SLD Data Preservation

2nd Workshop on Data Preservation and Long Term Analysis in HEP

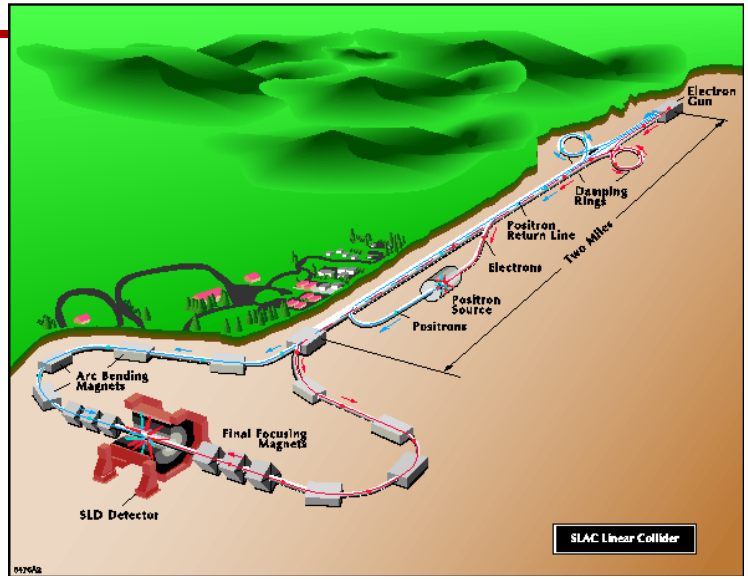
Tony Johnson

26th May 2009

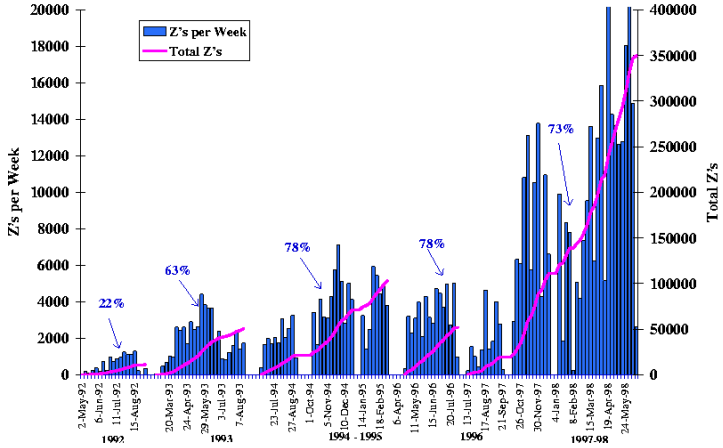
SLAC NATIONAL ACCELERATOR LABORATORY

SLD Experiment

- SLD Experiment ran at the SLAC Linear Collider (SLC) from 1992-1998
  - Produced >500,000 Z's
  - Data sample from world's first (and so far only) linear collider
    - Polarized beams at SLC make the SLD dataset unique
    - Potential use in validating future linear collider simulations
  - Hard to quantify value of data
    - So far we have kept it mostly because we can

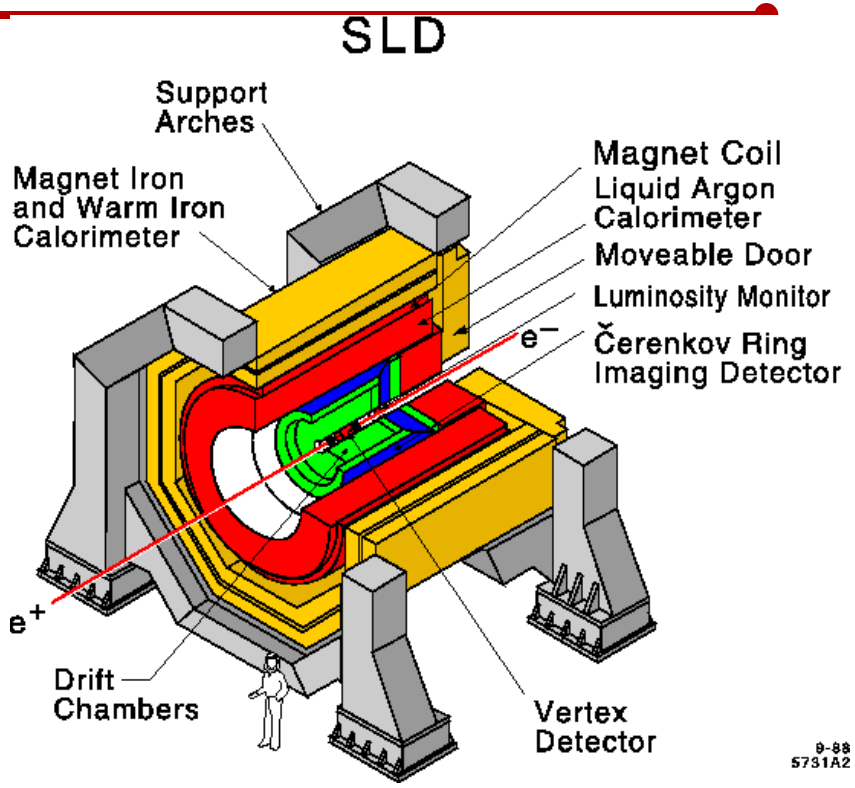


1992 - 1998 SLD Polarized Beam Running



SLD Data Sample

- Reconstruction/Analysis software almost entirely written in Fortran (actually Mortran)
- Designed to be 100% portable
  - runs on “both” operating systems
    - IBM VM
    - DEC VMS
- Raw/Reconstructed data = ~9000 tapes
- Final mini-DST = 500 Gbytes
- Data format – “Jazelle”
  - “Self describing data format but unique to SLD”
- Much analysis code/data left in user’s individual directories and probably lost
- Pre Web! Documentation on paper!



SLD Data Preservation

- Almost no thought given to data preservation until experiment was over
  - One effort to store LEP/SLD data in unified 4-vector format
    - never went far
    - and actively opposed by some
- Tapes were all copied once to next generation tape technology
- New HP VMS server purchased in 2004 to replace original VMS cluster
  - Has 50% of CPU power of final VMS cluster
  - Simulation/reconstruction/analysis software still runs
  - RAID array to contain copy of mini-DST data
  - No longer any easy access to tapes
    - Oracle database describing tape contents still exists
- Very few people left who understand data
- No manpower left to manage system
- Cyber Security?

SLD Data Preservation - Future

- Storing data in the cloud
  - Storing SLD mini-dst on Amazon EC2 cloud would cost <$50/month (probably much less).
  - VMS runs on Itanium, or using emulator under Linux
    - In theory possible to set up virtualized “on-demand” server to maintain OS long-term.
- Reading the data without the software
  - Preserving “bits” is easy
  - Preserving running software is harder
  - Although SLD data is “self-describing” that is not sufficient to read data in absence of software
    - We wrote a standalone <2000 line Java program which can access all objects in mini-dst with no other software required
      - Serves as proof-of-principle that it would be possible (if sufficiently motivated) to read data without any SLD software.

Data Preservation - Observations

- Root format poses (several) unique problems for data preservation
  - I will mention only one – lack of low level documentation on IO format
    - FITS
      - Detailed documentation on binary format, >30 implementation, C, C++, Java, .NET, Python…
    - HDF5
      - Detailed documentation on binary format, 1 implementation, C++, Python, Java interfaces
    - LCIO
      - Detailed documentation on binary format and physics objects, C++, Fortran, Java implementations
  - Root
    - Considerably more sophisticated than formats above
      - No documentation of binary level IO. No supported alternatives to C++ implementation.
      - Even more reason that detailed documentation on binary format is required
    - I don't believe archiving root files alone constitutes data preservation
      - Documentation on binary format is required
        - » Maybe creation of such documentation should be sponsored by this group.
      - Proof-of-principle implementation based on documentation highly desirable

Conclusions

- SLD Data still exists (just)
  - Some mechanism to preserve it long-term would be great
    - Especially if it requires no manpower and no $
- Time to think about long-term data preservation is when experiment is starting
  - Leaving data preservation to end of experiment likely to be too late
    - Archiving data in format used by experiment will result in maintenance problems
    - Moving data to different format unlikely to be successful unless data is used for analysis to validate it
- Current OS/Language choices of HEP likely to look as outdated in 20 years as SLD's choices look today
  - Need to give serious thought to how data can be accessed in future
- Data preservation is hard (public access even harder)
  - But other fields show it is possible when required/funded by funding agency