

INTEGRATION, PROCESSING, ANALYSIS METHODOLOGIES AND TOOLS FOR ENSURING HIGH DATA QUALITY AND RAPID DATA ACCESS IN THE TIM MONITORING SYSTEM

A. Suwalska, M. Bräger, M. Brightwell, E. Koufakis, R. Martini, P. Sollander, CERN, Geneva, Switzerland

Abstract

Processing, storing and analysing large amounts of real-time data is a challenge for every monitoring system. The performance of the system strongly depends on high quality configuration data and the ability of the system to cope with data anomalies. The Technical Infrastructure Monitoring system (TIM) addresses data quality issues by enforcing a workflow of strict procedures to integrate or modify data tag configurations. TIM's data acquisition layer architecture allows real-time analysis and rejection of irrelevant data. The discarded raw data (90,000,000 transactions/day) are stored in a database, then purged after gathering statistics. The remaining operational data (2,000,000 transactions/day) are transferred to a server running an in-memory database, ensuring its rapid processing. These data are currently stored for 30 days allowing ad hoc historical data analysis. In this paper we describe the methods and tools used to guarantee the quality of configuration data and highlight the advanced architecture that ensures optimal access to operational data as well as the tools used to perform off-line data analysis.

INTRODUCTION

Being successful in providing software for monitoring systems is not obvious. And not because of the complex mix of requirements the system needs to comply with. In our rapidly evolving world, new technologies allow us to do much more than what was possible a few years ago. But are we humans able to change as fast as the technology? Not sure. Even the most complete monitoring and control systems can fail to meet their goals if procedures for data definition are not clearly specified and followed.

There are more risks related to monitored data. Nowadays, automatic systems generate a colossal amount of data, but are we able to effectively analyse it in a timely way and benefit from the information? Are our systems able to withstand this enormous data traffic? Whereas it is reassuring to have a wealth of information at hand, the complex processes that handle the underlying data can often be the source of problems. Incorrectly defined sources can generate so much traffic that the system's fluidity is lost together with its monitoring capacities. The provision of a high quality monitoring system depends on a number of factors, namely, rigorous data integration procedures, thorough data validation, the regular analysis of the performance and behaviour of the

system, as well as ensuring that the best suited technology is chosen in each domain.

For the needs of this paper we will briefly describe the architecture of TIM and how it is used. Operational since 2005, TIM is built on a typical 3-tier concept with a data acquisition layer DAQ, business layer and client applications layer. The DAQ primary mission is measurements collection, analysis and filtering. From this layer data are sent either to a business layer built on the Cern Control and Monitoring (C²MON) platform[1] or to a statistics module. A business layer carries out further data processing functions before transmitting the required results to the client application layer. Since TIM is an entirely data driven monitoring system, a configuration database and its tools are used to ensure the high quality of defined data.

TIM monitors a heterogeneous set of data points that originate from different sources, managed by different organisation units and covering a variety of infrastructure applications. These data points are used by the Cern Control Centre (CCC) operators who need to respond quickly and accurately to alarms as well as interpret the behaviour of synoptic diagrams that represent the current state of the various systems being monitored. In such an environment it is essential that the data seen by the operators is both correct and easily understandable. In addition, operators must be protected from floods irrelevant and redundant data. The system must be able itself to detect and eliminate a potential data overload, transmitting only relevant messages. Further, the system must deliver operational data in a timely way to operator consoles and also allow detailed data analyses at all time.

DATA INTEGRATION AND CONFIGURATION OF OPERATIONAL SYSTEM

In this chapter we present the methods used to ensure the data quality before even it is declared for being monitored and explain our motivation for building an extended tools for data verification and validation.

Data Integration

The configuration reference database for the monitoring data (TIMRefDB) holds all the definitions of what is monitored. Storing monitoring data in a single repository is important for several reasons. It enables on-line validation and business rule checking during data declaration and modification, guaranteeing the highest

level of data quality at all times. It allows the reconfiguration of the monitoring system at any time on demand, without otherwise disturbing the running of the system. It provides access to complementary information, which despite not being configured must be always accessible as it is essential to the correct interpretation of the events that the system detects, and ensures the appropriate actions are carried out.

Monitoring data are defined by "equipment groups", responsible for the different parts of the technical infrastructure. They use the Monitoring Data Entry System of the Technical Infrastructure (MoDESTI) procedure for declaring their data to TIM. This procedure is managed by CERN's Engineering Data Management System (EDMS) which ensures that each step in the data integration workflow is followed correctly by the appropriate people. This process applies to all changes to be made to the data that TIM monitors, whether these are data point creation, modification, or removal.

The first step is the request creation, where users submit specially designed Excel forms that contain the data point details to be handled by TIM. Basic checks are included in the form so that errors can be identified at the time of entry. Once the form is submitted, a periodically running job checks that all submitted requests are using valid input documents, and informs the users if the request cannot be processed. In the next step, the Technical Infrastructure (TI) operators manually check the input to ensure that requested data points are meaningful, and that in the case of alarm points, any specified instructions to be carried out are unambiguous and comprehensible. The TI operators may add information to the requested data points that will make them easier to understand when used in monitoring applications. They may also reject requests should the information not be clear or complete.

Once the operators have accepted the request, the job periodically checking requests will validate the requested changes against the TIMRefDB, to ensure that rules describing correctly defined monitoring points are enforced. The users are informed should any data validations fail, as they will have to make corrections before resubmitting the entire request.

Configuration Process

As described in the previous paragraph, TIM data are verified for correctness during the data declaration procedure thus simplifying the configuration process.

When a configuration request is received by the C²MON server, data are loaded to cache and made immediately available across cluster servers. At the same time it is also persisted to the operational database. Any changes related to data acquisition modules are automatically applied on the corresponding DAQs.

Tests of Newly Declared Data

Immediately after being configured, data points become active. However, for all newly declared or modified points and in particular for alarms, tests must be

performed before they obtain an "operational" status. There are several reasons for thorough testing of the alarm system that will be described below: data accuracy, trust in the system, operator understanding and official commissioning.

When a new system is connected to TIM and starts sending data, a number of alarms will instantly be sent and synoptic diagrams will display states, values and faults. It is essential that what is displayed is also accurate. Not only do we want to avoid "False positives" i.e. false alarms littering the screens, but we must also make sure that there are no "False negatives" where an actual alarm or value does not get through the system properly.

Tests should be carried out from equipment to operator console with, on the one end the technicians responsible for running equipment and on the other, control room operators responsible for the remote monitoring and operation of the installation. The data acceptance procedure is long and requires synchronization of activities between several specialists at a time. To help with this procedure we have introduced a MoDESTI testing tool based on the TIM declaration data and additional sets of tools for reporting the test results. This tool is always used during the tests and allows consistent feedback on test results and comments in real time during the test.

If some data or information about a point or an alarm is found to be wrong during the test, another tool, the web based reference database administration tool, allows quick and easy correction of single entries. If many entries need modification, the MoDESTI procedure is used. The test results can be consulted directly from the active alarm screen so that an operator can check when and by whom an alarm was tested as well as its result and comments on the test.

Unfortunately in some circumstances points and alarms are very difficult to test. Extreme temperatures or complete power outages cannot be triggered just to test the systems. In these cases simulations are performed either on the sensor level or at the local SCADA system. The test procedure still remains important as it will catch any transmission problem and will also validate data intelligibility and the availability of system components.

ACQUISITION LAYER DATA PROCESSING AND ANALYSES

In the previous chapter we have seen the importance of tools and procedures which ensure the quality at the data declaration stage. With that background we can now describe how the monitored data are verified and sorted on the DAQ layer to protect the server against irrelevant data. We will also focus on methodologies for analysing TIM data and how we benefit from the statistics collected.

Data Filtering

TIM monitors almost a 100,000 sensors from around a 150 TIM processes. The probability of monitoring a tag

with an erroneous parameter definition is high. Incorrectly scaled or faulty sensors can generate an avalanche of signals. Additionally, some sources, due to their design, send repetitive updates. TIM DAQ modules manage this large amount of data, and detect incorrect behaviour by applying several filtering strategies that allow only important data be transmitted to the central layer. Our statistics show that 97% of all signals received by TIM system are filtered out.

TIM system applies two levels of filtering. The initial, static filtering is governed by the tag configuration parameters defined by the user. For all tags we apply time dead band filtering, value dead band filtering is added for analogue values. In addition to this filtering, DAQs evaluate value-tag occurrences and apply dynamic filtering. In case of too frequent updates the DAQ automatically applies time dead band on the concerned tags to further limit repetitive or irrelevant value changes. The rejected data are transferred for analysis in the statistics module.

Off-line Analyses and Statistics

As mentioned earlier around 90 million rows are filtered daily. These raw data are stored in a *FilterLog* table for up to 48 hours - the time period needed to perform all computations which are later used in the analysis module. This table is the basis of statistics calculation; it is partitioned by date and time to optimize access. The filtered figures are correlated with the operational data written to the *ShortTermLog* to get overall statistics of TIM system.

Due to the large amount of data, statistical analysis is implemented in a modular way [2]. The first layer manages the storage in the *FilterLog* and *ShortTermLog* tables for discarded and operational data. The second computes and stores daily statistics based on predefined combination of the characteristics of the tag. The next layer applies a predefined aggregation functions to these results to compute weekly, monthly and yearly statistics. Finally, a last layer combines the statistics results with additional reference information and transforms those figures into graphical representations on a display module. The graphs (example: Fig. 1) allow user inspection of the data. TIM specialists are mostly interested in the global service statistics while the equipment specialists and CCC operators are mainly concerned by the data acquisition module figures and possible abnormal behaviour. The global statistics combined together with tag statistics allow the detection and correction of anomalies in the monitored data.

This off-line analysis has some drawbacks: we cannot analyse data as quickly as it arrives and the time of correction depends on the human ability of recognizing the anomalies. For this reason we plan to replace the current statistics module by a *Complex Event Processing* engine. The CEP technology is created for Big Data processing, and allows real-time analytics of incoming data streams. This will make up to date statistics immediately available to the TIM system, which can then

detect abnormal situations, generate warnings or alarms, and directly trigger corrective action.

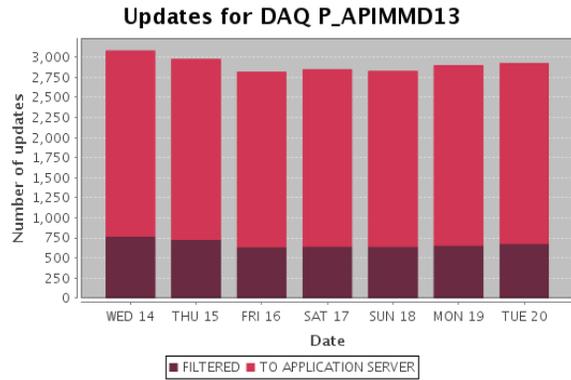


Figure 1: DAQ statistics.

RAPID DATA ACCESS AND ANALYSIS

To make good operational decisions, rapid access to the necessary data is of course critical. For monitoring systems, this usually means immediate notifications of changes and access to the latest values for all monitored data points. But increasingly, integration of historical data into real-time decisions is expected, for instance by monitoring a sliding average for a particular value, rather than the value itself. What's more, although less time-critical, users value the ability to replay historical events for understanding events or training purposes. TIM addresses these requirements by combining in-memory caching and database access, both of which we outline below.

Server Data Analyses

A number of factors affect the performance of data access. Chief amongst these are the number of independent actors performing read/write access, the consistency and persistence constraints applied on the data, and finally the chosen storage structure and medium. With the increasing number of storage options available today, it is important to choose the correct product for a given scenario. Whilst relational databases continue to offer the strongest durability and transactional guarantees, many requirements may be able to compromise on these and benefit from a much better search performance for instance. What's more, combining these technologies in a tailored design may prove the best solution, as is done in the traditional database-behind-cache architecture, combining an in-memory storage for rapid read access with a database responsible for the long term persistence of the data. To make matters more complex, most modern requirements integrate some distribution of the system over several nodes, either for scalability or availability.

We turn now to the C²MON platform on which TIM is based and describe how the architecture combines database and cache. The latest data point values of all operational data that reach the TIM server are stored in-

memory, as opposed to historical data which is so far only stored in the database. The in-memory storage allows for rapid client access to the current status of the monitored equipment, for initializing synoptic displays for instance, and equally rapid computation of business rules. Technically, the in-memory store makes use of the Java caching product Ehcache [3]. As well as being one of the best-documented caching APIs available in the Java world, this cache comes with two essential functionalities, justifying its integration into the C²MON platform. First, it can be distributed across multiple nodes, using the Terracotta [4] technology provided by the same company: the cache is transparently available from any number of C²MON instances and remains at all times consistent. This distribution means the latest data remains accessible, even in the case of a single node failure. Secondly, Ehcache was one of the first Java caches to store the cache off-heap, using the proprietary BigMemory technology. This feature allows for in-memory caches of arbitrary large size contrary to Java, where any on-heap memory is limited by the capacity of the Java garbage collector, in essence a few gigabytes; (garbage collection of off-heap memory must be managed by the application itself). In practice, this off-heap storage removes the limit on the amount of monitoring data that can be stored in-memory. Although currently we use this technology to process only real-time data we plan in the near future to make use of it to accelerate access and analysis of historical and DAQ filtered data.

Short Term Log and History Player

As mentioned in the previous paragraph, all data are persisted to the TIMDB database and kept for 30 days. As with the DAQ filtering module (chapter 2), these historical data are analysed to detect any anomalous behaviour.

Detailed analyses of historical data are available through other tools such as trend views and the history player. The TIM History Player (Fig. 2) can replay all events from any moment within the last 30 days to animate visual applications for the chosen timeframe. We can also save events over a selected timeframe as a named set, so that they can be replayed at any time in the future. This tool allows CCC operators to replay situations which need to be better understood.

Having said that, we have to acknowledge that performances of current tools for history analysis are not fully optimised today. Despite the well-structured algorithms for fetching data there is still an inconvenient delay due to database access. The in-memory management solutions we outlined above would eliminate this delay by providing a database independent historical analysis tool.

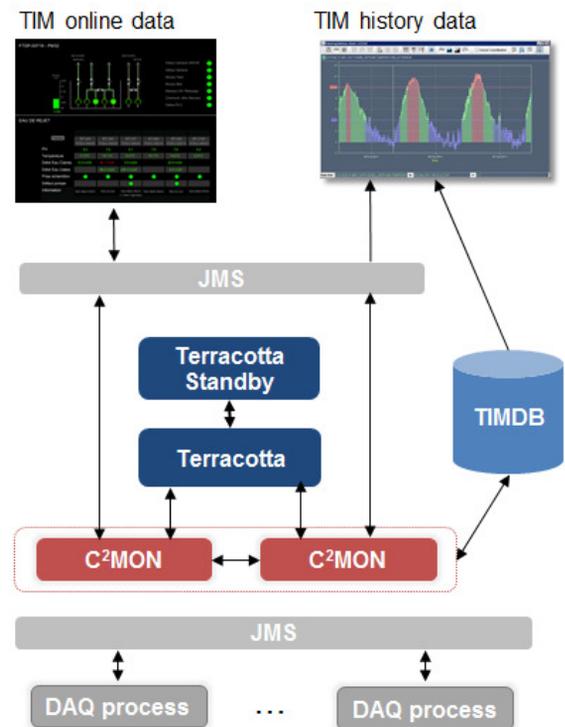


Figure 2: Access to operational and history data.

CONCLUSION

Monitoring complex heterogeneous systems is not only about transmitting data from the source to the client applications rapidly. Two independent but critical issues need to be addressed satisfactorily in order for a monitoring system to fulfil its duties: the integrity of the data that defines what is monitored and what must be done when certain events occur, and the way the monitoring system can protect itself from “pollution” by redundant or irrelevant data and data avalanches caused by malfunctioning components. The appropriate tools have been developed and implemented to satisfy these requirements, but they must constantly evolve and can be improved to keep up with technological change to provide the best possible service to users.

REFERENCES

- [1] M.Braeger, M.Brightwell, A.Lang, A.Suwalska, “A customizable platform for high-availability monitoring, control and data distribution at Cern” ICALEPCS,2011; <https://edms.cern.ch/document/1163395>
- [2] M.Brightwell “Data acquisition statistics and filtering mechanisms for technical infrastructure monitoring”, EPFL, 2009; <https://edms.cern.ch/document/1018821>
- [3] <http://www.ehcache.org>
- [4] <http://terracotta.org/>