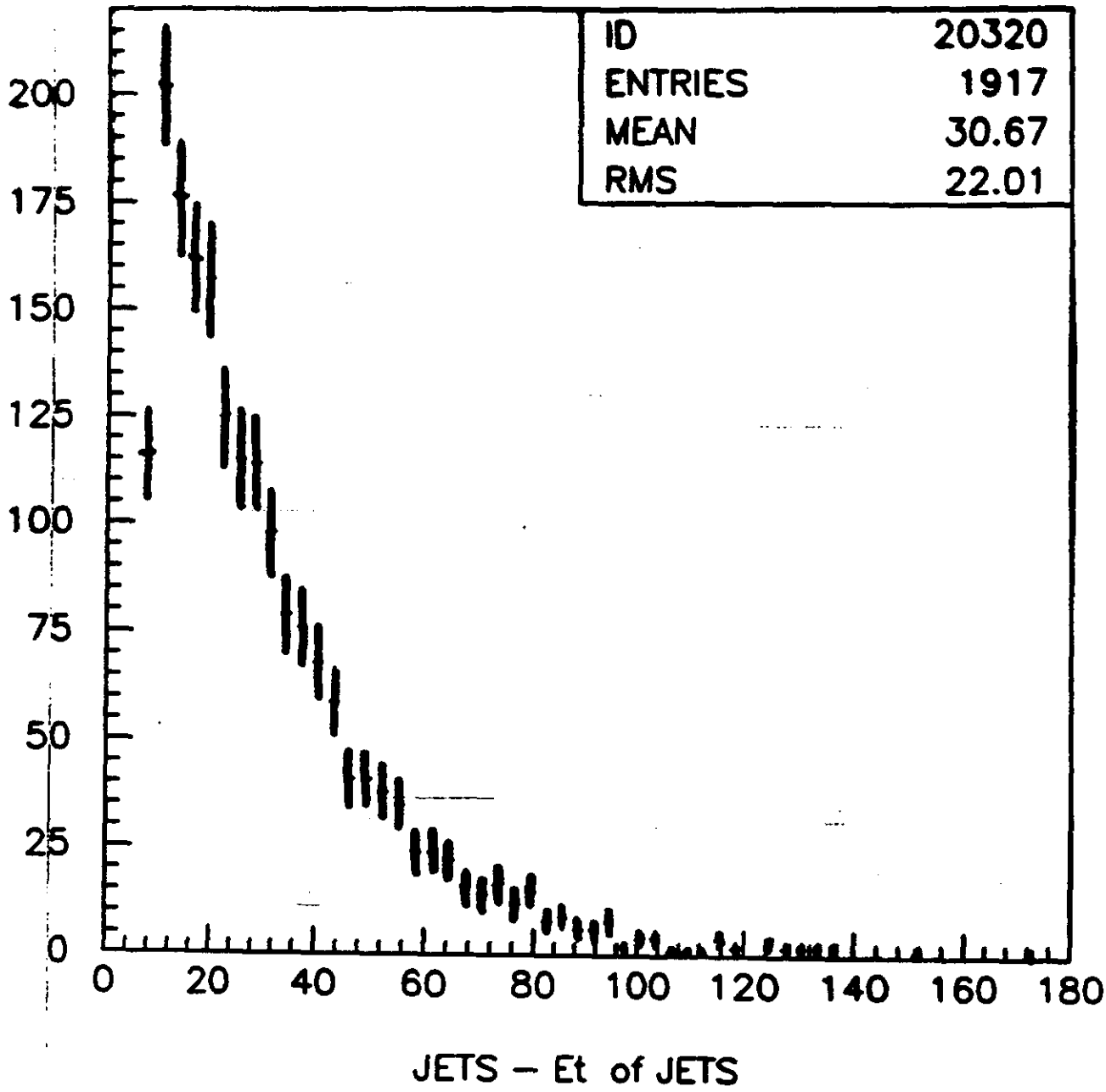
Speedup factor 422 in CPU time

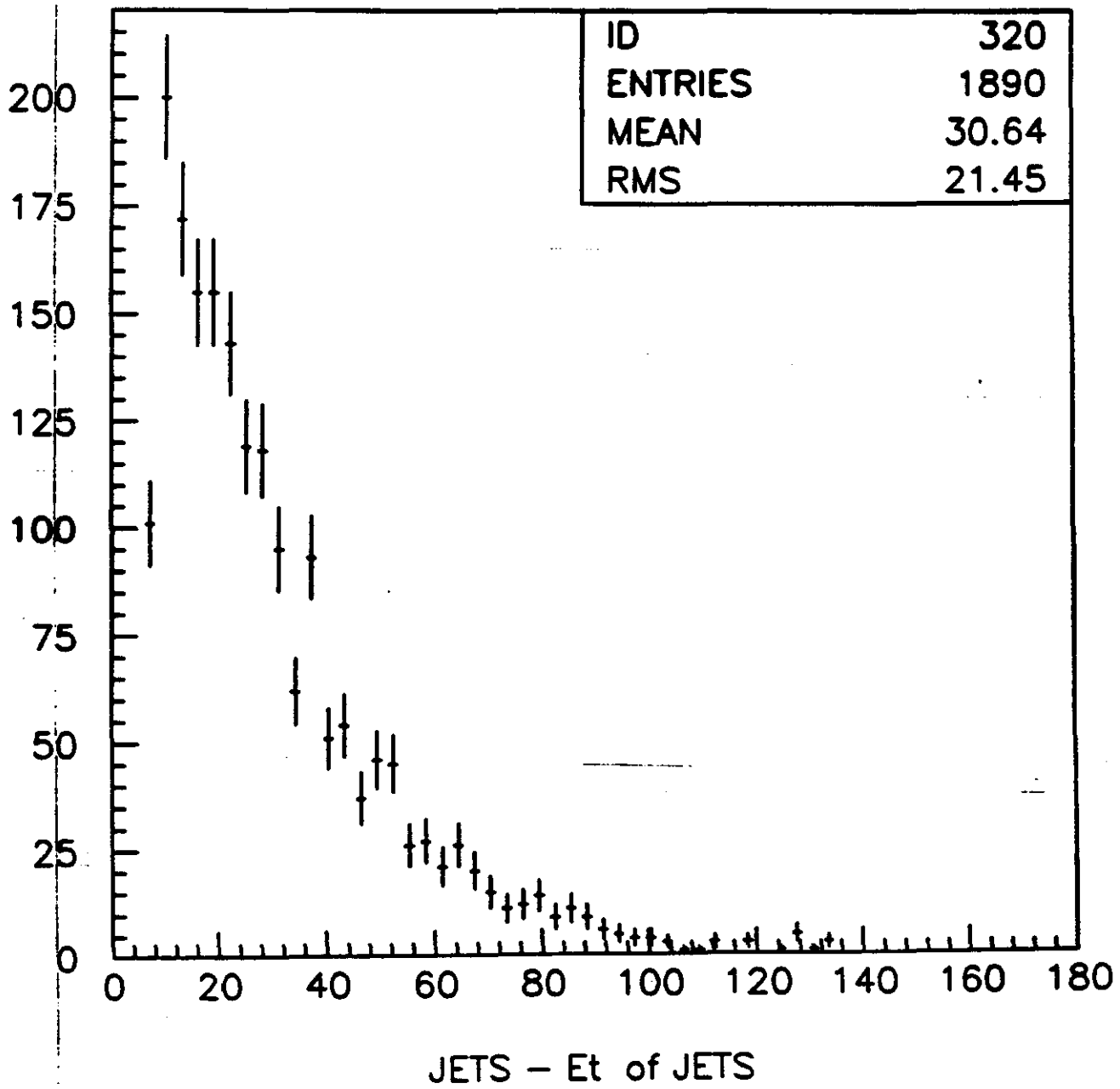

ID 20020
ENTRIES 402
MEAN 4.822
RMS 1.563
120
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
18
JETS – No. of Particle Jets(NCJ)


ID 20
ENTRIES 401
MEAN 4.766
RMS 1.525
120
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
18
JETS – No. of Particle Jets(NCJ)

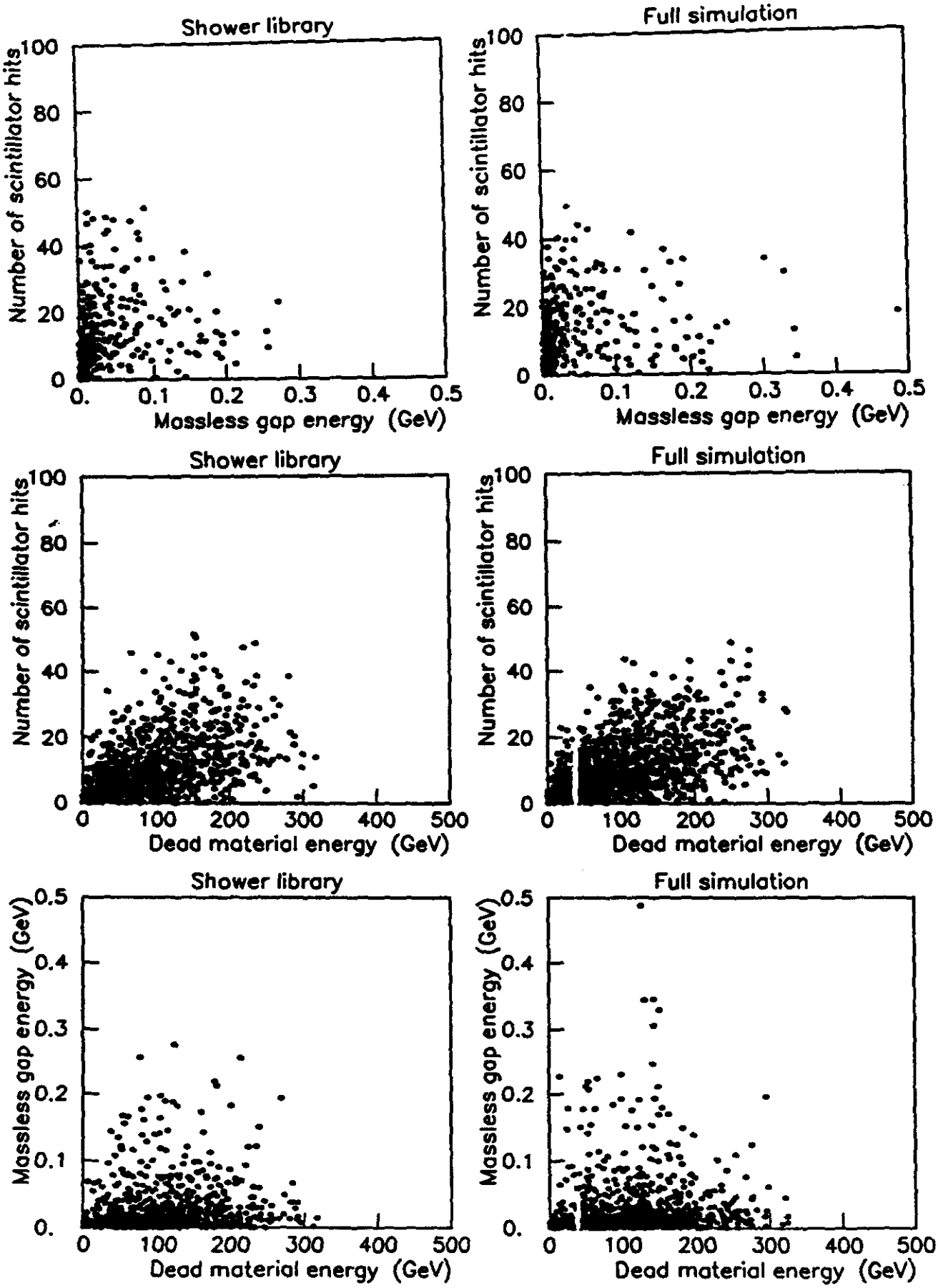
A Sample of 400 $t\bar{t}$ events



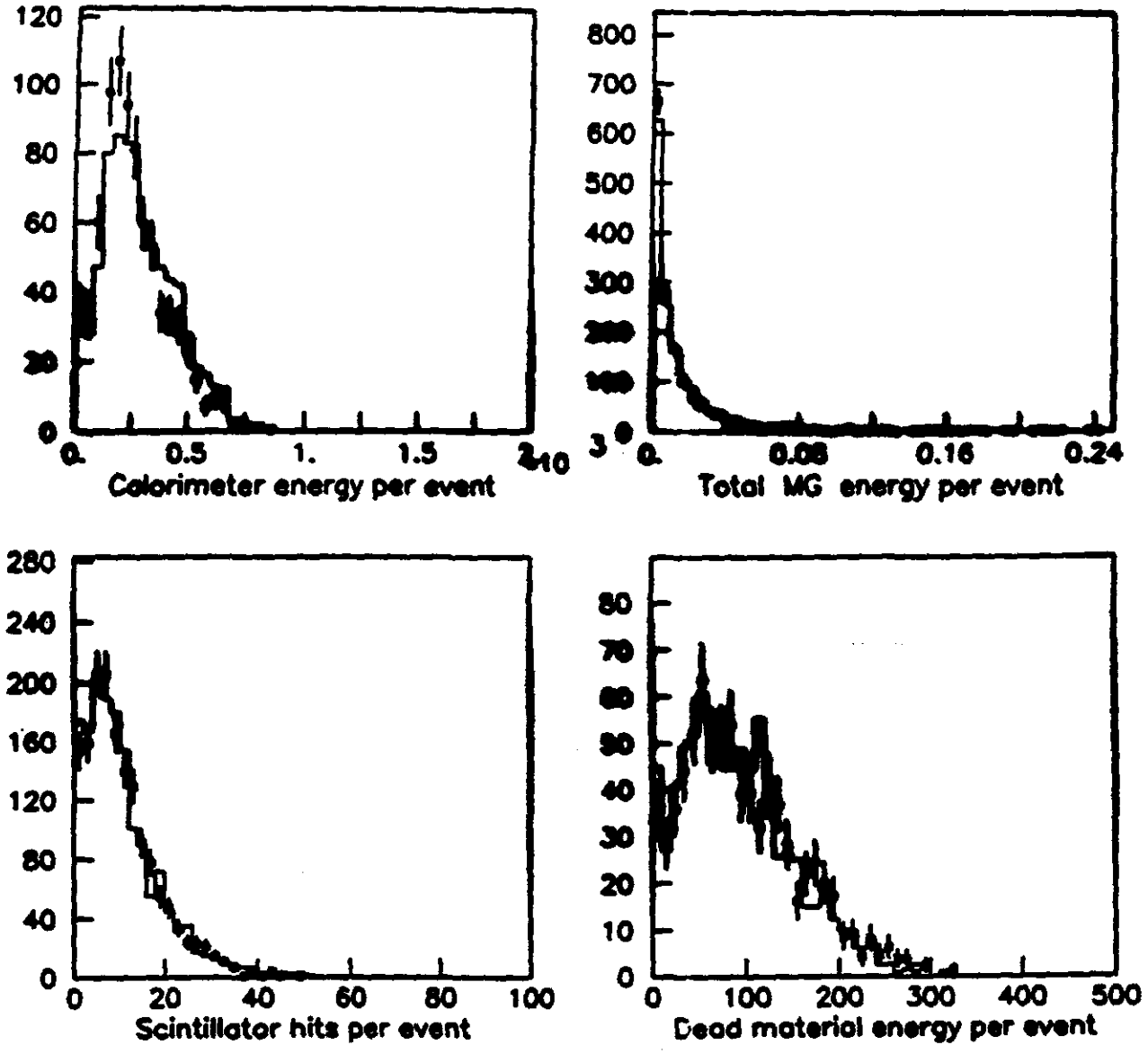

ID 20320
ENTRIES 1917
MEAN 30.67
RMS 22.01
200
175
150
125
100
75
50
25
0
0
20
40
60
80
100
120
140
160
180
JETS – Et of JETS


ID 320
ENTRIES 1890
MEAN 30.64
RMS 21.45
200
175
150
125
100
75
50
25
0
0
20
40
60
80
100
120
140
160
180
JETS – Et of JETS

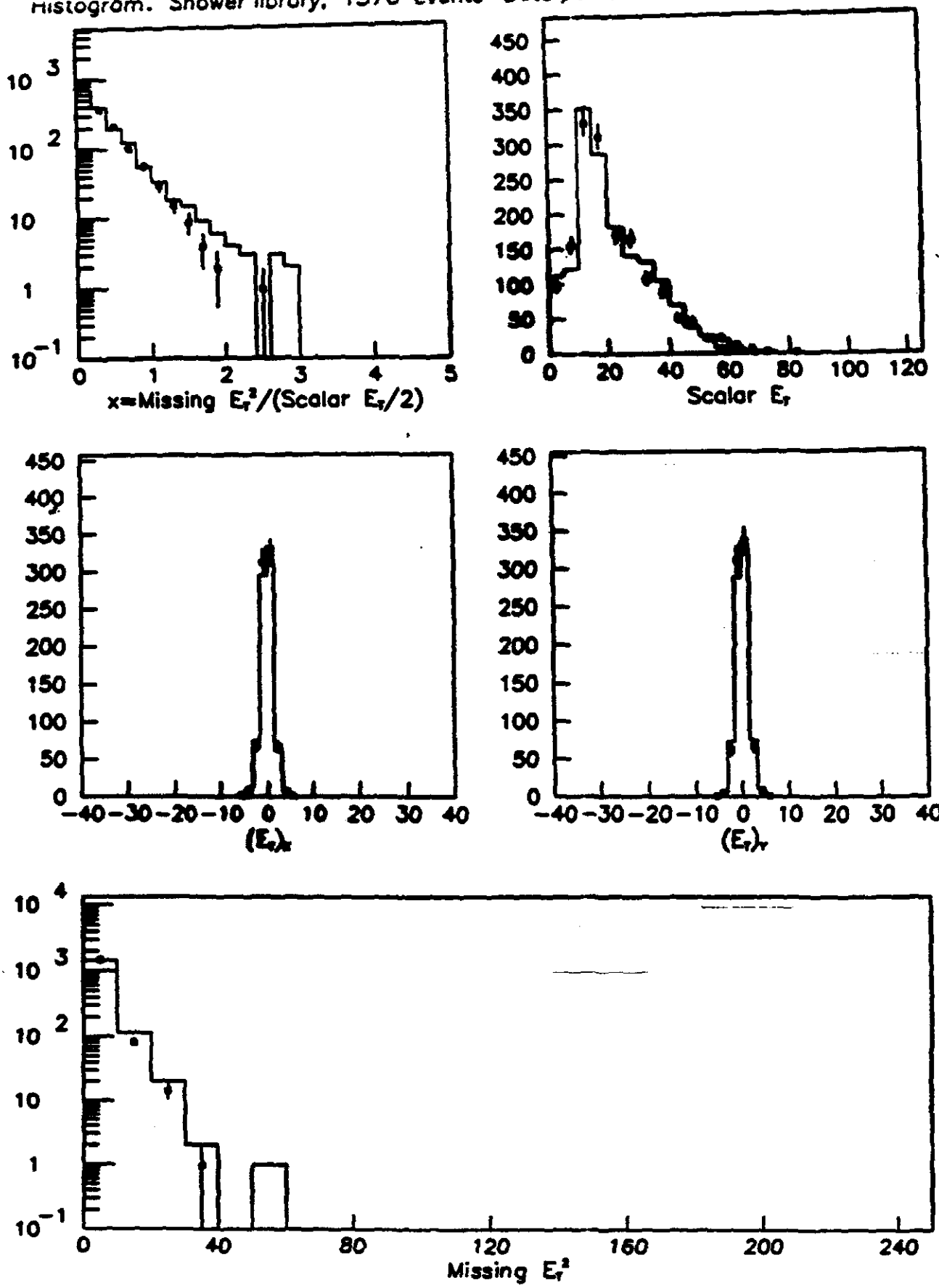
DOGEANT Minimum bias events



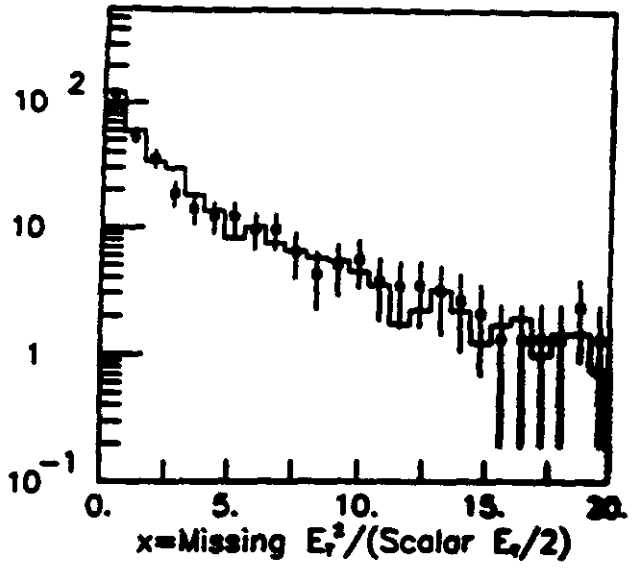
Histogram: Shower library, 1576 events Data points: Full simulation, 1645 events

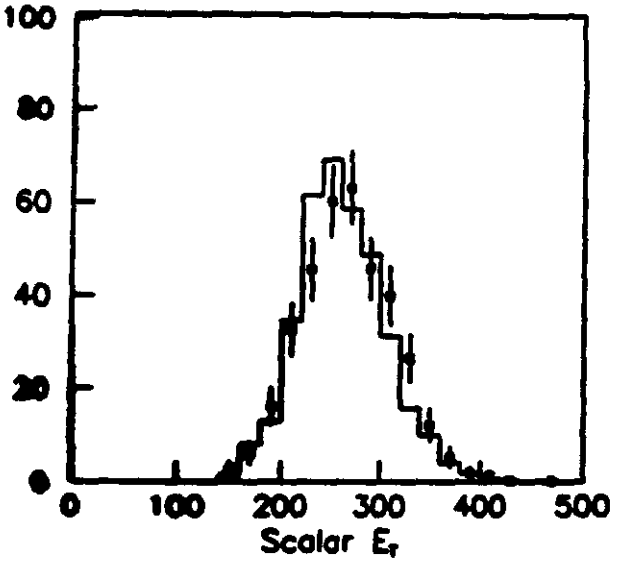


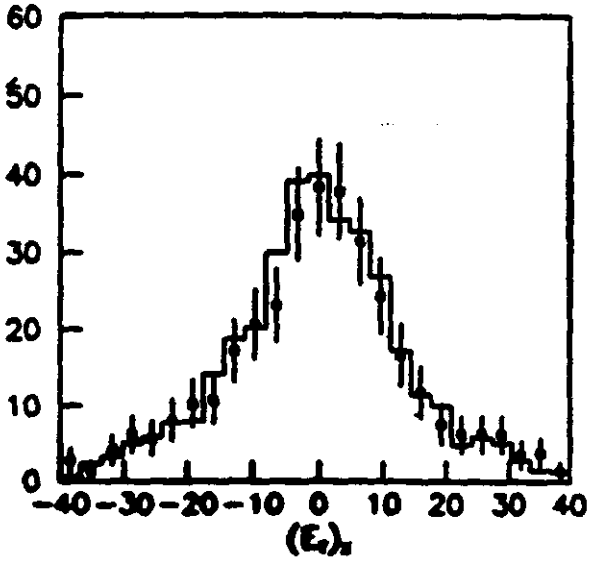
Histogram. Shower library, 1576 events Data points: Full simulation, 1645 events

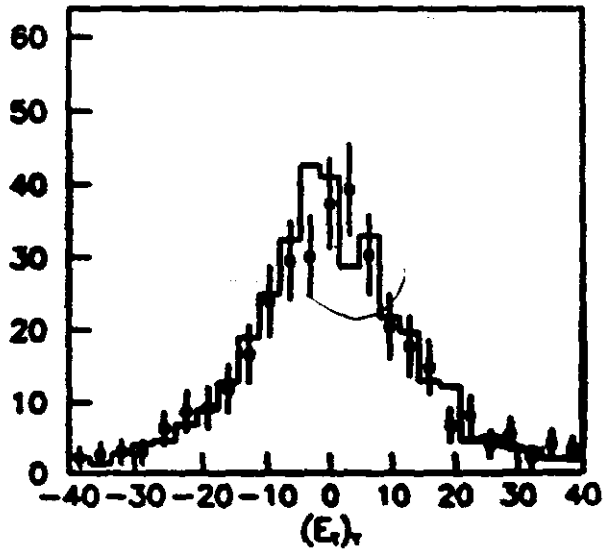


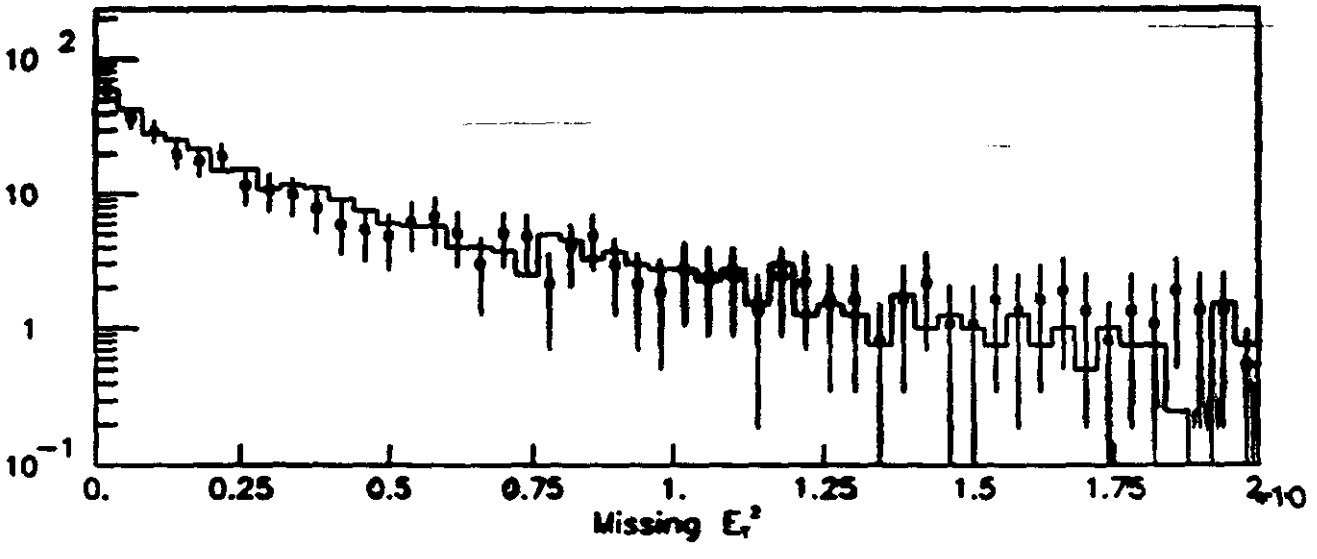
Histogram: Shower library, 1438 events Data points: Full simulation, 1338 events





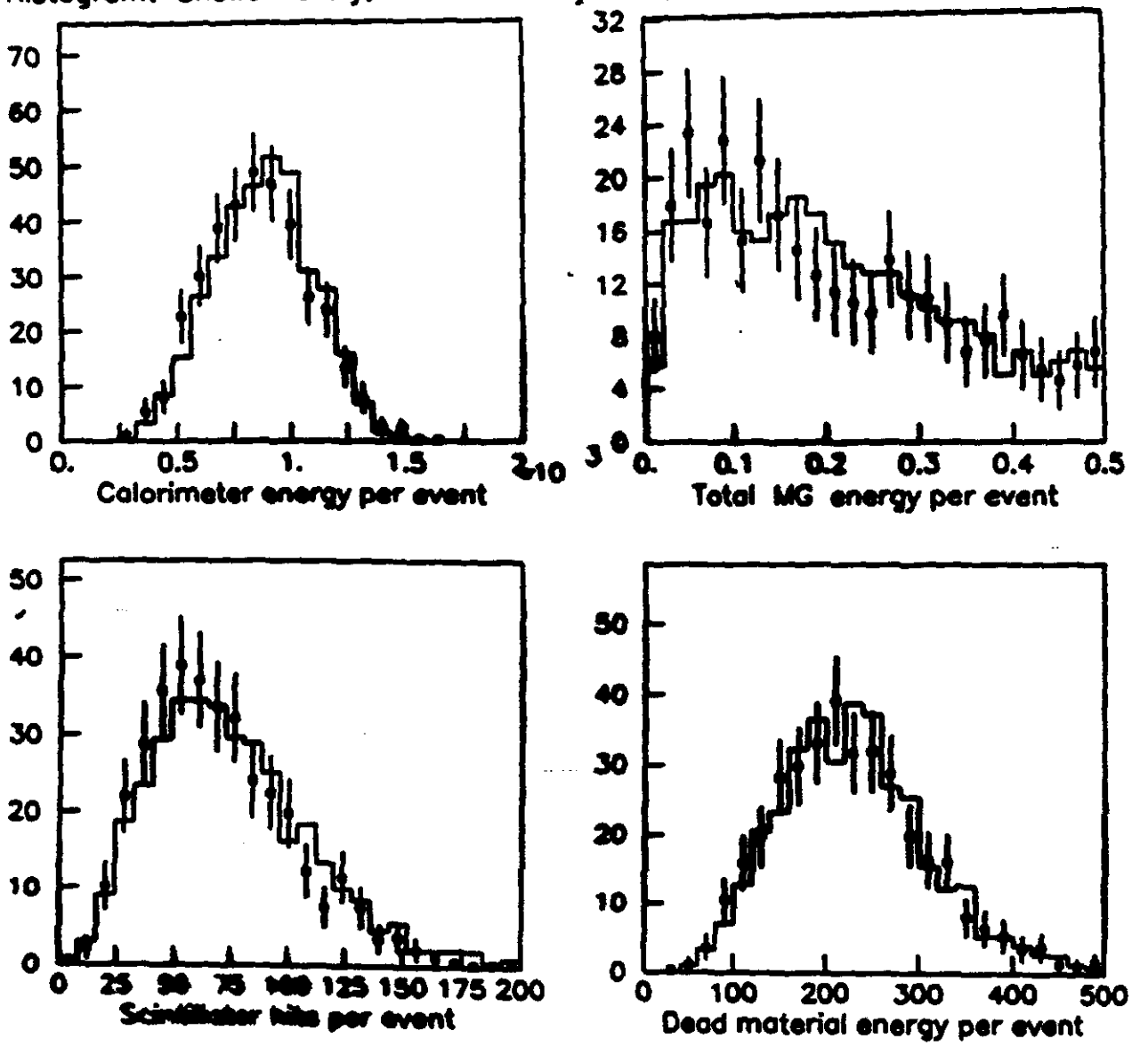






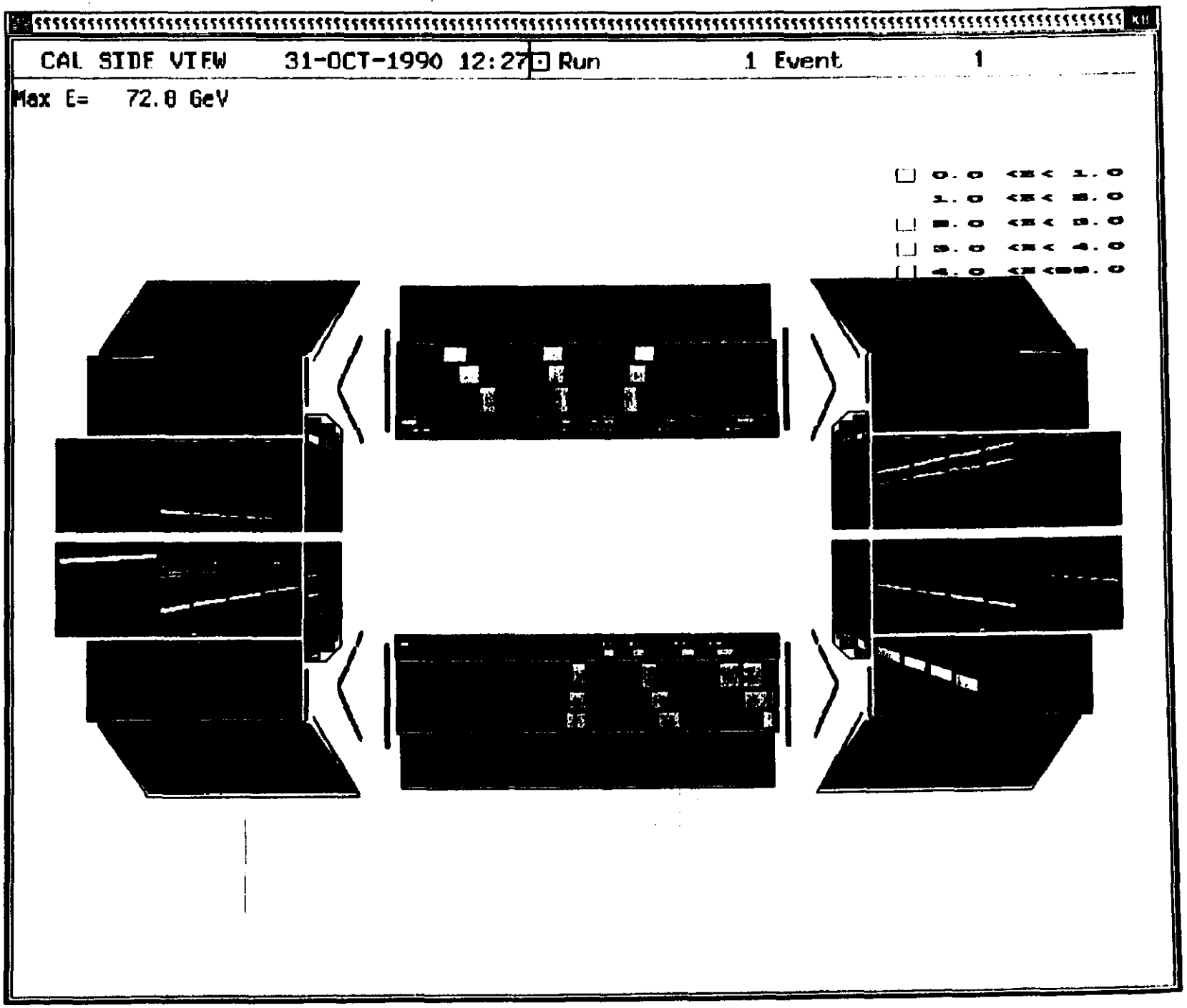
DUGEANT Two jet events, p_T 120 ...

Histogram: Shower library, 1438 events Data points: Full simulation, 1338 events



86-766-9

[redacted]


CAL SIDE VIEW 31-OCT-1990 12:27 Run 1 Event 1
Max E= 72.8 GeV

OFFLINE EVENT RECONSTRUCTION

Manpower organization

Code management (D0 Library)

Tools and rules for code development

Program Builder

Status of reconstruction program (D0RECO)

Computing needs

What remains to be done.

The DO software library consists of two major directories:

DOCMS is the repository of CMS sources with a history of all the changes. There is only one copy of this directory which resides at the BNL VAX cluster.

DOLIBRARY contains all the fortran sources,object libraries, documentation, etc. Identical copies of this directory are kept on all VAX's in the collaboration.

Each of the main directories are logically subdivided into subdirectories which groups modules according to their function.

The files in DOLIBRARY are derived from files in DOCMS using MMS.

Logical symbols are defined to make environment for the users identical on any VAX.

DOCMS

Subdirectories for each major utility library
and applications program.

A CMS subdirectory contains elements bundled in groups.
One element = one module = one subroutine (by decree)
Any element can be in any number of groups.

Element name is unique.
Module name is the same as the subroutine name.
If more than one version of a subroutine is
needed a prefix is added to the module name.
Multiple versions needed for interfaces and to
handle machine dependent code.

Documentation is also maintained with CMS.
Applications programs and some utility libraries
have manuals in their respective subdirectory.

LOCAL_CMS

Test_***

y-

For a simple utility or self contained library subdirectories are connected via MMS procedures in the following way:

[DOCMS.XXX]

generates

fortran sources in
[DOLIBRARY.TEST.XXX.SOURCE]

[DOLIBRARY.XXX.SOURCE]

generate

object library XXX.OLB
[DOLIBRARY.TEST.XXX]

[DOLIBRARY.XXX]

RULES FOR RECONSTRUCTION PROGRAM

Follow standards for offline code (detailed document):
All code FORTRAN 77 with minor exceptions
Special rules for machine dependent code

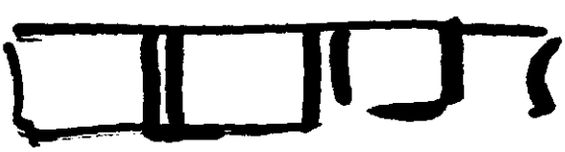
Tools have been developed for checking and enforcing standards (special editor EVEDT, checker DOFLAVOR) prior to code being distributed.

New tool CMSINTERFACE to enforce standards before code is inserted in the CMS library.

Program designed as built of a series of independent tasks. Tasks communicate only via Zebra banks and a limited set of interface subroutines.

Task control parameters must be supplied via RCP files

Program builder combines the tasks into a predefined framework by building the user hooks as a set of calls to the interfaces. The program builder generates code and the link file.



EZGET (!ARCHIVE, VER, ...
ARLOT 3E-0 !

TOOLS FOR CODE DEVELOPMENT

EVEDT

D0 editor, provides templates for headers, Zebra bank documentation,etc. Finds entry points in library, has language sensitive features.

OFLAVOR

Checks that offline standards are satisfied, handles machine dependent features.

OENTRY

Find modules in library.

BANK

Combines Zebra bank documentation with debugger.

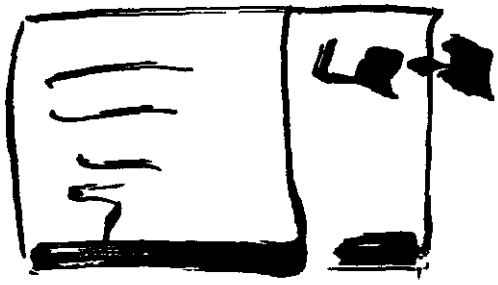
BANK

Generates templates for all subroutines that must be created for a ZEBRA bank.

MSINTERFACE

Checks for code standards and module name uniqueness before insertion into CMS library.

CBIF S1-GRO



CBEW

DATA MANAGEMENT

Event data and constants (calibration, geometry,etc.)
handled with CERN data structure management system ZEBRA.

All data in banks, groups of data connected by pointers (links).
Basic bank structure:

links

identifiers

data

Banks are defined before any coding.
Every bank must have utilities and documentation.

A tool for assembling programs
and enforcing discipline.

Programs consist of a frame, utilities
and applications. Frame and applications
may call utilities but not the reverse.

An application is inserted in a frame via
a set of interface subroutines (package).

Applications communicate with other applications
only through Zebra banks or a very limited set
of subroutine calls.

Requirements for using PBD:

1) A framework program

2) Library and/or user packages

3) A Framework description

4) Package descriptions

5) a list of library packages

6) a link file template (.LNK)

7) a setup command file for the framework

PROGRAM BUILDER FRAMEWORKS:

DOUSER

for interactive analysis and code development. Uses COMPACK.

CALOR_OFF

calorimeter offline code development (no COMPACK, not interactive)

DOGEANT

DO montecarlo using GEANT

EXAMINE2

online framework, interactive analysis of data in shared common. Uses COMPACK.

DORECO

Offline reconstruction framework. Strictly batch for production.

PRODUCTION PASS DEFINITION

Well defined program (.EXE) + unique set of parameters.

Unique ID stamped on HSTR bank generated during production pass.

ID= one composite integer
= number alias for program name
+ production pass number

ID's are keys to entries in data base consisting of:

1) Production pass number
2) Production program name
3) Production program version number
4) Date & time of production pass definition.

TIMING STUDIES

Production version (tracking by road)

double blind

45 sec/event/MIP

W +jets

54 sec/event/MIP

t t

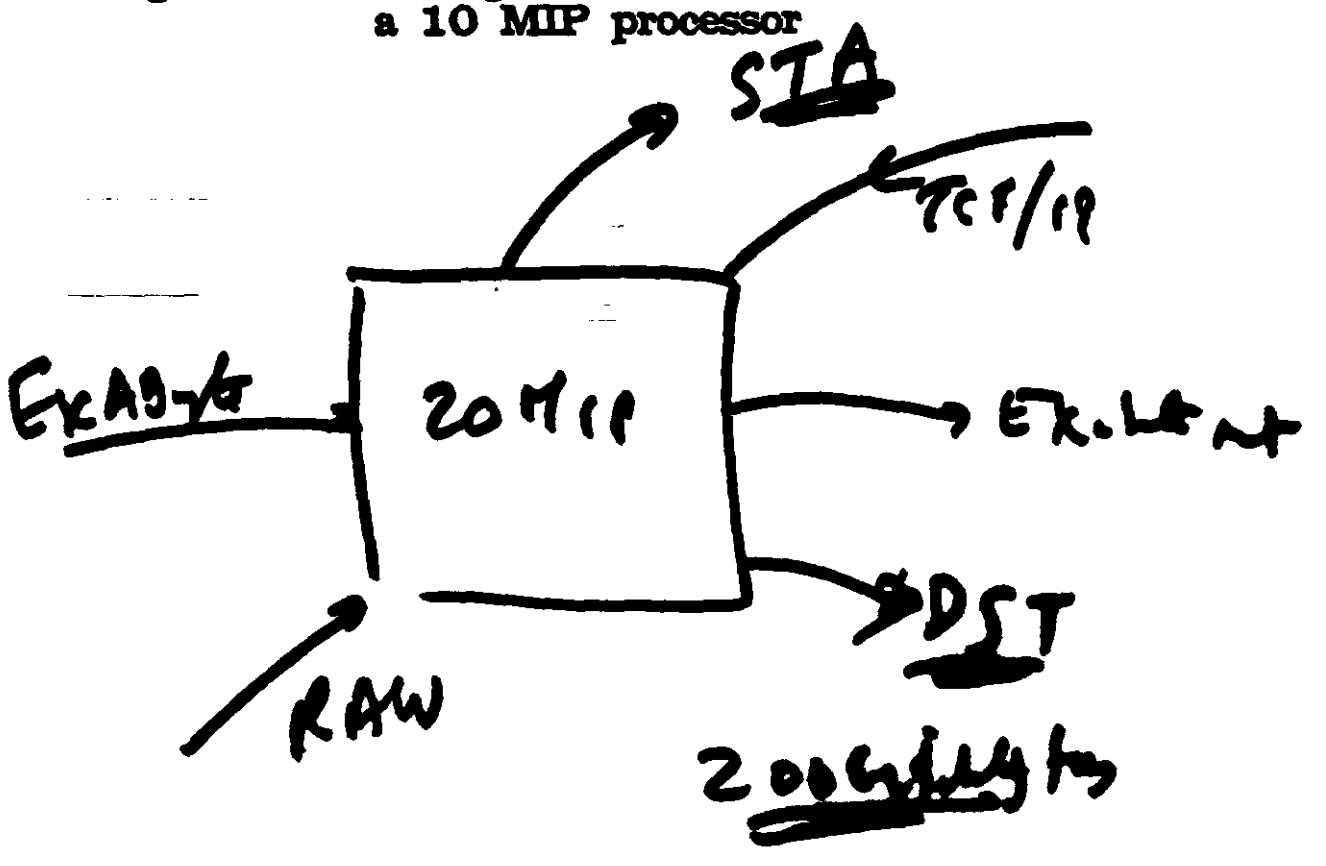
75 sec/event/MIP

Accurate vertex (use all CDC tracks)
increases time by 33%

Full tracking increases time
by factor of 3

Reading event from 8mm tape = 6 sec.

⇒ processing time = reading time in
a 10 MIP processor



Improvements to reconstruction program:
multi-vertex handling
better electron ID (use TRD)
improve tracking efficiency
improve calorimeter resolution

Setup a production manager
(programs and manpower)

Setup a data base to keep track of data processing

Establish priorities on data processing
identify what goes on the express line.

Define procedures for handling the express line
identify special manpower

Define filtering programs to select data for analysis

Setup procedures for distributing DSTs
outside FNAL

Automatic bookkeeping system for data files
(FATMEN?)

200Gb capacity file server to serve data summary files from the complete data set to analysis workstations.

1Tbyte robotic mass storage mounting of 8mm tape.

A facility for tape stripping, copying and data set editting.

A modest amount of compute capacity for jobs which can not be practically run on local workstations.

FDDI to ethernet bridges to connect server to backbone.

Server Details:

200Gbyte Store:

Dual hosted VAXserver 4000 boot nodes for the 200GB disk farm LAVC. These nodes are to be equipped with 64Mb memeory each, 5 GB local dual hosted disk and 8mm tapes for backup.
M76 and VCS softawre to act as console system.

Twelve VAXstation 3100 M76 server nodes with ~17 GB SCSI disk of the the highest capacity currently available on each node. Use of 2 GB drives implies eight drives on each node.
Each node equipped with 32Mb memory and local paging disk.

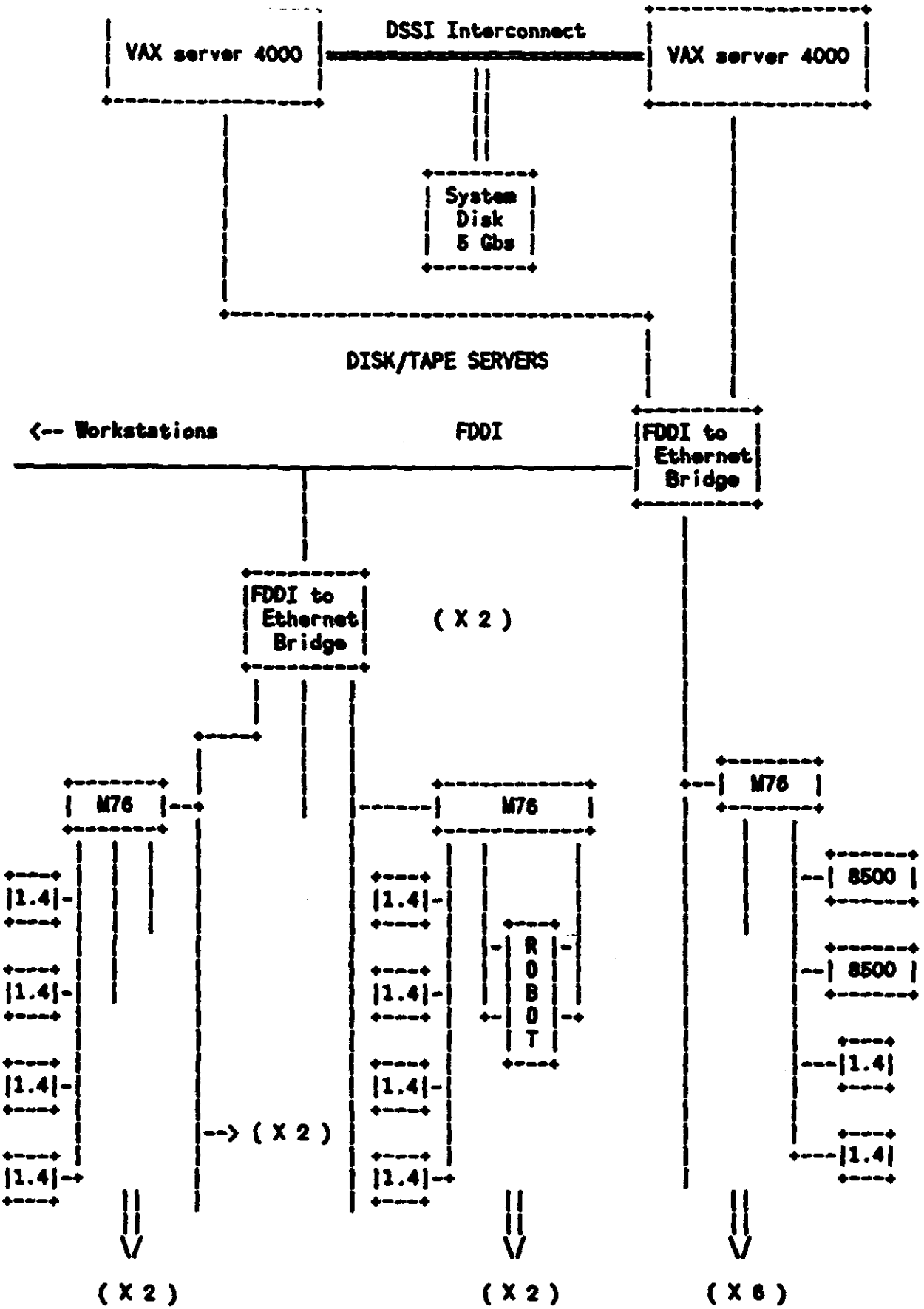
Tape stripping/filtering:

Twelve VAXstation 3100 M76 server nodes, each with two 8mm double density drive and four SCSI disks for local staging and storage.

Robotic Storage:

Two Exabyte 8mm robots with 4 EXB 8500 tape subsystems. Each robot is to be connected to a pair of VS3100/M76 nodes, one providing control of the robot via a SCSI bus. CMD software used to control robot. Total capcity is ~1Tbyte.

Full system has ~204 MIPS available, 16MIPS on boot nodes, 96 MIPS on tape subsystem, and 96 MIPS on disk server nodes.


VAX server 4000
DSSI Interconnect
VAX server 4000
System Disk 5 Gbs
DISK/TAPE SERVERS
<-- Workstations
FDDI
FDDI to Ethernet Bridge
FDDI to Ethernet Bridge
(X 2)
M76
M76
M76
1.4
1.4
1.4
1.4
--> (X 2)
1.4
1.4
1.4
1.4
R O B O T
8500
8500
1.4
1.4
(X 2)
(X 2)
(X 6)

"The SLD Experience"

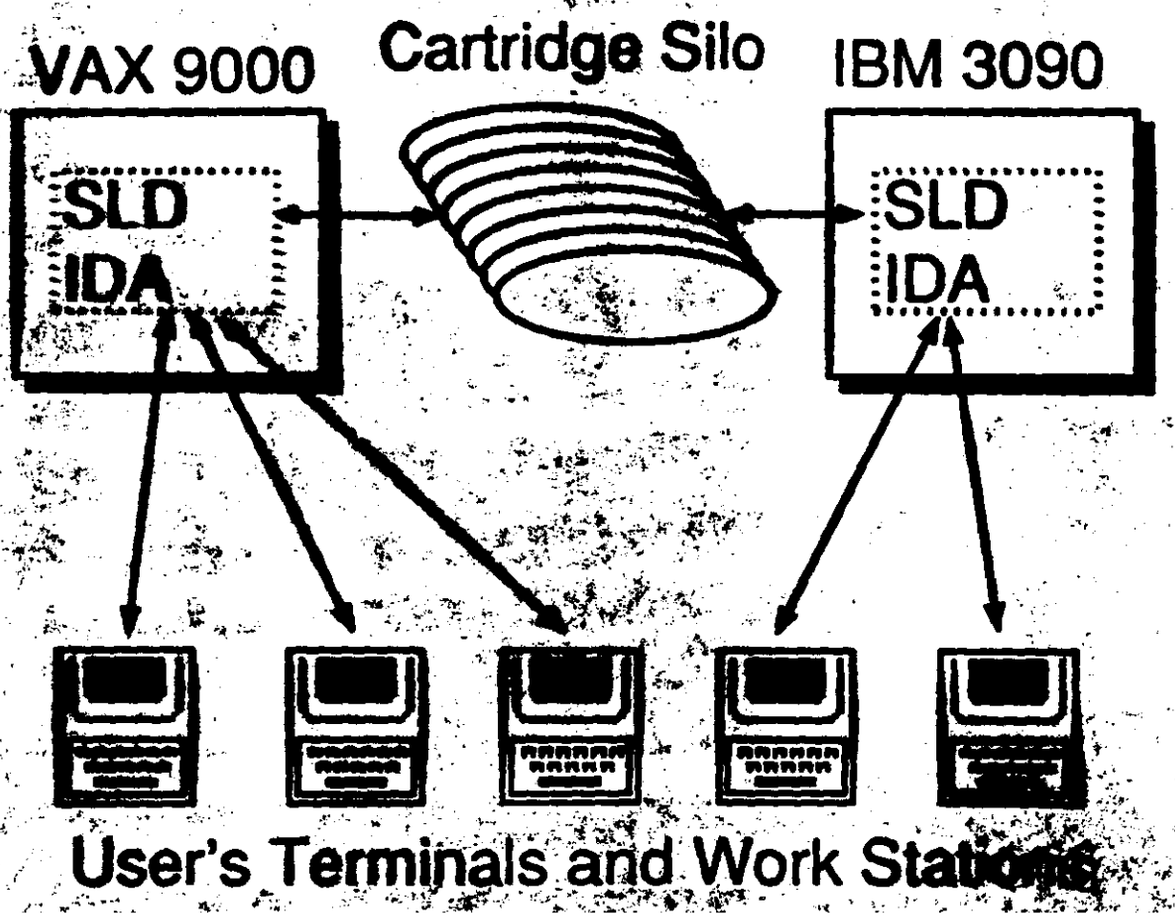
Tony Johnson
Boston University

or

What can be done if you have 8 years before getting any data.

Interactive Data Analysis

Building on previous SLAC experience SLD has placed a heavy emphasis on providing physicists with specially developed tools to allow rapid development of physics analysis code.



THE SLD CURENTLY USES:

- A VAX 8800 running VMS as the acquisition computer for user interface and data logging. Attached to this machine are ~ 40 VAX workstations providing window and SCP support.

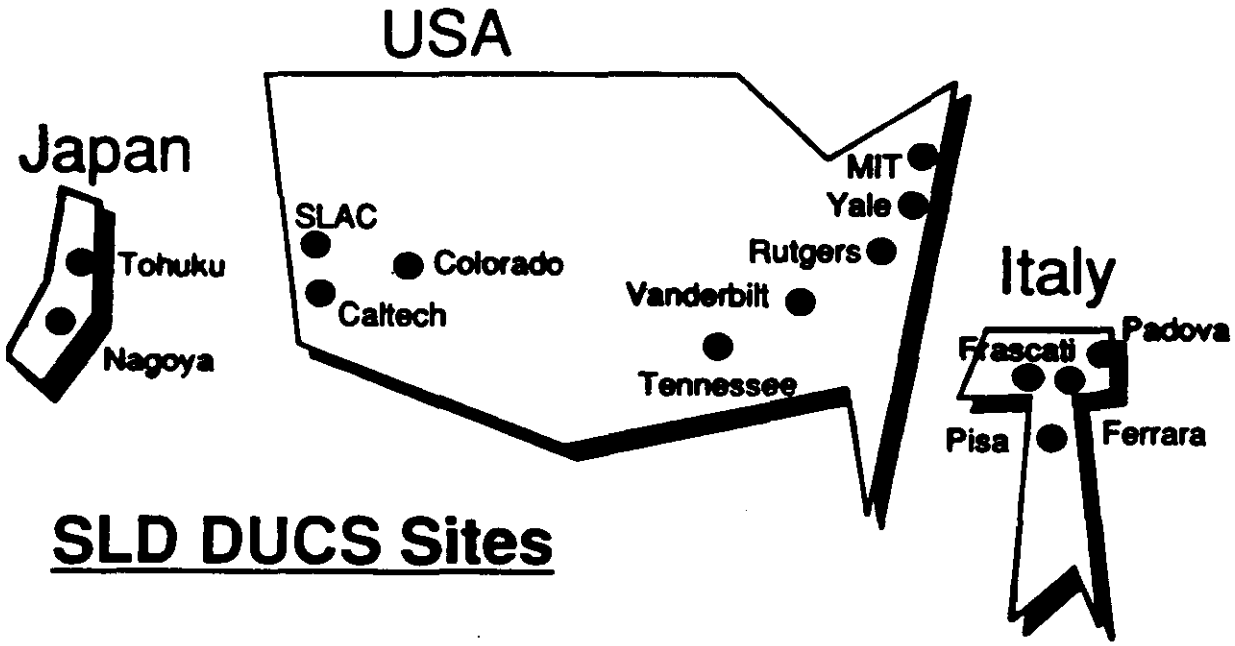
- The current IBM/VM system for production Monte Carlo and reconstruction. This system is the principle control point for the 3480 tape silos.

- A newly acquired VAX 9000 (roughly 30 Mips) running VMS. This machine is used for the bulk of the off-line code development and debugging. We have also extended SLD's IDA system to utilize the windowing system (DEC windows/ X windows) to enhance our analysis capabilities. It is also attached to the tape silo. The SLD code writes IBM format tapes, independent of the system it executes from, to allow this bimodal access.

- Over 1000 Mips of CPU power in the Fastbus acquisition system.

SLD's Use of SLAC Computer Facilities.

SLD has made extensive use of SLAC's network connections to create an environment in which remote collaborators can contribute to developing, testing and using analysis sofware.

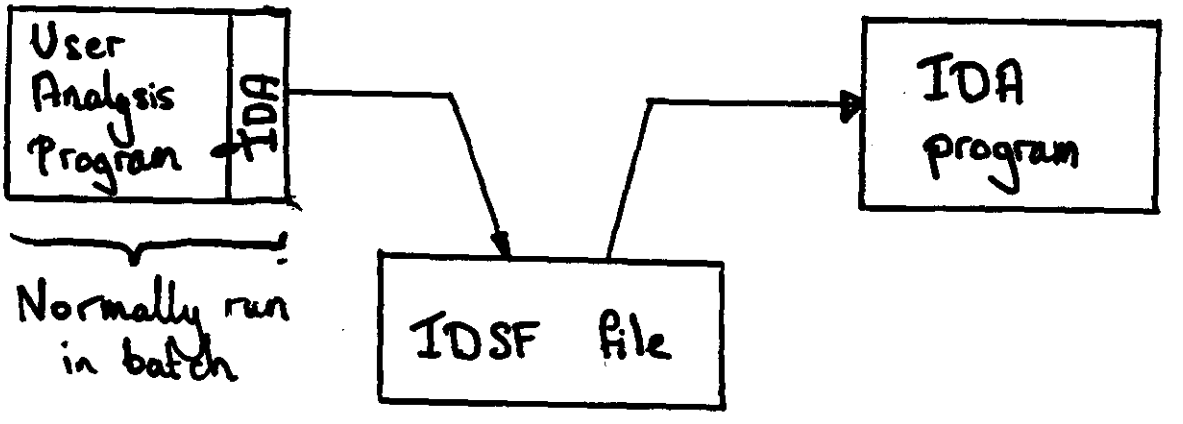


SLD DUCS Sites

Updates to code, documentation, or discussion conferences made by any SLD collaborator at a DUCS site is almost instantaneously relayed to his colleagues around the world.

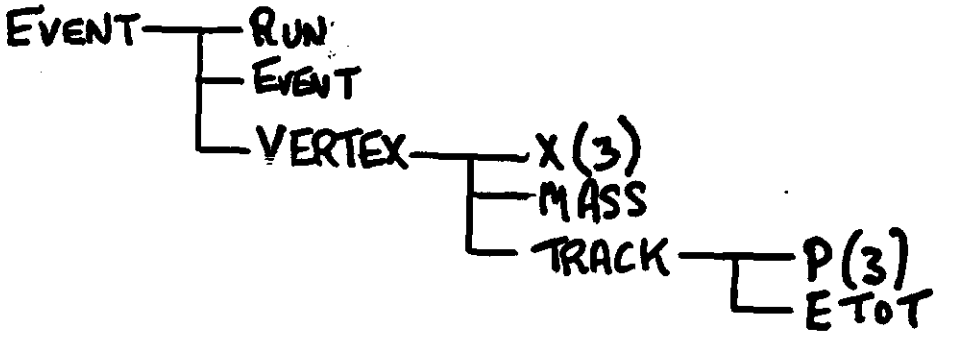
Interactive Data Analysis @ SLAC

- IDA program developed by Toby Burnett for use by MARK 3 experiment at SPEAR



IDSF file contains events each of which may contain nested structures of arbitrarily named real variables

Example IDSF structure



Mark 3 IDA allows:

- Creation of histograms/scatter plots of single or multiple quatities.

  HIST ETOT from 0 to 10

  PLOT MASS vs RUN from 0 to 1000

  SCAT X(1) vs X(2)

- Application of cuts and arithmetic operations

  CUT SQRT(X(1)**2 + X(2)**2 + X(3)**2) > 1

- Manipulation of histograms/fitting etc.
- Output of plots

→ Used extensively by MARK 3 and several other SLAC experiments

→ Very successful → claimed to improve "analysis efficiency" by a factor of 10.

Limitations of MARK 3 IDA

- Only Suitable for Final DST analysis
- Have to frequently remake IDSF file
- Limited ability to call user written code

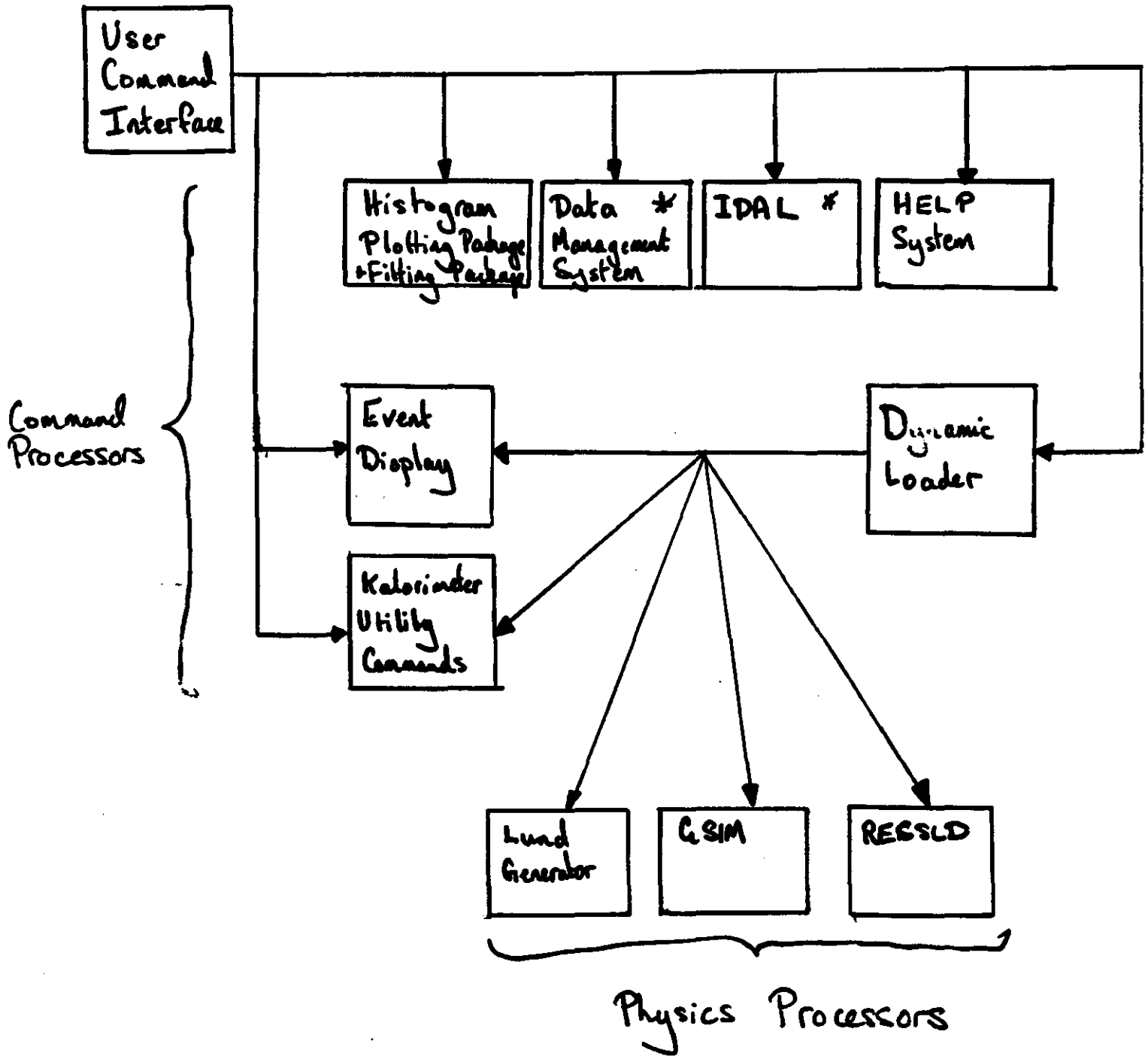
SLD's Goals

- Wanted to place an emphasis on interactive tools for analysis & for developing & testing MC / simulation / recon.
- Wanted a uniform approach so that same tools were available throughout the offline environment
- Immediate interactive access to all data with no intermediate file.
- Modular code design — easily extensible
- Include fully interactive event display
- Analysis environment should be fully/easily customizable by user.
- "Interactive" facilities must also be available in batch

→ Maximize physicist productivity by providing a powerful, user-friendly analysis environment.

The Result

One program (SLDIDA) which is the shell for all offline analysis programs:



IDAL ≡ Physics Analysis Language

Simple semi-interpretive language optimised for rapid development of physics analysis programs.

Standard Programing Statements:

DEF routine (args) ; ENDDEF

IF ; ELSEIF ; ELSE ; ENDIF

DO ; ENDDO

Assignment / Arithmetic Operations.

Invokation of External Programs

CALL routine (args)

A = routine (args)

Histogram/plotting

HIST / PLOT / SCAT

Access to Data Structures

BANK LOOP

TABLE LOOP

POINTER

```
$ IDA
IDA>    Processor    MCLUND
        Processor    GSIM
        Processor    RECSLD

        POKE         MCLUNDP. NUMFLAV  = 1
        Poke         MCLUNDP. FLAVOR   = BQUARK
        Peek         MCLUNDP

        Def   EVANAL

            Call    MCLUND
            Call    GSIM
            Call    RECSLD
            Call    ANALYSIS
            WRITE
        EndDef

        Def   ANALYSIS

            BankLoop  PHPART

                Hist  PHPART%E  from 0 to 10
            End Loop
        EndDef


        OPEN  WRITE  $DAT: MYDATA
        SEGLIST  CREATE  OUTLIST  MCEVENT  DST


        IF  BATCH

            GO  100
            HOUT  ALL

        ENDIF
```

Why was JAZELLE created?

JAZELLE = SLD Data Manager

Jazelle was created for the SLD collaboration to meet their requirements for a powerful, user-friendly data management system suitable for use in an environment strongly emphasizing interactive computing.

- Commercial database programs are available but in general they are poorly suited to the range of different sizes and types of HEP data, and do not provide sufficiently efficient and easy to use program interfaces.
- Many languages have powerful data structure handling capabilities, but they do not in general have good facilities for moving data between program or for providing easy interactive access to the data.
- Other HEP data management systems fall short in the area of user friendliness, requiring users to memorize numeric offsets within structures, and providing few tools for interactive data analysis.

Main features of JAZELLE

- The structure of each bank is defined in a *Template* file. Each element within the bank is named and type cast.
- Multiple computer language interfaces exist. Data is always accessed by name.
- Many facilities allow users intuitive access to data.
- Interactive access to data from debuggers and other programs supported.
- Hiearchical and relational data structures are supported.
- Virtual Memory services of the host operating system are used for bank allocation and deallocation.
- Machine independent IO to both sequential and indexed files.
- Asynchronous IO for inter-process and network communication.

Templates

The structure of each Jazelle bank is defined in a user defined template file. In addition to defining the structure of the bank the template allows:

- Each element to be named and typed (Real, Integer etc.).
- A description to be attached to each element (the template therefore fulfills an important documentation function).
- Initial values to be attached to elements.
- Dimensions to be attached to elements.
- Elements to be grouped together into blocks.
- Additional formatting information to be attached to elements.
- Elements of different types to be freely mixed within banks.

SLD use of Jazelle

SLD has used Jazelle for all data structures from on-line data acquisition to off-line analysis. Jazelle has virtually eliminated the use of common blocks in SLD code. Jazelle is used for:

- RAWDATA, RECON, MC and DST data
- Analysis "micro" DSTs
- Geometric description of detector
- All calibration and run dependent constants.
- Particle properties and decay modes
- Inter-routine communication
- Program control structures
- Parameters for MC generators and reconstruction programs.

Storing control and parameter information in Jazelle banks is particularly useful since it provides a uniform way for physicists to examine and modify these parameters.

Example Template

Bank declaration specifies family name, context and title.

```
BANK NONSENSE CONTEXT=JUNK NOMAXID "Title"

      INTEGER    A "Description of A"
      REAL       C
      INTEGER    NELEMS
      REAL*8     D(10)
      LOGICAL*1  X(-1:7)
      INTEGER    Y/0/
      PARAMETER  SIZE=8
      INTEGER    N(SIZE)/SIZE*7/

      BLOCK      VBLK(NELEMS)

         STRING   F(80)
         INTEGER  I
         BLOCK INNER(2)
            REAL X
            REAL Y
         ENDBLOCK

      ENDBLOCK

ENDBANK
```

Simple scalar elements.

Vector elements.

Declaration and use of parameter.

Initial values.

Variably dimensioned block created by use of previously declared integer element as dimension.

Nested blocks.

Element Types

All types may be used as scalars or vectors (fixed or variably dimensioned). Types may be freely mixed within banks.

- INTEGER [*4|*2]
- HEX [*4|*2]
- REAL [*4|*8]
- LOGICAL [*4|*1]
- STRING [*1|*4|*8]
- COMPLEX
- POINTER
- KEY
- ENUM [*4|*2]
- BITS [*4|*2]

Pointer and Key types are used to relate banks (see later).

Enumerated and Bits types are used to provide mnemonic access to common HEP types (see example).

[] indicates optional size declaration (in bytes).

In addition to the built-in types listed here Jazelle also supports user-defined types.

Real Template

```
Bank MCPART Context=MCEVENT Maxid=1000 "MC Particle parameters"

  Real    P(3)        "X,Y,Z momentum of particle at track origin"
  Real    E           "Energy of particle"
  Real    PTOT        "Total momentum at track origin"

  Partid  PTYPE       "Particle type"
  Real    CHARGE      "Charge of particle"

  Bits    ORIGIN      "Where did this particle come from/go to" ! Note 1
  Real    XT(3)       "X,Y,Z of termination"

  Key     PARENT-->MCPART "Key giving parent particle"
!
! Notes:
!
!  1) The following bits may be set in ORIGIN (to be extended).
!
!                              Meaning of bit if set
  Value ORIGIN
   Parameter DECAYED =0   ! Decayed by generator
   Parameter DECAYFLT=1   ! Decayed in flight by swimmer
   Parameter BEAMPIPE=2   ! Does not hit detector ( down beam pipe
   Parameter NOINTER =3   ! Traverses detector w/o interacting
   Parameter STOPPED =4   ! Energy below cut; other bits may say why
   Parameter INTERACT=5   ! Interacted, no further decision to be made
   Parameter INTSHDEP=6   ! Interacted, further decision depends on
                          ! selection of shower deposition
   Parameter PRIMARY =8   ! e+e- primary interaction vertex
   Parameter ISTOP1  =9   ! GEANT ISTOP=1, KCASE ne DCAY or
                          !   HADR or (PAIR&Gamma)
                          !
   Parameter NOTTRACK=16  ! Not tracked by user request (neutrino, etc.)
   Parameter ISR     =17  ! Initial state radiation
   Parameter BEAM    =18  ! Initial state beam particle (*)
   Parameter PREFRAG =19  ! Pre-fragmentation particles (*)
                          !   (gluons, partons etc.)
  EndValue
!
!     (*) Particles in these categories will not normally be stored in
!         the bank, but may be put here at the discretion of the
!         generator, maybe under control of some flag.
!
EndBank
```

User defined type

Real Template

```
Bank MCPART Context=MCEVENT Maxid=1000 "MC Particle parameters"

  Real    P(3)        "X,Y,Z momentum of particle at track origin"
  Real    E           "Energy of particle"
  Real    PTOT        "Total momentum at track origin"

  Partid  PTYPE       "Particle type"
  Real    CHARGE      "Charge of particle"

  Bits    ORIGIN      "Where did this particle come from/go to" ! Note 1
  Real    XT(3)       "X,Y,Z of termination"

  Key     PARENT-->MCPART "Key giving parent particle"
!
! Notes:
!
!  1) The following bits may be set in ORIGIN (to be extended).
!
!                              Meaning of bit if set
  Value ORIGIN
   Parameter DECAYED =0   ! Decayed by generator
   Parameter DECAYFLT=1   ! Decayed in flight by swimmer
   Parameter BEAMPIPE=2   ! Does not hit detector ( down beam pipe
   Parameter NOINTER =3   ! Traverses detector w/o interacting
   Parameter STOPPED =4   ! Energy below cut; other bits may say why
   Parameter INTERACT=5   ! Interacted, no further decision to be made
   Parameter INTSHDEP=6   ! Interacted, further decision depends on
                          ! selection of shower deposition
   Parameter PRIMARY =8   ! e+e- primary interaction vertex
   Parameter ISTOP1  =9   ! GEANT ISTOP=1, KCASE ne DCAY or
                          !   HADR or (PAIR&Gamma)
                          !
   Parameter NOTTRACK=16  ! Not tracked by user request (neutrino, etc.)
   Parameter ISR     =17  ! Initial state radiation
   Parameter BEAM    =18  ! Initial state beam particle (*)
   Parameter PREFRAG =19  ! Pre-fragmentation particles (*)
                          !   (gluons, partons etc.)
  EndValue
!
!     (*) Particles in these categories will not normally be stored in
!         the bank, but may be put here at the discretion of the
!         generator, maybe under control of some flag.
!
EndBank
```

User defined type

Fortran-77 Interface

We will use the Fortran interface as an example, but all interfaces are similar. The goal has been to make Jazelle fit as a natural extension of each language.

Pointer Declarations

A variable type *POINTER* is defined to allow access to Jazelle data. Pointer declarations can appear anywhere an Integer declaration can be used. Pointers can be:

- Local Variables
- Common block elements
- Real and dummy arguments

Arrays of pointers can also be declared.

Examples:

```
POINTER KAON-->MCPART
POINTER CHILDREN(6)-->MCPART
POINTER BPTR-->NONSENSE
```

Pointer name

Family it will point to

Accessing Data

Once a pointer has been declared and a value assigned to it (by specifying it as the output argument to JZBFND for example) it can be used to access data from the bank.

This is achieved by using expressions of the form:

```
BPTR%(A)
BPTR%(D(5))
BPTR%(X(K-1))
BPTR%(VBLK(3),I)
BPTR%(VBLK(3),INNER(1),X)
```

The Jazelle expression inherits its type from the Jazelle element referenced. Jazelle expressions may be used anywhere a Fortran expression of the same type could be used.

Examples:

```
ENERGY = MCPART%(E)
PHI=ATAN2(MCPART%(P(2)),MCPART%(P(3)))
MCPART%(PTOT) = SQRT(PTSQ)
```

Dumping Banks

Jazelle provides a number of facilities for producing human readable dumps of banks. Considerable attention has been given to producing easy-to-read dumps based on the information given in the bank's template.

Dump of MCPART bank -- Brief

```
Family: MCPART      ID:     15   Template Version: 0.00
Title:  MC Particle parameters

Name      Value         Name      Value         Name      Value
----      -----         ----      -----         ----      -----
P         See below     E         1.2342        PTOT      1.1055
PTYPE     24(ETA)       CHARGE    0.0000        ORIGIN    0(DECAYED)
XT        See below     PARENT    MCPART(2)

P         1:-.54111 +.95788 -.10835
XT        1:0.0000  0.0000  0.0000
```

Jazelle allows different levels of detail to be selected by the user. Below is the same dump at higher level of detail.

Dump of MCPART bank -- Detailed

```
Family: MCPART      ID:     15   Template Version: 0.00
Title:  MC Particle parameters

Name      Value       Offset Type     Comment
----      -----       ------ ----     -------
P         See below        0 R*4      X,Y,Z momentum of particle at origin
E         1.2342          12 R*4      Energy of particle
PTOT      1.1055          16 R*4      Total momentum at track origin
PTYPE     24(ETA)         20 PARTID   Particle type
CHARGE    0.0000          24 R*4      Charge of particle
ORIGIN    0(DECAYED)      28 BITS     Where did this particle come from/go to
XT        See below       32 R*4      X,Y,Z of termination
PARENT    MCPART(2)       44 PTR      Pointer to the data associated with key

P         1:-.54111 +.95788 -.10835
XT        1:0.0000  0.0000  0.0000
```

Note the output from the user-defined type PARTID (SLD defined in this case) and the output of the word of type BITS (ORIGIN).

Jazelle also allows tabular output to be generated. Columns can be added and removed from tables under user control. Multiple table styles can exist for one family.

Tabular Output of MCPART

Family: MCPART Template Version: 0.00
Title: MC Particle parameters

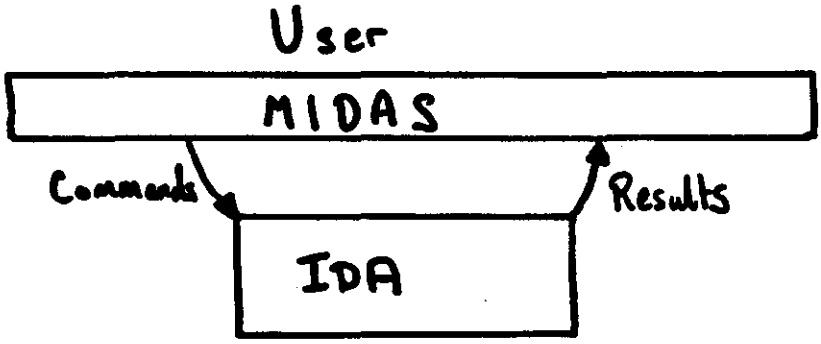
#	Ptype	E	Origin	#	Ptype	E	Origin
1	GAMMA	0.64695	NOINTER	31	PI0	0.16558	DECAYED
2	C	44.128	DECAYED+PREFRAG	32	PI0	0.55768	DECAYED
3	GLUON	1.4228	DECAYED+PREFRAG	33	PI0	0.51094	DECAYED
4	GLUON	2.1941	DECAYED+PREFRAG	34	PI+	0.50439	INTSHDEP
5	GLUON	6.6292	DECAYED+PREFRAG	35	PI-	0.28838	DECAYFLT
6	GLUON	1.6309	DECAYED+PREFRAG	36	PI0	0.16250	DECAYED
7	GLUON	2.6767	DECAYED+PREFRAG	37	PI0	0.39443	DECAYED
8	CBAR	32.671	DECAYED+PREFRAG	38	PI0	0.15733	DECAYED
9	D*-	20.089	DECAYED	39	ETA	0.84457	DECAYED
10	RHO+	4.1097	DECAYED	40	PI+	2.7674	INTSHDEP
11	D*0	22.258	DECAYED	41	PI-	0.69694	INTSHDEP
12	OMEGA	21.030	DECAYED	42	K+	3.3271	INTSHDEP
13	K-	4.7338	INTSHDEP	43	PI-	6.5507	INTSHDEP
14	K0	4.9375	DECAYED	44	PI-	1.7667	INTSHDEP
15	ETA	1.2342	DECAYED	45	PI-	2.5019	DECAYFLT
16	PI+	0.87129	INTSHDEP	46	PI+	2.5375	INTSHDEP
17	OMEGA	0.95527	DECAYED	47	PI-	1.2346	INTSHDEP
18	PI-	3.5826	INTSHDEP	48	PI+	0.79124	INTSHDEP
19	ETAP	1.3963	DECAYED	49	GAMMA	0.67767	INTSHDEP
20	RHO0	3.4643	DECAYED	50	GAMMA	0.70153	INTSHDEP
21	PI+	2.6912	INTSHDEP	51	GAMMA	1.0973	INTSHDEP
22	D-	18.710	DECAYED	52	GAMMA	0.81761	INTERACT
23	PI0	1.3792	DECAYED	53	K0^	16.057	DECAYED
24	PI+	2.1948	INTSHDEP	54	ETA	4.7673	DECAYED
25	PI0	1.9149	DECAYED	55	GAMMA	0.14502	INTSHDEP
26	D0	20.824	DECAYED	56	GAMMA	1.2892	INTSHDEP
27	PI0	1.4342	DECAYED	57	GAMMA	6.2230	INTSHDEP
28	GAMMA	9.7995	INTERACT	58	GAMMA	5.0071	INTSHDEP
29	PI0	11.230	DECAYED	59	GAMMA	0.076595	INTSHDEP
30	K0S	4.9375	DECAYFLT	60	GAMMA	0.088982	INTSHDEP

IDA meets GUI's

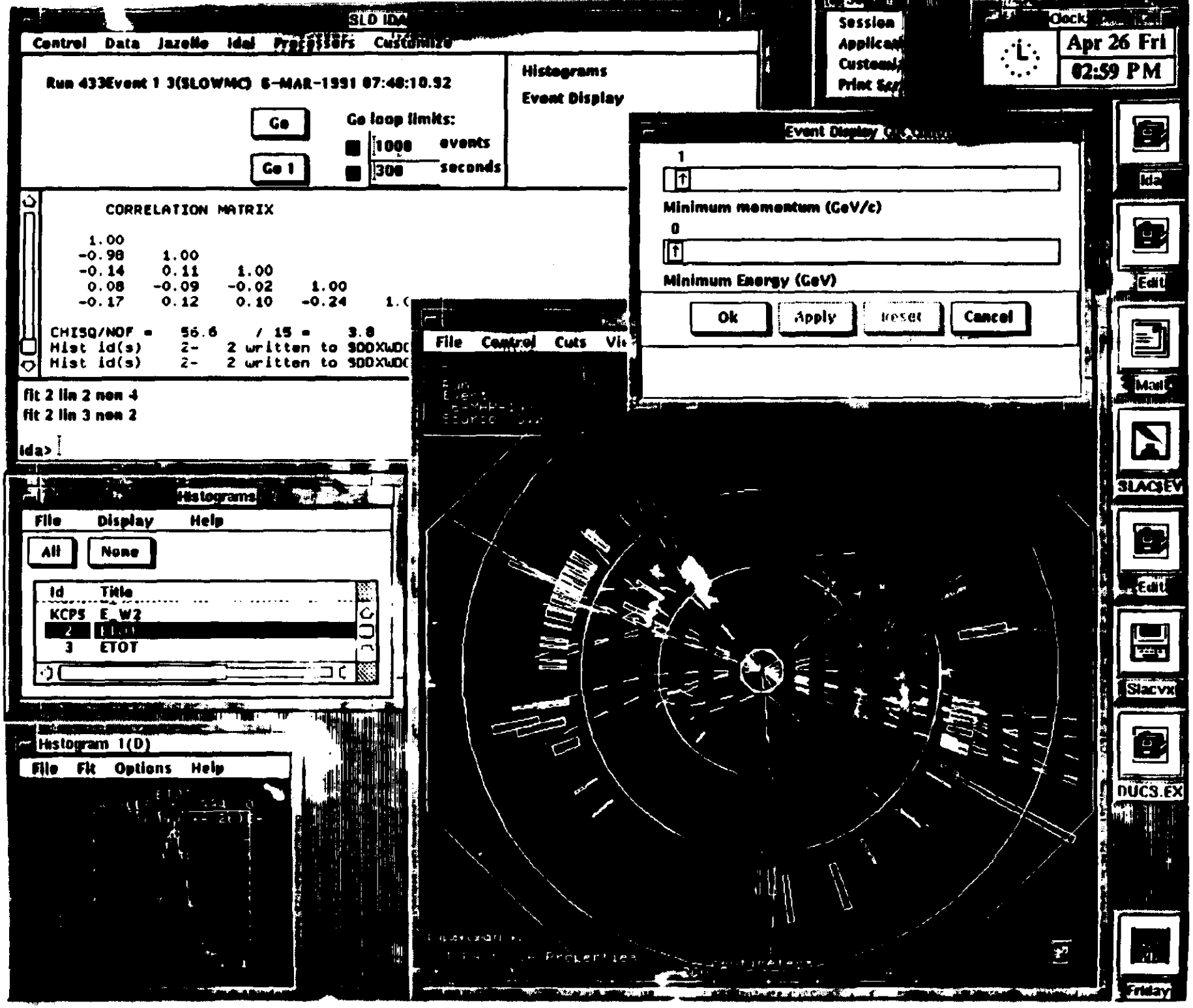
What relevance has a command driven system in the ERA of GUI's ??

Possibly a great deal!

Motif
Interactive
Data
Analysis
Shell



Commands good for:
- Programming
- Batch
- Experts


SLD IDA
Control Data Jazelle Idal Processors Customize
Run 433Event 1 3(SLOWMC) 6-MAR-1991 07:48:10.92
Go
Go 1
Go loop limits:
1000 events
300 seconds
Histograms
Event Display
CORRELATION MATRIX
1.00
-0.98 1.00
-0.14 0.11 1.00
0.08 -0.09 -0.02 1.00
-0.17 0.12 0.10 -0.24
CHISQ/NDF = 56.6 / 15 = 3.8
Hist id(s) 2- 2 written to
Hist id(s) 2- 2 written to
fit 2 lin 2 non 4
fit 2 lin 3 non 2
ida>
Histograms
File Display Help
All
None
Id Title
KCPS E_W2
2
3 ETOT
Histogram 1(D)
File Fit Options Help
File Control Cuts
Event Display
1
Minimum momentum (GeV/c)
0
Minimum Energy (GeV)
Ok
Apply
Reset
Cancel
Session
Apr 26 Fri
02:59 PM
Ida
Edit
Mail
SLACSEV
Edit
Slacvx
DUCS.EX
Friday

Conclusions

- Interactive Data Analysis has been very successful & has much potential to increase ease of learning & using physics recon/analysis packages, especially when combined with GUI.

- Has specific requirements for:

  → Data Manager ← C compatible

  → Program structure

Meta - Conclusions

- ⇒ Programs developed for LOI/EOI/Proposal should not have undue influence on long-term offline programming decissions
- ⇒ Should not ignore C++
- ⇒ Zebra isn't the only data-manager