

LHC SUPERTABLE

M. Pereira, T.E. Lahey*, M. Lamont, G.J. Müller, D. D. Teixeira, CERN, Geneva, Switzerland
E.S. McCrory, Fermilab, Batavia, USA

Abstract

LHC operations generate enormous amounts of data. This data is being stored in many different databases. Hence, it is difficult for operators, physicists, engineers and management to have a clear view on the overall accelerator performance. Until recently the logging database, through its desktop interface TIMBER, was the only way of retrieving information on a fill-by-fill basis. The LHC Supertable has been developed to provide a summary of key LHC performance parameters in a clear, consistent and comprehensive format. The columns in this table represent main parameters that describe the collider's operation such as luminosity, beam intensity, emittance, etc. The data is organized in a tabular fill-by-fill manner with different levels of detail. Particular emphasis was placed on data sharing by making data available in various open formats. Typically the contents are calculated for periods of time that map to the accelerator's states or beam modes such as Injection, Stable Beams, etc. Data retrieval and calculation is triggered automatically after the end of each fill. The LHC Supertable project currently publishes 80 columns of data on around 100 fills.

MOTIVATION

Even before the LHC came into operation it was already evident that there would be a need for means of easily obtaining general statistics on the performance of the accelerator. The LHC has been in stable activity for several months now and a large amount of data has already been produced. The increase of accelerator's performance and the thirst of physicists for data highlighted the need for a tool that processes and displays information on performance and other statistics.

Although the data is commonly available and accessible through dedicated tools it is troublesome to create an overall view of the operation of the LHC. One of the main data sources is the LHC Logging Service [1]. This service consists of two databases where data from all LHC devices and parameters is stored. Access to these databases is done through a Java API or through a desktop application called *TIMBER*. While the Java API allows other developers to retrieve data programatically, *TIMBER* can provide plots and data export into different formats. Although these tools fulfill their intended purpose they are much better at providing data of specific aspects of the LHC than producing the full picture. Another limitation of the available data mining tools available is that their use is typically restricted within CERN.

* On leave from SLAC, Menlo Park, California, USA

As for the Tevatron [2], an application was developed to provide summary data of the performance of the LHC.

THE SUPERTABLE

The supertable is a spreadsheet which provides a summary of LHC key parameters that can be used to assess the accelerators performance [3]. The columns of this table represent beam or accelerator parameters, such as, luminosities, beam intensities, crossing angles, filling scheme, etc. Each row contains the values of these parameters for each fill. The intent of the LHC Supertable is not to provide detailed data on each of these parameters. For that purpose, specialized and dedicated tools have been developed over the years and are used by experts on each of the domains. Instead this application aggregates and formats data in a concise and clear way so that physicists, experts and management alike can be given a general overview of the collider. In summary the LHC Supertable aims at:

- Providing a summary of the key parameters that detail the performance of the LHC.
- Providing a concrete mechanism for determining these key parameters on which everyone can agree (or, at least, through which the parties that disagree can have a basis for their argument).
- Displaying data in easily understandable formats such as web pages and excel sheets.
- Being a small, simple and concise resource.

A similar system has been implemented in Fermilab for the Tevatron accelerator. Fermilab's Supertable has been in operation since 2004 and currently stores data on thousands of fills. In order to profit from Fermilab's experience with their implementation, the LAFS workgroup was called in to give their input and contribution to the project. As a result, CERN's Supertable architecture is greatly inspired on Fermilab's experience [4]. A proposal was made [5] to implement a web application that would gather data from various data sources, process and store in a dedicated database and finally publish the digested data. The solution would profit from Fermilab's experience while the implementation would take in account the specificities of the LHC.

IMPLEMENTATION DETAILS

The LHC Supertable is a web application developed in Java and using the Spring framework. The application is divided in two components: persistence and web.

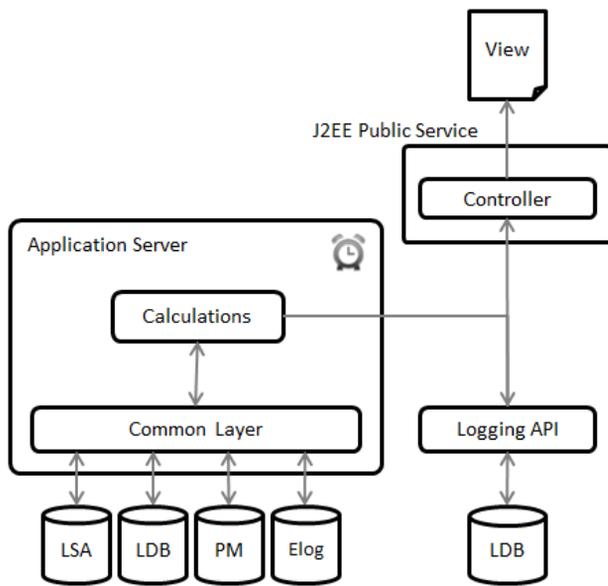


Figure 1: Simplified architecture overview of the LHC Supertable application.

Persistence Module

The persistence module is responsible for gathering data from the various data sources, processing the data and persist it in the database. This process is triggered by a cron job scheduled to run every 30 minutes. When the process is started there is first a verification if there are new fills to be calculated. If so, the *processor* will iterate through the columns of the supertable, calculating them one by one for that particular fill. The algorithms themselves are encapsulated in Java classes called *cells*. Each *cell* in the Java code is responsible for the calculation of one or more supertable cells. In example, the beta star scheme in the supertable is calculated within the *BetaStarSchemeCell* Java class, while luminosities are calculated in the *LuminosityCell*. With this separation of concerns each *cell* becomes an independent unit responsible for processing a set of columns in the table. Figure 1 shows a simplified view of the architecture.

Each attribute in the Supertable has an associate algorithm description and unit. These two elements are of great importance as beam or accelerator parameters may be calculated in different ways. However, the algorithm used to calculate a certain property may change over time. The Supertable offers the possibility to change algorithms in-between fills or recalculate past fills. The calculation of parameters, i.e. peak ATLAS luminosity, typically involves the retrieval of data from one of the data sources. The Supertable is currently gathering data from the Logging Database, LSA [6], Postmortem [7] and the E-logbook. The current implementation is flexible enough to support the introduction of new data sources. Since one of the primary motivations of the Supertable is for a simple, clean and organized presentation the columns have been classified into one or more specific categories, which we call

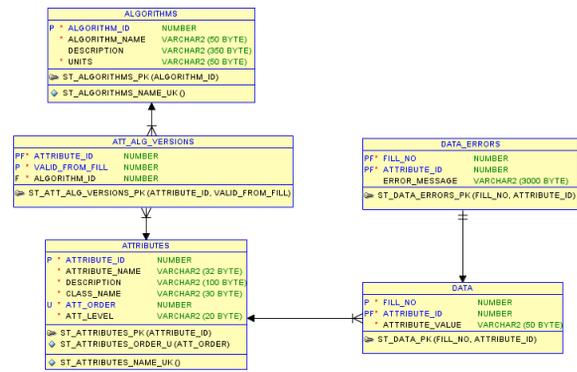


Figure 2: Implemented database schema.

views. The user can select one of these simple views to focus his/her attention on his/her area of interest. For example, the Luminosity view will only show columns concerning luminosity. The organization of columns into *views* provides different levels of detail according to the user's needs.

Data Storage

After all columns have being calculated the data is persisted in the database. For that purpose a schema has been created in the Logging Database. The Logging Database ensures the persistence of the data for the living time of the LHC. Hence, for the sake of consistency and centralization of services, the data of the Supertable was stored in the same environment as the rest of the logging data.

As shown in Figure 2 the database schema is quite simple but provides a lot of flexibility in the supertable. The table ATTRIBUTES is where the columns of the Supertable are defined. It currently holds 87 different attributes and, as mentioned before, each one has a class name field which corresponds to the name of the Java class responsible for calculating the value of the attribute. Additionally, the ordering and level of the columns are also defined in this table allowing to easily change the appearance of the Supertable in the web interface. Add, remove and edit operations on attributes can be performed via a dedicated administration package. This data driven approach makes the structure more flexible making it unnecessary to redeploy the application module responsible for the reads and writes in the database. In the ALGORITHMS table are defined both the algorithm and units per attribute of the Supertable. It was decided not to separate algorithms and units in different tables since they are strongly coupled and the separation wouldn't bring a significant added value in terms on flexibility. Since it is possible to change the algorithm being used in the calculation of a Supertable column without having to recalculate past fills an extra table was needed. The ATT_ALG_VERSIONS table keeps track of which algorithms were used to calculate each attribute throughout the calculated fills. Whenever a new algorithm is added

for the calculation of a column, this table will contain the fill number from which the algorithm went into effect. In case errors occurred at the time of the calculation of a fill these are stored in the data errors table. These errors should be displayed to the user to explain the reason behind eventual missing values for some columns. The choice of the logging database as final location for this scheme brought along an inherent limitation. Due to how the backup process occurs in the logging database, data becomes read only after a undefined period of time. In practice this means the recalculations of fills can only be performed provided that the backup process hasn't handled those particular fills. This schema doesn't currently offer any kind of versioning of data but we believe that this could be one of the solutions to work around the mentioned constraint. Data from the Supertable can be programmatically accessed through the Logging API using the *SupertableController*.

Web Module

The second module of the LHC Supertable is the web interface. The technology chosen was Google Web Toolkit. GWT allowed the creation and maintenance of a powerful Javascript front-end developed 100% in Java. The interface was made clean and as simple as possible but providing enough functionality to be useful. It displays data in tabular form and provides the algorithms used to calculate each parameter as well as the units. Columns are organized in 5 views and by default only a small subset will be displayed. In addition, exports in Excel and JSON are available allowing users to use data as they see fit.

CURRENT STATUS & OUTLOOK

The project as specified has been fully implemented as shown in Figure 3. The LHC Supertable currently publishes 87 columns of data for over 300 fills. Several views are available providing more or less detail according to the users needs. Data can be exported in both excel and JSON [9] formats along with a static html version. Despite the implementation of all planned functionalities the Supertable suffered from a great problem early on: quality of data. For some of the trickiest parameters to calculate algorithms were adopted from Fermilab's Supertable such as emittances, luminosities from emittances, among others. However, some of the implemented algorithms didn't translate well to the LHC's reality. Even though some of those algorithms have been changed, some errors still persist in the calculations. At the same time the LHC Supertable project was being developed another project was under way. The LHC Performance and Statistics although similar in goal started from a different point.

The Statistics application focused at first on creating a repository on ROOT files containing data on a set of key performance parameters of the LHC during the course of a fill. These files are meant to be used by physicists or experts who want to explore in more detail what happened

Figure 3: LHC Supertable web interface.

during a particular fill. Additionally, the Statistics application also made available a set of ROOT generated plots and a small table describing a limited subset of parameters also characterized by the Supertable. After some analysis of the functionality and architecture of both applications it was clear that these weren't concurrent systems but rather complementary. Hence a decision was taken to merge the two systems and provide a better service to the users of both applications. At the time of this writing the developer teams of both projects are reviewing the calculation algorithms used for the LHC Supertable which will be later applied in the new application. A system that provides statistics on the operation of the LHC is very much needed and of the utmost importance, however, it is rendered useless if it provides erroneous data.

CONCLUSIONS

The LHC Supertable has been designed, implemented and commissioned to provide a clear and consistent view of LHC operations data to all interested parties in the CERN community. As with any complex project, this project has evolved over its short lifetime to adapt to the environment in which it must exist. While input from the Avatron Supertable has been invaluable, the LHC calculations often have been subtly different from the comparable Tevatron calculations, leading to some confusion in the customer base. As the user base grows it has become clear that this application is filling a void and providing functionalities that were not available in the existing expert tools. The merge with the LHC Statistics and Performance project is ongoing and should soon go into production. The final product will bring users a new set of functionalities and data for our users.

REFERENCES

- [1] C. Roderick et al., "The LHC Logging Service: Handling Terabytes of On-line Data", ICALPCS'09, Kobe, Japan, January 2010.
- [2] T. Bolshakov et al., 'Data Acquisition and Analysis for the Fermilab Collider RunII', June 2004, FERMILAB-Conf-04/093-AD.

- [3] D. McGinis, “Collider Supertable-Table for a Modern Hadron Collider”, September 2008.
- [4] Elliott McCrory, “The LHC Super-Table Project”, June 2010, <http://beamdocs.fnal.gov/AD/DocDB/0036/003628/002/Experience%20with%20Super%20Table.docx>
- [5] Elliott McCrory, “A Proposal for the Super-Table Generation Software”, June 2010, <http://beamdocs.fnal.gov/AD/DocDB/0036/003628/002/A%20Proposal%20for%20a%20Framework%20for%20Building%20the%20SuperTable.docx>
- [6] G. Kruk et al., “LHC Software Architecture [LSA] - Evolution toward LHC Beam Commissioning”, ICALEPCS’07, Knoxville, USA, October 2007, WOPA03, p.307.
- [7] Odd Oyvind Andreassen et al., “The LHC Post Mortem Analysis Framework”, ICALEPCS’09, Kobe, Japan, January 2010
- [8] Elliott McCrory, “Data Storage Schema for the LHC Super-Table, June 2010, <http://beamdocs.fnal.gov/AD/DocDB/0036/003628/002/Data%20Storage%20Schema%20for%20the%20LHC%20SuperTable.docx>
- [9] Introducing JSON, <http://www.json.org/>