Title
LISA, AN ON- AND OFF-LINE DATA ANALYZING PROGRAM

Author
Kolb, B.

Publication Date
1983-06-01

Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

Presented at the Conference on Real-Time Computer Application in Nuclear and Particle Physics, Lawrence Berkeley Laboratory, Berkeley, CA, May 15-19, 1983; and to be published in the Proceedings

LISA, AN ON- AND OFF-LINE DATA ANALYZING PROGRAM

B. Kolb

June 1983

Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

LISA, AN ON- AND OFF-LINE DATA ANALYZING PROGRAM

Burkhard Kolb

Gesellschaft fur Schwerionenforschung, Postfach 110541
D-6100 Darmstadt, West Germany

With the installation of the VAX 11/780 at the Bevatron in Berkeley to improve the on- and off-line data analysis for the Plastic Ball and the Streamer Chamber, it was necessary to have adequate software. Before the installation of the VAX the program package MULTI[1] was used on the PDP 11/44, which was coupled through a shared disk to the data-taking PDP 11/50. Now the PDP 11/50 is replaced by the PDP 11/44, which has enough memory to handle the data taking and some on-line analysis. But with an improved data-taking program and the higher bus activity MULTI analyzed just a few percent of the incoming data. The program package LISA, which was developed in Heidelberg and used there for the Crystal Ball experiment, seemed to be a promising alternative to the old MULTI, which was designed for small computers like a PDP 11.

LISA is a FORTRAN program package to evaluate on-line and off-line experimental data of nuclear and particle physics on VAX computers. The design goal of LISA was to provide the user with a flexible, easy to use, safe, and expandable system. The analysis of experiments consists basically of two steps, i.e., reading and sorting events into histograms and the manipulation and graphical or numerical output of the histograms. Accordingly, LISA is divided into a process that deals with events and a large number of processes that define the scope of the analysis or act on the accumulated histograms.

All LISA processes communicate through common blocks, which are mapped as a global section. While the event-loop is continuously executing as a subprocess, all other processes are invoked by a command, execute their code (for instance, display a histogram), and then terminate. For certain tasks the command process sets an event-flag to synchronize execution with the event-loop (for instance, to open a new input or output file). The user has complete control over his terminal and can run whatever program he likes. While the event-loop is accumulating events, the user can develop completely independent new commands. If the user makes a typing mistake or a command process is aborted with an error condition, the event-loop will still work. Even if the event-loop process aborts for any reason, all the accumulated data are still available in the global section and can be saved.

The code for the event-loop is optimized for speed of execution and contains only few modules of straight code. The event-loop checks at the beginning of every event the contents of one variable to determine if the user has given a command that needs synchronization and waits in this case for an event-flag. The event-loop calls sequentially subroutines to read and unpack an event, test conditions, increment spectra, do user-routines, and make scatter plots. Each of the predefined routines uses lists to determine what action to perform. The command processes store the needed information in the lists after checking the input parameters. The code for all the command processes is highly modular and consists of some hundred subroutines contained in a library.

LISA can read events from a disk file or can access the on-line data through a shared disk from the data-taking front end computer. Data on magnetic tape have to be copied to disk with a special program, which checks the consistency of the events and which can select or exclude special event-types. This copy operation has to be done to free the tape drive for other users and allow multiple read operations of the same data from different users.

There is no special command dispatcher or supervisor. All commands are processed as normal DCL-commandlines and invoke an interactive process. Parameters for commands can either be passed on the commandline or are prompted by the command process. Interactive commands define one- and two-dimensional histograms, conditions, queues, and scatter plots. All the spectra can be filled automatically with or without conditions, or the user can supply a special routine with more sophisticated calculations.

For complicated experiments the user has to provide an unpacking routine to pick events from the input buffer and put the information in the one-event buffer. The one-event buffer is an array where the position in the array denotes a unique quantity like the ADC-value of a detector. For experiments with large arrays of similar detectors, the more efficient approach of storing the value together with its detector number in a queue-like structure was used. This facilitates the automatic incrementing of histograms (all similar quantities in one one-dimensional histogram or in a two-dimensional histogram with the detector number as the Y-axis). The Plastic Ball, for example, consists of 815 identical modules where each module can provide four pieces of information (E, two DEs, and a time signal). In the input buffer the information is stored as ADC-address and ADC-value for all Es, then the information for the DEs, and so on. The decoding routine translates the coded address into a modulenumbere and fills the number and the value in the corresponding queue. After all the information has been filled into the individual queues, the queues are scanned for the same modulenumbers, and the information belonging to a single module is inserted into another queue.

The memory spectra are always treated as REAL*4 words for simplicity. The automatic incrementing routine handles bit-spectra, normal one- and two-dimensional spectra from the one-event-buffer, and several types of spectra from the queue structures: increment a one-dimensional spectrum for each element of a queue using either the element number or the contents, and increment two-dimensional spectra using the element number and the contents as x and y or contents vs contents for a selected range of element numbers. The spectra definition commands also allow the definition of nonincrementing spectra, which can be filled by an user routine with increments differing from 1. Because the definition of spectra is done according to the HBOOK[2] standard, a command was introduced to read data-files that were written with the HSTORE command from HBOOK. This allows the user to use the same display utilities for the on-line data and for the output of CASCADE calculations, for example.

The graphic display of data is based on an expanded version of PLOT10.[3] Graphic output exists in the form of histograms, contour plots, pointdensity plots, and scatter plots. The spectra can be fitted with gaussians, polynomials, and exponential functions, and the result can be displayed in the histograms.

In the last Plastic Ball experiment at the Bevalac, LISA was already used for on-line analysis. Compared to the old MULTI system, running on the PDP 11/44, the LISA package on the VAX 11/780 was 200 times faster with exactly the same analysis. The data were read from the shared disk and analyzed with two concurrently running LISAs with different user routines. Both programs were able to keep up with the newest data without skipping events. The data rate during the beam spill was approximately 100 events with a mean length of 600 bytes in 800 milliseconds, followed by a beam-off period of 5 seconds. This is an average of 10 kbytes/sec. For each event the data were unpacked, the information belonging to a single detector was packed and filled into queues, the contents of the queues were scanned under conditions, and the results were stored in about 50 one- and two-dimensional spectra.

This work was supported in part by the Director, Office of Energy Research, Division of Nuclear Physics of the Office of High Energy and Nuclear Physics of the U.S. Department of Energy under Contract DE-AC03-76SF00098.

References

1. J.F. Bartlett, et al., FERMILAB Report PN-97.4
2. CERN Data Handling Division
3. Trademark Tektronix

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720