EFDA–JET–CP(09)04/08

R. Castro, K. Kneupner, J. Vega, G. De Arcas, J.M. López, K. Purahoo, A. Murari,
A. Fonseca, A. Pereira, A. Portas and JET EFDA contributors

Real-Time Remote Diagnostic Monitoring Test-Bed in JET

Real-Time Remote Diagnostic Monitoring Test-Bed in JET

R. Castro[1], K. Kneupner[2], J. Vega[1], G. De Arcas[3], J.M. López[3], K. Purahoo[2], A. Murari[4], A. Fonseca[5], A. Pereira[1], A. Portas[1] and JET EFDA contributors*

JET-EFDA, Culham Science Centre, OX14 3DB, Abingdon, UK

[1]*Asociación EURATOM/CIEMAT para Fusión. Madrid, Spain*
[2]*EURATOM-UKAEA Fusion Association, Culham Science Centre, OX14 3DB, Abingdon, OXON, UK*
[3]*Universidad Politécnica de Madrid. Grupo I2A2. Madrid, Spain.*
[4]*Associazione EURATOM-ENEA per la Fusione, Consorzio RFX, 4-35127 Padova, Italy*
[5]*Associação EURATOM/IST, Instituto de Plasmas e Fusão Nuclear, Av. Rovisco Pais, 1049-001 Lisboa, Portugal*
** See annex of F. Romanelli et al, "Overview of JET Results",*
(Proc. 22[nd] IAEA Fusion Energy Conference, Geneva, Switzerland (2008)).

Preprint of Paper to be submitted for publication in Proceedings of the
7th Technical Meeting on Control, Data Acquisition and Remote Participation for Fusion Research,
Aix-en-Provence, France.
(15th June 2009 - 19th June 2009)

ABSTRACT.

Based on the remote experimentation concept oriented to long pulse shots, a test-bed system has been implemented in JET. It main functionality is the real-time monitoring, on remote, of a reflectometer diagnostic, to visualize different data outputs and status information. The architecture of the system is formed by: the data generator components, the data distribution system, an access control service, and the client applications. In the test-bed there is one data generator, which is the acquisition equipment associated with the reflectometer diagnostic that generates data and status information. The data distribution system has been implemented using a publishing-subscribing technology that receives data from data generators and redistributes them to client applications. And finally, for monitoring, a client application based on JAVA Web Start technology has been used.

There are three interesting results from this project. The first one is the analysis of different aspects (data formats, data frame rate, data resolution, etc) related with remote real-time diagnostic monitoring oriented to long pulse experiments. The second one is the definition and implementation of an architecture flexible enough to be applied to different types of data generated from other diagnostics, and that fits with remote access requirements. Finally, the third result is a secure system, taking into account internal networks and firewalls aspects of JET, and securing the access from remote users. For this last issue, PAPI technology has been used, enabling access control based on user attributes, enabling mobile users to monitor diagnostics in real-time, and enabling the integration of this service into the EFDA-Federation [1].

1. INTRODUCTION

The objective of the project described in this paper is to implement a system for monitoring on remote over Internet and in real time a KG8B correlation reflectometry system, which is currently located in JET. This means that a user can follow, using an Internet connection and in a realistic way, the evolution of the diagnostic during a pulse or part of it. To achieve this, a complete solution that covers the entire range from the diagnostic data output until the monitoring visualization has been designed and implemented taking advantage of CIEMAT experience in remote participation systems [2,3]. The result of the project is a flexible system in which other diagnostics can be easily integrated.

Additionally to the main objective, there is a set of properties that have been taken into account in the development:

Long pulse compatibility: The presented solution is ready to work in long pulse environments. This has been possible, on the one hand, using data streaming technologies that do not require a fixed experiment or data length, and, on the other hand, implementing stateless protocols and flexible components that enable both, services and users, to connect or to be restarted in the middle of a pulse.

Scalability: The solution has to be ready to support multiple concurrent clients monitoring a diagnostic at the same time. So, the solution must be scalable independently of the number of clients that will be connected.

Realism: The final system has to run in a real environment with limited resources at computing and communication levels, and it can therefore have delays in the remote monitoring process. Notwithstanding this problem, the solution must show a valid monitoring of the diagnostic, providing supplementary and useful information about the state of the monitoring process irrispective ofregarding the state of the experiment, and enabling to adjust different parameters in order to decrease the total delay with respect to the beginning of the shot.

Security: The final system must implement security solutions that enable complete access control mechanisms based, for example, on: user roles, time ranges, projects participation, etc.

2. THE DIAGNOSTIC TO MONITOR

JET's KG8B correlation reflectometry systems are composed of four reflectometers, resulting in sixteen signals that have to be recorded during the discharge. Correlation reflectometry is known to present strong variations in the bandwidth of the useful signal, depending on the plasma scenario and turbulence level. This diagnostic would benefit from a DAQ architecture based on adaptive sampling , which consists of acquiring data with a variable sampling rate which is continuously adapted depending on the bandwidth of the input signals , therefore optimizing the volume of data generated without loosing any information.

The heart of the system is the software application running in the controller that: provides a user interface to setup all acquisition parameters, acquires the data, adjusts the sample rate depending on the signal's bandwidth evolution during the experiment, and stores the results in the external storage system and sends it to the remote participation system.

3. SOLUTION DESIGN

One of the ideas in the design of the solution is to develop a system as much open and flexible as possible to simplify the integration of new diagnostics. In accordance with this idea, the system is based on a distributed architecture, so all the components can be placed in different network locations, and all the connections are based on IP protocol.

In the presented solution, as shown in Fig 1, when a new shot is detected, the diagnostic connects when a new shot starts streaming data to a data distribution component that, in real time, redistributes the received data to connected monitoring applications. The idea is to use a publish-subscription mechanism that will enable, in a simple way, to extend the functionality of the system to multiple diagnostics and multiple clients.

3.1 DATA FRAME

In case of feedback control for long pulses there are two important requirements that have to be taken into account. The first one is that data has to be managed or visualized during the experiment so it is not possible to organise data in structures related entire shots and it is necessary to manage data structures which are continuously generated. The second requirement is the possibility that any component can be connected or restarted in the middle of an experiment, so the data structures have to be coherent enough for this new started component to start working without waiting for a new shot.

In our design, the frame concept satisfies both requirements, defining a data structure that is coherent, can be processed as a unit without additional frames and is able to encapsulate online data that is continuously generated and can be streamed.

Another very closed concept, which is managed in the system, is the "frame rate", which is the number of frames per second that are distributed into the system.

3.2 DATA RESOLUTION

Since diagnostics can generate very large amounts of data , in the case of DiagKG8B up to 2MBsample/s, the limits of the networks and computers involved in all the monitoring process make it mandatory to control the amount of data per second that is managed. Data resolution is a concept that represents the ratio between the totaltotal data that is managed devided bandy the total data generated or received. In the case of theRegarding monitoring data generator, in this case KG8B reflectometer, data resolution is the ratio between the data sent for monitoring and the total amount generated by the diagnostic. In the case of theRegarding monitoring data visualizer, data resolution is the ratio between the data visualized and the total data received.

In the presented solution, the concept of data resolution in the data source is implemented into the data monitoring protocol, and is fixed by the diagnostic. Changing this parameter it is possible to control delays related with the network or with the data distribution. In the case of the monitoring application it is possible to change the resolution of data visualization; minimizing delays related with monitoring application.

3.3 DATA MONITORING PROTOCOL

The protocol is completely oriented to data streaming, so all the data is structured in frames. And the information included in every frame is coherent enough for enabling a monitoring application to start in the middle of a shot. The frame structure has been designed in a modular way to separate general monitoring data from data specific to KG8B reflectometer monitoring. The objective of this division is to develop a robust protocol that can also bealso be applied to the monitoring of other diagnostics.

The header of the frame is composed of the name of the diagnostic, the shot number and the size of the frame.

Into the data monitoring part, the frame has been subdivided in four sections:

Time delays: Fields related with time stamps and time delays. They are commented in section 5.

Shot Information: Fields related with current shot information: starting time of the experiment, estimated duration, date of the experiment, and time of the experiment.

Data Distribution Parameters: Fields related with data distribution configuration: resolution, and frame rate. Resolution is a generic parameter with values between 0 and 100. Its objective is to give to clients an estimation of the data monitoring resolution. The value is calculated by the diagnostic system depending on the monitoring data type.

State: Fields related with the state of the diagnostic: acquisition state and status error if exists.

In the case of KG8B reflectometerreflectometer ,the part of the frame reserved for specific diagnostic data is composed of the following fields: sample rate, number of channels, name of he channels, frequency range, block data size, power spectrums, and bandwidth evolution.

4. SYSTEM ARCHITECTURE

The solution is based on a distributed architecture over IP protocols. This structure improves the scalability and modularity of the solution, and simplifies the integration of new technologies. For example, it is very simple to integrate and test new data distribution technologies without changing the rest of the system.

Other important characteristic of this system is that almost all their components are implemented in JAVA and can be ported to many different platforms without recompiling. Only the diagnostic data module is not JAVA.

The system is composed, as it is showed in the Fig.2, by three subsystems whose implementation is going to be explained in the following points.

4.1 DIAGNOSTIC DATA MODULE

This module has been implemented in LabView for KG8B refelectometerreflectometer and its main function is to get processed data (previously commented in section 2) from the diagnostic, to change the resolution of the input data and generate monitoring data, to encapsulate it into frame structures of the data monitoring protocol (previously commented in point 3), and to send it to the corresponding data distribution server.

The implementation of this module in LabView is based on parallel tasks (one per each described phase) connected by intermediate buffers.

At connection level, this module tries to connect to the data distribution server when a new shot starts.

4.2 DATA DISTRIBUTION SERVER

This service is implemented based on the publisher-subscriber concept but using a technology that is oriented to data streaming solutions instead of asynchronous messaging solutions.

In the presented architecture the data distribution service is implemented using "Ring Buffering Network Bus" (RBNB) that is an open source implementation of the ring buffering technology, and it is involved in the data turbine initiative . The ring buffering is a data buffer structure with different storage levels. There is a buffer in RAM memory that is sequentially filled with the frames that continuously are received, and when the buffer gets its limit, the filling process start again from the beginning of the buffer, and the frames previously stored in that positions are sent to files. All this process is completely optimized for data streaming. There are two types of clients in RBNB: sources and sinks. A source sends data to a channel previously registered in the server and a sink gets data from requested channels.

This implementation of the data distribution service supports channels mirroring and routing between servers. Thanks to this, it provides scalability to the solution independently of the number of clients.

4.3 MONITORING APPLICATION

The application is implemented in JAVA using a three layers design, and Java Web Start technology as application distribution solution and start up mechanism. The application structure is based on a modular concept for being able to integrate other diagnostics. Additionally, specific classes have been developed for interacting with the data distribution service, and decoupling the rest of the application of this function. It simplifies the integration of other data distribution technologies without modifying the rest of the application.

The graphical interface shown in the Fig.3 is divided in monitoring information panel and diagnostic panel. This division is based on the idea of rendering this solution applicable to other diagnostics. In the first panel is shown the information related to: the status of the diagnostic, information of the current shot, monitoring delay respect the experiment and data distribution parameters, is displayed. In the second panel, specific information of KG8B reflectometry such as power spectrum and bandwidth evolution graphs, are displayed. Integrated into the application there is a "Delays Screen" that shows graphs related to the evolution of different delays along the shot. In section 5 a detailed description of these delays is provided.

From the architecture point of view, the classes of the application can be grouped, based on their functionality, in three groups:

Connection and data fetch: Its objective is to manage the connection to the server (authentication included) and to get from the data distribution server the frames of the diagnostic to monitor.

Delay calculation: It s objective is to calculate the different types of delays to monitor, based on the received frames, and to print this new data in the delay evolution graphs which are located into the "Delays Screen".

Monitoring information panel display: Its objective is to display visually information related with monitoring in the monitoring information panel, previously described.

Diagnostic panel display: Its objective is to display the information of KG8B reflectometry in the diagnostic panel.

The objective of this modularization of the internal structure of the application is (once more) to do it flexible enough to integrate mew diagnostics or new data distribution technologies.

5. DELAYS

A very important objective in this development is the realism of the monitoring solution. One of the most important elements for a remote user, who is following the evolution of a diagnostic in real time, is the information about the state of the monitoring process with respect the state of the real shot. In the implemented solution, the data, in its route fromroute from the diagnostic until the monitoring visualization, goegoes through four main processes and two network connections. Based on this and taking into account the points of the system where time delays can be measured, four delays have been calculated:

Delay of Processing: delay produced in the pre-processing data phase respect of the real experiment time (called acquisition time in the system). This phase is an internal diagnostic process.

Delay of ReadyToSend: delay produced between the end of pre-processing phase and the end of frame formatting process, where data is ready to be sent.

Delay of Sending: delay produced between the ready to be sent point in the diagnostic and the reception of frames point in the data distribution server. This delay includes the network connection between the diagnostic and the data distribution server.

Delay of Distributing: delay produced between the frame reception point in the data distribution server and the frame reception point in the monitoring application. This delay includes the data distribution delay plus the delay of the network connection between data distribution server and monitoring application.

Ideally, the optimum way of measuring these delays would be with synchronized clocks at all the measurement points. Unfortunately, this is not possible because of the heterogenic nature of the components and their different locations. This is why delays are calculated as relative differences. All the delays are calculated using delays of previous processes with respect to the beginning of the discharge, which is used as reference.

The three delays graphs of the "Delays Screen" show the temporal evolution of: every of the four described delays from the beginning of the shot, the instant delays of every type in a second, and the total delay of the monitoring respect the total shot time.

It is important to remark that the four previously described delays delays have been included into the data monitoring protocol in a way that, if a monitoring application is started in the middle of a shot, this application is going to have complete information about the total delay of the monitoring state respect the shot state. This includes a strong requirement in the system: "The diagnostic must be switched on when the shot starts and it is not possible to restarted in the middle". This requirement could be avoided if there were synchronization time signals in JET along the shot.

6. SECURITY

From the security point of view, the system is compatible with EFDA Federation and its security access mechanism is implemented in PAPI. Taking into account that PAPI is HTTP protocol oriented, a new facility has been included into PAPI to integrate not HTTP services. This new facility is based on token service concept, which is widely used in Kerberos technology. The idea is request to a PAPI controlled service for an encrypted service token element.

The way this mechanism works is: first a client application requests to connect to a protected service. Then, before opening theopening the connection, a service token request is sent to a PAPI component. If the user is authorized the application receives an encrypted service token (only valid for the service it is going to connect to). Finally, the application connects to the service and sends the service token to authorize the connection.

This new feature opens a range of applications of PAPI and increases the number of services that can be integrated into EFDA Federation.

CONCLUSIONS

The remote diagnostic monitoring system has been successfully tested achieving the objectives of monitoring in real time JET KG8B reflectometer . reflectometer. The implemented solution has been integrated into PAPI authentication and authorization technology rendering it compatible with the current EFDA federation infrastructure, and enabling federated users to monitor the diagnostic on remote.

The use of RBNB implementation of the ring buffering technology as data distribution server has provided very good results, giving low delay levels even with high input data flows. A basic data distribution functionality of this technology has been used in this test-bed, but it has very interesting additional features that would increase the functionality of monitoring systems for long pulse environments.

The developed system is very flexible and can teherforetherefore be extended in a simple way to other diagnostics and can integrate other data distribution solutions.

REFERENCES

[1]. R. Castro, J. Vega, A. Portas, D.R. López, et al. “ PAPI based federation as a test-bed for a common security infrastructure in EFDA sites”. Fusion Engineering and Design. Volume **83**, April 2008, Pages 486-490

[2]. J. Vega, E. Sánchez, A. Portas, et al. “ Overview of the TJ-II remote participation system”. Fusion Engineering and Design, Volume 81, Issues 15-17, July 2006, Pages 2045-2050

[3]. J. Vega, E. Sánchez, A. Portas, A. Pereira, et al. “ TJ-II operation tracking from Cadarache”. Fusion Science and Technology. Volume **50** · Nº 3 · (2006) · Pages 464-471

[4]. E. Barrera et al. “PXI-based Architecture for Real Time Data Acquisition and Distributed Dynamic Data Processing”. IEEE TNS 53-3 (2006) 923-926

[5]. G. de Arcas, J.M. López, M. Ruiz, E. Barrera, J. Vega, et al. “Self-adaptive sampling rate data acquisition in JET’s correlation reflectometer” Rev. Sci. Instrum. **79**, 10F336 (2008)

[6]. S. Tilak, P. Hubbard, M. Miller, T. Fountain. "The Ring Buffer Network Bus (RBNB) DataTurbine Streaming Data Middleware for Environmental Observing Systems". e-Science and Grid Computing, IEEE International Conference on. Volume , Issue , **10-13** Dec. 2007 Page(s):125 - 133

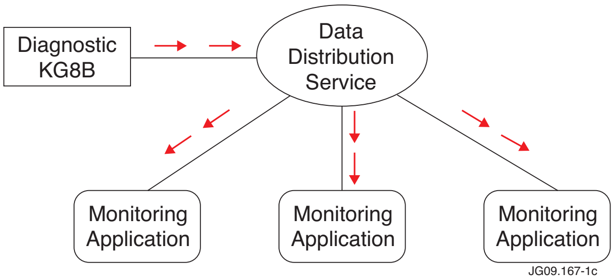

Figure 1: Design of the solution in which data is sent from the diagnostic to the data distribution server that redistributes it to clients.

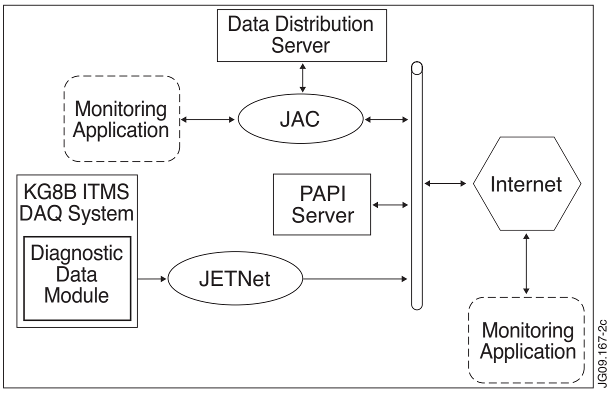

Figure 2: Connection schema of the different components of the system.

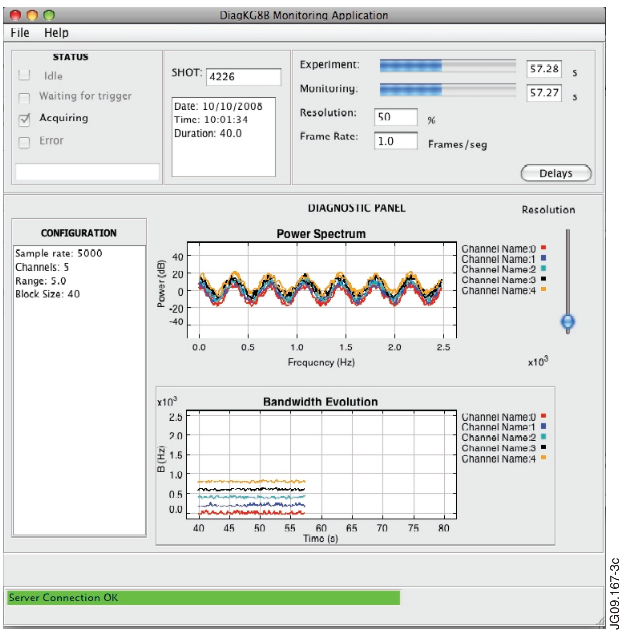

Figure 3: Monitoring application, in which monitoring information panel and diagnostic panel can be clearly distinguish.

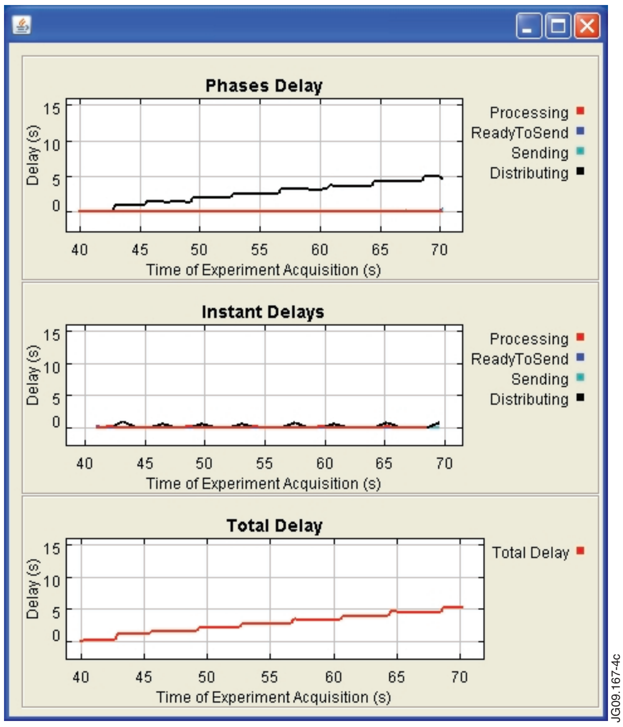

Figure 4: "Delays screen" with the delays graph, instant delays graph and total delay graph.