Managing the SLD Offline on a Shoestring via WWW

Richard Dubois

Stanford Linear Accelerator Center

As a traditionally undermanned experiment, SLD has found it necessary to automate as much of its offline machinery and monitoring as possible. This includes all documentation, user education, group communication, monitoring of local and remote MC Farms, shift assignments, integrity of the code and monitoring of the analysis cluster. Additionally, monitoring of the data processing is performed on the web by remote users. They can access statistics, plots and event displays to ensure high data quality. The SLD web [a] is designed to be an all-encompassing tool that allows maintenance of the offline system in good working order by a small group of people.

[a] http://www-sld.slac.stanford.edu/sldwww/sld-working.html

Key words: WWW; system monitoring; user training; Oracle

1 User Training and Resource Monitoring

An extensive User Workbook [1] details in depth for new users, and as a refresher for veterans, the environmental elements that are unique to SLD, including our analysis shell, coding examples, input/output, our data structure manager, the operating system and debugging. It provides a jumping off place to more detailed documentation. We devoted about 1 man-year to this project and found that this was time very well spent. It has reduced the queues outside the doors of the support people to a mere trickle.

Users keep the workbook open in a browser while they work through the examples in a terminal window or analysis GUI. A good effort was made to embed screen dumps in the workbook to make it very clear to the user what he could expect to see. The workbook also intersperses usage suggestions and policies in with the learning material. It should take the user about two weeks

[1] http://www-sld.slac.stanford.edu/sldwww/workbook/workbook_prod.html

to go through the workbook, though it allows them a one-day mini-tour which is sufficient to get them started analyzing data.

Experts and users alike can monitor from the web many properties [2] of the analysis cluster: system logs; batch jobs (including log and source files); batch queues; available disk space; the tape staging system; the tape cartridge system; our code distribution system; CPU usage trends.

2 Access to Oracle Databases from the Web

Our automated data processing server stores all of its input/output and state information in an Oracle database. This includes over 100 pieces of data describing the reconstruction of each run. We wrote a generic interface [3] that allows browsing of our Oracle tables and of output files on our processing cluster. The interface requires that the data in Oracle be organized on run or day boundaries, which can be viewed as tables, plots or sums and averages.

Examples of recorded quantities are beam polarization, number of Z^0's found, polarization asymmetries, $< z >$ of the interaction point from tracking and so on. Several quantities are standard to monitor and are available as predefined ratios to plot or tabulate over a selected time or run range. Any of the quantities or pairs of quantities for ratios can be displayed.

2.1 Monitoring Production Processing

Since the production processing is fully automated, it is critical to check [4] in detail that no runs have been missed, and that the results are reasonable. Our online run record keeping is also kept in Oracle, so we provide a comparison on the web of offline and online databases to verify that all runs taken have been, or are in the queue for being, processed. A web view of the processing job batch queues allows us to check that unprocessed jobs are running or queued and what their state is. In case of failure, we can access the job log files from the web. The server pages us if it detects any serious errors itself in run processing, for example if a subsystem does not log any data for an entire run.

---

[2] http://www-sld.slac.stanford.edu/sldwww/slacvx/slacvx.html
[3] http://www-sld.slac.stanford.edu/sldwww/runinfo/standint.html
[4] http://www-sld.slac.stanford.edu/sldwww/offshift/processing.html

2.2 Offsite Monitoring of Data Quality

We have exploited manpower available remote to the SLAC site for prompt review of the processing results. Remote collaborators are assigned offline shifts to review the data and report daily to the run captain. A checklist of statistics, plots and event displays to review is provided. Some 125 individually selectable histograms from the full reconstruction can be viewed using a web histogram browser. We intend to extend this facility for the upcoming run by providing standard groupings of histograms and for overlays of known standards. We find this tremendously enhances the reviewer's ability to spot anomalies. In addition we will add automatic checking of rote statistics, freeing the user to think more about the data rather than doing bookkeeping.

2.3 Monitoring Monte Carlo and Reprocessing Farms

SLD operates processing farms [5] both at SLAC and at Vanderbilt University. The setups are clones of each other. The SLAC farm is operated by an offsite collaborator. All monitoring of the farms activities is done via the web, again important since the farm operations are entirely automated. The monitoring includes status of each job, examining log files and volume of output to tape.

2.4 Verifying Code Integrity

The offline production system is comprised of a large amount of code segregated into many shareable images. We verify [6] that the code remains functional and doing what we expect by running the same job every night to do full simulation and reconstruction of a particular dataset, always using the same random number seeds. Hence the output of these jobs should always be identical. This is verified by comparing the same 125 histograms as checked in the offline shifts against a standard. Should a change be detected in the histograms, the keepers are notified. Examination of the individual χ^2 from the web readily allows tracking down the change in the code repository. When sanctioned changes are made to the code, the standard is remade.

Several elements are stored in Oracle for each job. These track the CPU and elapsed times for each step, memory use, χ^2 of the histogram comparison, status and location of log files and the standard that was used, plus some setup

[5] http://www-sld.slac.stanford.edu/sldmcwww/mc/mcpage.html
[6] http://www-sld.slac.stanford.edu/sldwww/runinfo/tdint.html

information. These quantities track several aspects of the system: correctness of the code; shareables are properly linked; memory usage; and throughput.

2.5 Access to Public Datasets

Monte Carlo studies require generation of more than 10x the data sample. With the addition of specialty datasets, users must have an easy accessible, well organized presentation of what is available. We augmented the standard dataset description in Oracle with descriptions of the 'task' that produced the dataset, be it from Monte Carlo or data. The system [7] allows the user to obtain the names of the datasets that apply to the 'open' command, which also queries Oracle, in the analysis shell.

Our original system has been overwhelmed by the sheer number of datasets that the user must scroll through. We are reorganizing the system to improve the search capability, enhancing it over the original segmentations of Production, Monte Carlo and Data. We will provide for search by code version, category (eg multi-hadrons, τs) and allow searches for keywords in the dataset task comment fields.

3 Conclusions

We have made extensive use of the web directly and with scripts accessing our Oracle databases to maximize the efforts of the few people we have supporting the SLD offline system. We have found a large benefit from creating a good user workbook, monitoring tools for the processing and MC farm and analysis cluster, and access to public datasets.

[7] http://slacvm.slac.stanford.edu:8080/htbin/SLDCAT