Automated QA framework for PetaScale data challenges

G. Van Buren, L. Didenko, J. Lauret, E. Oldag and L. Ray

1 Brookhaven National Laboratory, P.O. Box 5000, Upton, New York 11973-5000, USA
2 University of Texas, Austin, Texas 78712, USA
E-mail: gene@bnl.gov

Abstract. Over the lifetime of the STAR Experiment, a large investment of workforce time has gone into a variety of QA efforts, including continuous processing of a portion of the data for automated calibration and iterative convergence and quality assurance purposes. A rotating workforce coupled with ever-increasing volumes of information to examine led to sometimes inconsistent or incomplete reporting of issues, eventually leading to additional work. The traditional approach of manually screening a data sample was no longer adequate and doomed to eventual failure with planned future growth in data extents. To prevent this collapse we have developed a new system employing user-defined reference histograms, permitting automated comparisons and flagging of issues. Based on the ROOT framework at its core, the front end is a web based service allowing shift personnel to visualize the results, and to set test parameters and thresholds defining success or failure. The versatile and flexible approach allows for a slew of histograms to be configured and grouped into categories (results and thresholds may depend on experimental triggers and data types) ensuring framework evolution with the years of running to come. Historical information is also saved to track changes and allow for rapid convergence of future tuning. Database storage and processing of data are handled outside the web server for security and fault tolerance.

1. Introduction

1.1. The STAR Experiment

For the past decade, the STAR Experiment at RHIC has been collecting data from high energy collisions of nuclei at the Relativistic Heavy Ion Collider facility (RHIC), located at Brookhaven National Laboratory [1]. Operations are generally constant for up to half of each year, and are projected to continue for another decade to come.

As is typical in a long-lived modern nuclear physics experiment, STAR is composed of numerous detector and control subsystems which undergo continual and significant upgrades and additions. Examples include a methodical increase in calorimetry coverage, time-of-flight, tracking elements, photon detectors, and muon chambers. Improvements in duty cycle efficiencies for both the experiment and collider have come with operational experience, but of greater impact have been the advancing capabilities of the data acquisition through a few orders of magnitude, from an original design event-recording rate of \( \sim 1-10 \text{ Hz} \) to nearly \( 1 \text{ kHz} \). Further advances are in discussion. STAR can now record in half an hour as many events as it took over weeks of running when it first began studying nuclear collisions. The consequences for such growth in components and capabilities are ever-increasing volumes of information to handle,
analyze, and inspect for quality. In 2010 alone, STAR wrote over a petabyte of raw data to tape and the challenge to ensure data quality has risen commensurately.

1.2. Quality Assurance in STAR
Between data acquisition and data production in STAR, two frameworks of quality assurance are supported. Online QA is applied to raw data as it comes in at the experiment, providing immediate feedback during data acquisition of general hardware performance. Complementarily, Offline QA utilizes the full event reconstruction software to enable more detailed validation of the data, assessment of the calibrations, and certification of the software.

STAR is equipped with an automated “FastOffline” framework for quickly processing the reconstruction of a portion of the incoming data [2]. Utilizing the available calibrations at the time of processing, FastOffline provides a means of testing and iterating on calibrations as well as a vehicle for examining data quality. This runtime cornerstone of the QA process has been operational in STAR since 2001, reliably enabling early investigation of the data. As calibrations are not necessarily finalized, physics gleaned from this reconstruction must be held in an acutely preliminary light. Histograms for Offline QA are generated during FastOffline within the ROOT framework, and are typically available for examination within a few hours after acquisition of a dataset [3]. Traditionally, personnel sitting shifts (one person per week-long shift) have visually studied the output histograms daily and reported on observed issues using a set of tools specific to the purpose [4].

The Offline QA framework can be and has been also used independently of FastOffline to serve other QA needs. Examples of this include QA of simulations and nightly library tests, and validation of new code libraries through a specific set of histograms generated at the time codes are frozen.

Over 10 years of operation and QA in STAR have provided valuable experience to establish lessons learned and growing needs, and it is clear that this long-standing scheme must be re-worked to handle the immense growth of the experiment and future data extents, or it will not succeed. In this paper, which focuses on STAR’s Offline QA framework, the major lesson points are enumerated and the design choices to address them in upgrading the framework with automation are presented. Further details of the implementation are also discussed, walking through several use cases.

2. Reasons & Requirements for an Automated Framework
2.1. Limited human resources
During 2010 alone, FastOffline processed approximately 320,000 files from nearly 10,000 data acquisition runs. Depending on trigger setup, each run produced between 100 and 300 histograms to be examined, leading to a staggering couple million histograms to be examined in total. The quantity of data to be reviewed each day exceeds the capacity of a one person crew to evaluate using traditional visual histogram scanning methods, forcing the shift personnel to skip some datasets. This opens the door to unnoticed issues, and sets up the possibility for severe, perhaps even disastrous omissions in the QA as the data volume growth continues. Downstream analyses then must redundantly perform QA for themselves, negating the benefits of the work done on shift, and likely forfeiting the ability to correct a problem as valuable feedback is not given early enough. Before demanding additional human resources from the collaboration for shift work, it is worth maximizing the efficiency of the shift personnel by reducing the time necessary to analyze the histograms.

As the study of QA histograms involves comparison to a reference histogram, an automated comparison performed by computer may generally determine the data quality very quickly, alleviating the shift crew of significant time spent inspecting the plots. The automation must be implemented with flexibility such that a variety of methods may be chosen to perform the
analysis, while allowing alternative methods to easily plug into the framework. STAR’s existing C++/ROOT tools for plotting QA histograms can provide a chassis for performing the analyses and encoding comparison algorithms.

2.2. Individual perceptions
With each new week of operation, a different person may take the Offline QA shift. This introduces the subjective human factor of individual perception on whether data quality is sufficiently poor to warrant reporting as an issue, and results in variation of provided QA between personnel. This inconsistency and lack of standardization forces additional work upon the experts who try to understand and resolve issues, as confidence in whether issues were properly reported is weakened. Again, the value of the time and effort spent by the QA shift workers is significantly diminished.

While the aforementioned automatic comparison can simply deliver a pass or fail result based on a quantitative analysis, it is important to recognize and appreciate the ability of the human mind to handle pattern recognition, which can often exceed the discrimination power of a simple algorithm. Situations of few statistics are a particular example.

As a compromise, automated flagging of whether the result constitutes a passing, questionable, or failing grade, along with the quantitative results of the analysis, can be provided to the shift crew as systematic and objective guidance. The shift crew can then decide from this information (and more, if they choose to inspect the histogram) whether the reporting of an issue is warranted.

A supplementary approach to improving issue reporting uniformity is to make available historical records of analysis results and correlations with other factors. Examples may include time of data acquisition, time of data processing, environmental or operational variables at the experiment, or the results from other analyses. These factors can be saved in databases, and a desired correlation can be presented as a graph of the result versus another factor.

2.3. Evolving data conditions
There are many dynamic characteristics to the data acquired in STAR. Prominent examples include operating the collider with different nuclear species and energies, and a variety of trigger setups may be used for a single collider operating condition. In finer detail, portions of the detectors may be turned on and off for extended periods. These complicate the automation of reference comparison, and drive the need for references which may similarly evolve.

A database of references must be kept, and the appropriate reference can be selected for a given dataset’s QA analysis. It is important that reference creation and updating be kept easy to use to promote their maintenance by individuals tasked with doing so, requiring an uncomplicated user interface which can be integrated with STAR’s existing QA browsing tools.

3. Implementation
3.1. Components
Nearly all aspects of the updated STAR Offline QA framework are available for use through a web-based graphical user interface, simplifying remote usage. The GUI is intended by default to serve the needs of the Offline QA shift crew, while providing access to tools necessary for experts and maintainers to configure the analyses and references. Contextual help is provided in popup windows for each stage of the GUI to guide the user as necessary. A few panels from the GUI are presented in Figs. 1, 2, and 3. These are discussed in the context of their usage below.

The processes of performing the automated comparisons and any other handling of the data files are kept decoupled from the web server by running on a separate node with access to file systems not made available to the web server. In addition to buttressing security for those file
systems, this segregation also prevents any faults in data processing from impacting the web server, which provides numerous services to many other areas of the experiment. Communication between the various components occurs via database, with restrictions on entered content to ensure security. Caching on disk of reference data and analysis specifications is implemented on the processing node to reduce overhead in obtaining sizable content from the database.

3.2. Defining references
As noted previously, the evolving experiment mandates ease in defining the references. To that end, the GUI allows two choices. An expert user may select all histograms in an examined dataset to be declared as a new reference set. Or the user may mark specific histograms from a dataset to replace those in an existing reference. In either case, a new reference set is generated, and must be tagged for the kind of data for which it is applicable (including run year and trigger setup). Multiple advancing reference sets may be defined for any data variety, so automatic versioning (and timestamp-ing) are implemented to maintain records and reproducibility.

The GUI component for defining a new reference is shown in Fig. 1. In addition to the tags, comments may be attached to any new reference set.

![Figure 1. An example of the GUI panel for defining a new reference histogram set. Any histograms to be updated are listed, and input options for tags and comments. Versioning is automatic. A contextual “Help” button in the upper right corner provides a popup window with documentation specific to this panel.](image1)

![Figure 2. An example of the GUI panel for defining a histogram analysis, including the mode of analysis test, any options for that mode, and the value of the pass/fail cut. A “View Trends” button displays a popup graphical viewer of historical results and permits correlation with other factors. Histogram titles and descriptions can also be updated here.](image2)

3.3. Defining analyses
Experts must decide what analysis method will be executed and what result constitutes a pass or fail for each histogram to be tested. Additionally, some STAR QA histograms exist with multiple,
independent instances for different trigger classes within a dataset (such as minimum bias triggers alongside specific physics triggers). In the updated framework, analysis specifications for any particular histogram are available on a trigger class level, as well as a fallback for all trigger classes for that histogram.

Selection of the analysis method (or mode) is made from a provided menu, including $\chi^2$ and Kolmogorov histogram tests, which are already provided in ROOT, as well as any custom algorithms provided (code which can be easily plugged into the analysis engine) [5]. The result of the analysis should be a scaler in the range $[0,1]$, representing increasing confidence in agreement between the data and the reference with increasing result value. Definition of pass/fail is then made by declaring a value within that range below which confidence of agreement is too weak to overlook as a possible issue.

Arriving at an optimal pass/fail cut cannot always be immediate. The historical records of analysis results, graphically displayed as trends in the GUI, can significantly aid in determination of such a cut by presenting a statistical distribution of acceptable result values. The cut can then be rapidly tuned over the first few days of operation until reasonable values are found.

Fig. 2 shows the relevant panel of the GUI, which also has an examine-only mode (no editing) for non-experts. It additionally features a section for display or edit of the histogram title and description.

3.4. Viewing results
The primary focus for QA shift personnel is the list of histograms which failed their respective analyses. This is the first panel presented to the user after the analyses are completed, giving an immediate perspective on the data quality. With each histogram is also indicated the numerical analysis result and cut value against which it was compared. The option to record the results is also made available to the user to enable the historical studies mentioned previously. Fig. 3 illustrates how this panel appears to the shift crew.

To go a step further, the user may change the view such that all histograms are listed, with further color coding to indicate the result status, as visible in Fig. 1. Passing results which are close to the cut are distinctly highlighted as questionable to help identify issues while cut values are not yet optimal, or in situations when a clear division cannot be made between passing and failing results (i.e. the distribution of “good” and “bad” results overlap).

![Figure 3](image.png)

**Figure 3.** An example of the GUI panel for listing histograms which have failed their reference analyses, along with their analysis results and the cuts which they fell below. The failed histograms are highlighted in red, with the full color-coded histogram list available via a single click, using different color highlighting codes for other statuses (see Fig. 1). Also shown to the left is the tabular layout for histogram access using location in a historical, graphical context.

Because the Offline QA framework is not completely new, organization of the histograms has a more than 10 year history involving trigger classes and layout in a graphical document. This
organization has for now been preserved, to maintain a degree of continuity for those persons who have been involved with the Offline QA for several years, via a simple tabular layout and location coordinates for histogram lookup. The traditional graphical documents are also still made available. A more hierarchical arrangement may supplant this organization as the program evolves.

4. Summary

Traditional means of visually comparing QA histograms between data and reference are insufficient for the challenges of the PetaScale era. Adequate quality assurance can only be achieved through reduction of the human workload to improve efficiency, standardization of analyses to provide uniformity and confidence, and simplified maintenance of the automation tools to allow evolution and adaptation. In STAR, we are addressing these requirements through updates to our Offline QA framework centered on automated analyses. The updates include flexible tools to define these analyses and study their results, automatic flagging of potential issues, and structured administration of reference data. The framework is fronted by a helpful and simple GUI decoupled from the data processing and any restricted access systems.

Deployment of the tool is happening for the winter 2011 operation of RHIC and STAR, and we are already looking forward to further growth of the tools. One such possibility is the automatic triggering of analyses as soon as histograms from production become available with sufficient accumulated statistics. The framework’s versatility is also expected to ease planned incorporation of other aspects of QA, such as validation of new code libraries. We believe that STAR’s updated Offline QA framework provides a platform which can sustain QA through PetaScale challenges and forward.

5. Acknowledgments

This work was supported by the Office of Nuclear Physics within the U.S. Department of Energy’s Office of Science.

References

[1] M. Anderson et al., “STAR detector overview”, Nucl. Instr. Meth. A499 (2003) 624.
[2] J. Lauret, “FastOffline”, http://drupal.star.bnl.gov/STAR/comp/prod/fast-offline.
[3] I. Antcheva et al., “ROOT - A C++ framework for petabyte data storage, statistical analysis and visualization”, Comp. Phys. Comm. 180 (2009) 2499; http://root.cern.ch/.
[4] G. Van Buren et al, “Avoiding the Tower of Babel syndrome: An integrated issue-based quality assurance system”, CHEP ’06 (2006); http://indico.cern.ch/contributionDisplay.py?contribId=157&sessionId=5&confId=048.
[5] I.M. Chakravarti, R.G. Laha, and J. Roy, “Handbook of methods of applied statistics, Volume I”, John Wiley and Sons (1967) 392; http://www.itl.nist.gov/div898/handbook/eda/section3/eda35f.htm; http://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm.