Vessel thermal map real-time system for the JET tokamak

D. Alves,[1] R. Felton,[2] S. Jachmich,[3,4] P. Lomas,[2] P. McCullen,[2] A. Neto,[1] D. F. Valcárcel,[1] G. Arnoux,[2] P. Card,[2] S. Devaux,[2] A. Goodyear,[2] D. Kinna,[2] A. Stephen,[2] and K.-D. Zastrow[2]

[1]*Associação EURATOM/IST, Instituto Superior Técnico, Instituto de Plasmas e Fusão Nuclear, Laboratório Associado, P-1049-001, Lisboa, Portugal*
[2]*EURATOM-CCFE Fusion Association, Culham Science Centre, Abingdon OX14 3EA, United Kingdom*
[3]*Laboratory for Plasma Physics, Ecole Royale Militaire/Koninklijke Militaire School, EURATOM-Association "Belgian State", Brussels, Belgium, Partner in the Trilateral Euregio Cluster (TEC)*
[4]*EFDA-CSU, Culham Science Centre, OX14 3DB, Abingdon, United Kingdom*

(Received 13 January 2012; published 24 May 2012)

The installation of international thermonuclear experimental reactor-relevant materials for the plasma facing components (PFCs) in the Joint European Torus (JET) is expected to have a strong impact on the operation and protection of the experiment. In particular, the use of all-beryllium tiles, which deteriorate at a substantially lower temperature than the formerly installed carbon fiber composite tiles, imposes strict thermal restrictions on the PFCs during operation. Prompt and precise responses are therefore required whenever anomalous temperatures are detected. The new vessel thermal map real-time application collects the temperature measurements provided by dedicated pyrometers and infrared cameras, groups them according to spatial location and probable offending heat source, and raises alarms that will trigger appropriate protective responses. In the context of the JET global scheme for the protection of the new wall, the system is required to run on a 10 ms cycle communicating with other systems through the real-time data network. In order to meet these requirements a commercial off-the-shelf solution has been adopted based on standard x86 multicore technology. Linux and the multithreaded application real-time executor (MARTe) software framework were respectively the operating system of choice and the real-time framework used to build the application. This paper presents an overview of the system with particular technical focus on the configuration of its real-time capability and the benefits of the modular development approach and advanced tools provided by the MARTe framework.

DOI: 10.1103/PhysRevSTAB.15.054701 PACS numbers: 28.41.My, 52.55.Fa

I. INTRODUCTION

As the fusion scientific community steers their efforts towards the operation of the international thermonuclear experimental reactor (ITER) tokamak, Joint European Torus (JET) aims to provide an important contribution as it will operate with similar plasma facing component (PFC) materials; in particular, demonstrating the predicted reduction of the tritium retention levels when compared with the previous carbon fiber composite (CFC)-based wall [1]. JET's new all metal wall surface made of solid beryllium, tungsten, and tungsten coated CFC is much less robust than the previous one, therefore raising serious challenges in high power operational conditions.

The protection of the ITER-like wall project was launched with the aim of providing the necessary tools to ensure the integrity of the vessel during JET's scientific campaigns. These include a set of 13 infrared (IR) cameras and 9 pyrometer diagnostics that, together with their real-time image processing systems [2], provide the temperatures of PFCs.

The vessel thermal map (VTM) collects these temperature measurements, groups them according to spatial location and probable offending heat source, and raises alarms that trigger the appropriate protective responses coordinated by the real-time protection sequencer [3] (RTPS) system, see Fig. 1. RTPS drives the local managers (LM) for all of JET's noninductive heating systems: lower hybrid

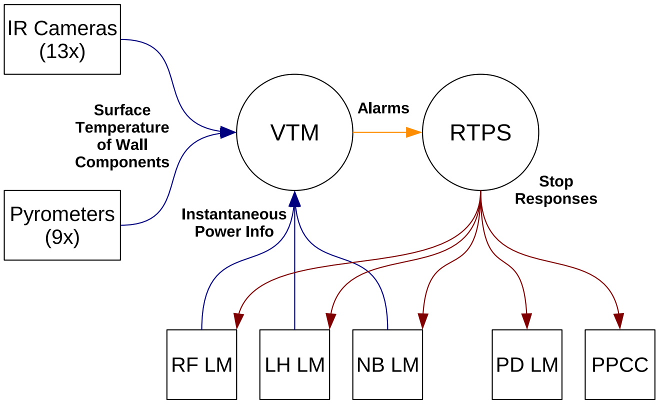

FIG. 1. VTM in the context of the wall protection project.

(LH), radio frequency (RF) and neutral beam (NB), the plasma density local manager (PDLM) and the plasma position, and current control (PPCC) system.

II. VTM FUNCTIONAL OVERVIEW

Temperature measurements of physical tile groups are processed by the VTM in macrosets called logical tiles (LTs). These macrosets take advantage of toroidal symmetry to establish the equivalence of temperature measurements that share the same poloidal position although taken at different toroidal locations. This feature allows for different camera views that, although not ‘‘looking at’’ particular physical tiles, monitor ‘‘equivalent’’ ones. Also, in case of measurement loss (e.g. when a camera fails in the middle of an experiment), VTM checks to see if the minimum amount of measurements for each LT is still fulfilled and, if so, the experiment proceeds without significant additional risk to the machine. After checking the validity of temperature measurements, VTM estimates LT temperatures as the maximum of all temperature measurements within the set.

At a higher level, sets of LTs are grouped into wall segments (WSs). Similar to the LT estimated temperature, the WS temperature is defined as the maximum of all LT temperatures within the set. Alarm triggering temperature thresholds are placed on WS estimated temperatures. VTM has the capability of processing a maximum number of 60 WSs (30 fixed and 30 user definable) and more than 400 tiles.

The system was specified to run at 100 Hz, twice the frequency of the real-time image processing units' temperature outputs and its real-time I/O was specified to be performed solely via the asynchronous transfer mode (ATM) based real-time data network (RTDN) [4].

III. HARDWARE

Because of its technical specifications, and after considering other options such as the VxWorks and VME or the RTAI and ATCA combinations, VTM was chosen to be implemented on a standard 4GB x86 multicore (6 cores) Linux-based PC with a PCI Gb ethernet network interface card (TRENDnet TEG PCITXR) for administration and a PCI ATM network interface card (FORE Systems PCA-200E) for real-time I/O. Despite JET's tradition of implementing the VxWorks and VME combination for real-time systems and the recent success of using the RTAI and ATCA combination for the new vertical stabilization system [5], the chosen path proved advantageous in terms of cost, development effort, debugging tools availability, and abundant support within the Linux and MARTe's community.

IV. SOFTWARE

The VTM system was built using the MARTe real-time framework [6] on a multicore vanilla (2.6.35.9) Linux platform. Real-time performance with a traditionally non-real-time operating system is achieved by configuring CPU isolation in terms of processes, threads, and interrupt request (IRQ) affinities, see Table I.

TABLE I. CPU affinities.

CPU #	Task
0	Linux & MARTe services
1	ATM IRQs
2	ATM synch Rx thread
3	ATM Rx threads
4	ATM Rx threads
5	Real-time thread

The synchronization of the VTM system with JET's central timing is done via the RTDN. Central timing is read by the real-time general services (RTGS) system directly from a VME electronics module and made available to the network. VTM not only time stamps its internal data using this clock but also triggers its own ‘‘control’’ cycle on the arrival of this ATM packet. Analyzing VTM's cycle time, measured with the internal cpu clock, for 961 JET pulses corresponding to more than 7.5 million cycles, reveals a probability of less than 1% of having a jitter greater than 100 μs, i.e., 1% of the cycle time. This means that a worst case jitter of 50 μs in the RTGS system is propagated over the RTDN traversing 2 ATM switches, including a shared inter-switch link fiber, and giving the distribution shown in Fig. 2. Lab tests consistently revealed worst case jitters of less than 5 μs when synchronizing on the internal cpu clock.

Additional lab tests were performed to assess system performance using the popular real-time patch for the Linux kernel. The aim was to assess whether using the patch would make the system more deterministic, i.e., lower jitter in the real-time cycle. For this test

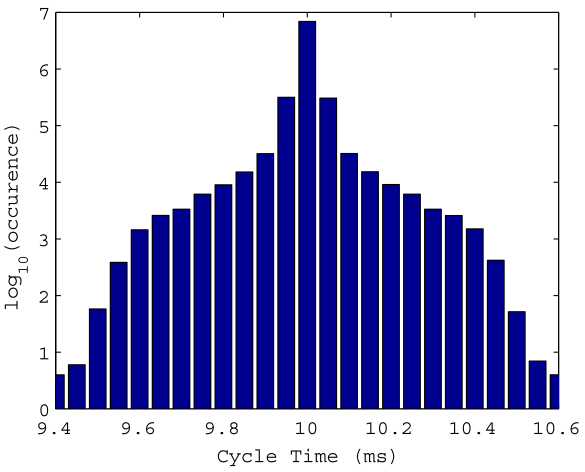

FIG. 2. VTM cycle time histogram.

the 2.6.33.7 kernel version was used and appropriately configured. Furthermore, the real-time priority of the IRQ and soft-irq ATM-related processes was elevated with respect to all other processes and threads running in the system. The cpu isolation mechanism and affinity distribution of threads and interrupts was kept exactly the same as in the case of using the plain Linux kernel. Both tests using synchronization on the ATM packet arrival and on the internal cpu clock revealed no measurable performance improvement of the patched kernel when compared to the plain one.

The VTM system receives a total of 19 ATM packets in real time (13 packets from cameras, 2 from pyrometers, 3 from additional heating systems plus the synchronization packet). A standard Linux ATM low-level driver kernel module is used together with a (user space) MARTe socket-based high-level driver implementation for receiving.

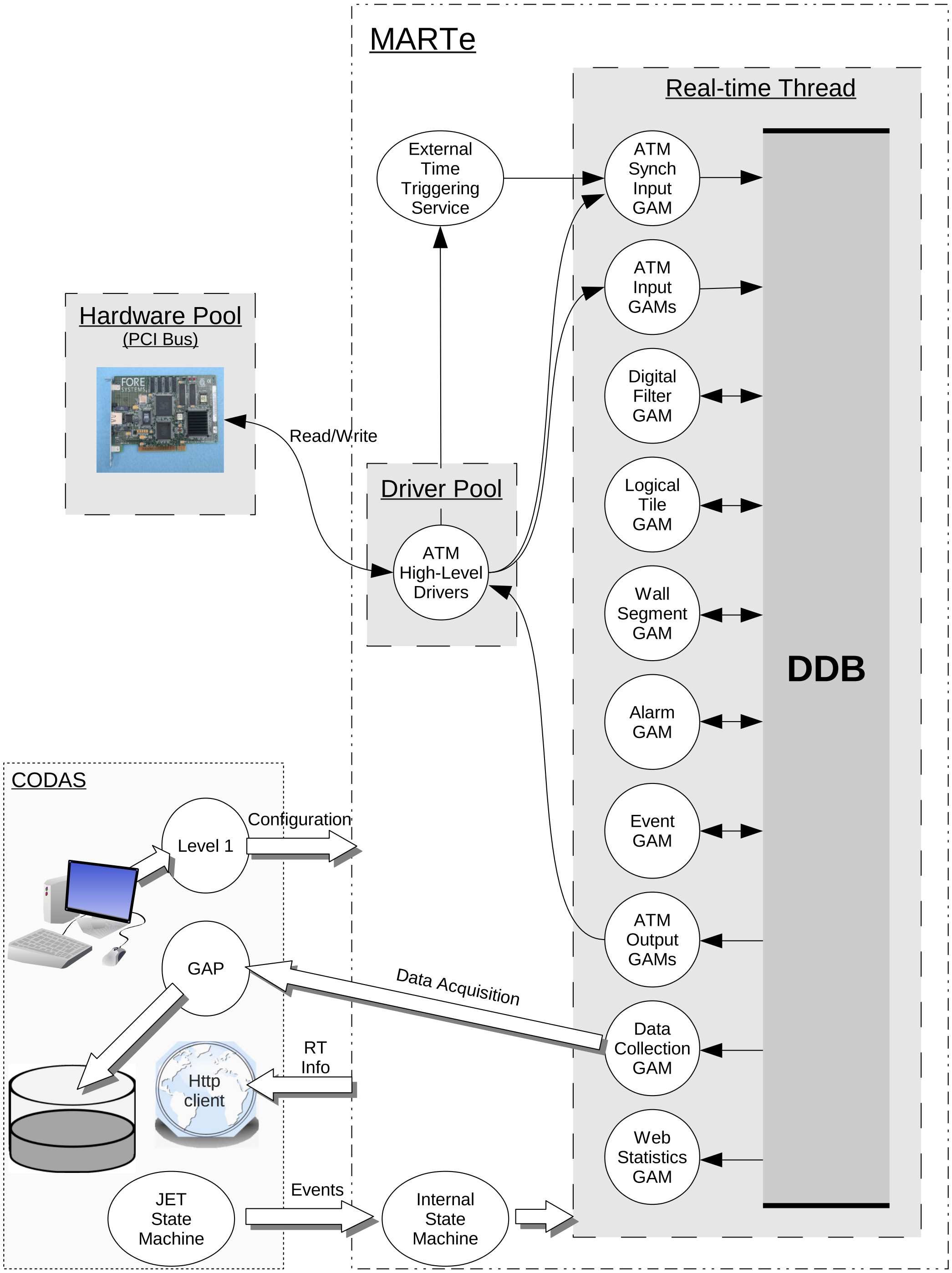

FIG. 3. VTM MARTe.

The system has been configured to boot remotely using the TFTP protocol and to mount its file system via NFS. In this way, independently of the hardware fault the system might experience, a straight hardware swap is sufficient to have the system up and running again without additional effort.

MARTe is a multiplatform framework for the development and deployment of data-driven, flexible and modular [7] real-time applications. It is based on a real-time oriented C++ library called BaseLib2. It currently supports the VxWorks [8], Linux [9,10], Linux/RTAI [5], Solaris, and Windows operating systems and has been increasingly adopted by the magnetic confinement fusion community [11]. Ideally, and assuming the required hardware components are supported, the developer is only required to write the pieces of software dealing with the logic/algorithmic aspects specific to the control application itself.

MARTe, from the user point of view, is primarily a sequential executor of generic application modules (GAMs) in a real-time priority context. MARTe also defines standard high-level interfaces for various activities including driver I/O, http display, and messaging. It provides a configurable internal state machine which, in the case of the VTM, is driven remotely by JET's global state machine. According to its current state, the appropriate list of GAMs is executed. Deeply embedded in MARTe's philosophy is the idea that GAMs should not be aware of one another and should perform specific and self-contained tasks communicating with each other exclusively by writing signals to and reading signals from the dynamic data buffer (DDB). This buffer is basically a signal pool with a standard access interface. MARTe also contains a driver pool of high-level interfaces for common (and/or maybe less common user implementable) activities, e.g., I/O. Finally, the external time triggering service (ETTS) is the entity that unleashes the execution of the GAMs. It can either be interrupt driven or polling a specific event. Figure 3 shows not only some of MARTe's architecture but also the specific VTM GAMs: (i) *ATM synch input GAM:*—is unblocked by the ETTS on the arrival of the ATM synch packet every 10 ms; (ii) *ATM input GAMs:*—make the ATM packets' data available as DDB signals; (iii) *digital filter GAM:*—filters DDB signals; (iv) *logical tile GAM:*—estimates LT temperatures based on the measurements; (v) *wall segment GAM:*—estimates WS temperatures based on LT temperatures; (vi) *alarm GAM:*—issues alarms based on WS temperatures and thresholds; (vii) *event GAM:*—records and displays both alarm events and the visual (ajax-based) thermal map of the machine interior; (viii) *ATM output GAMs:*—outputs

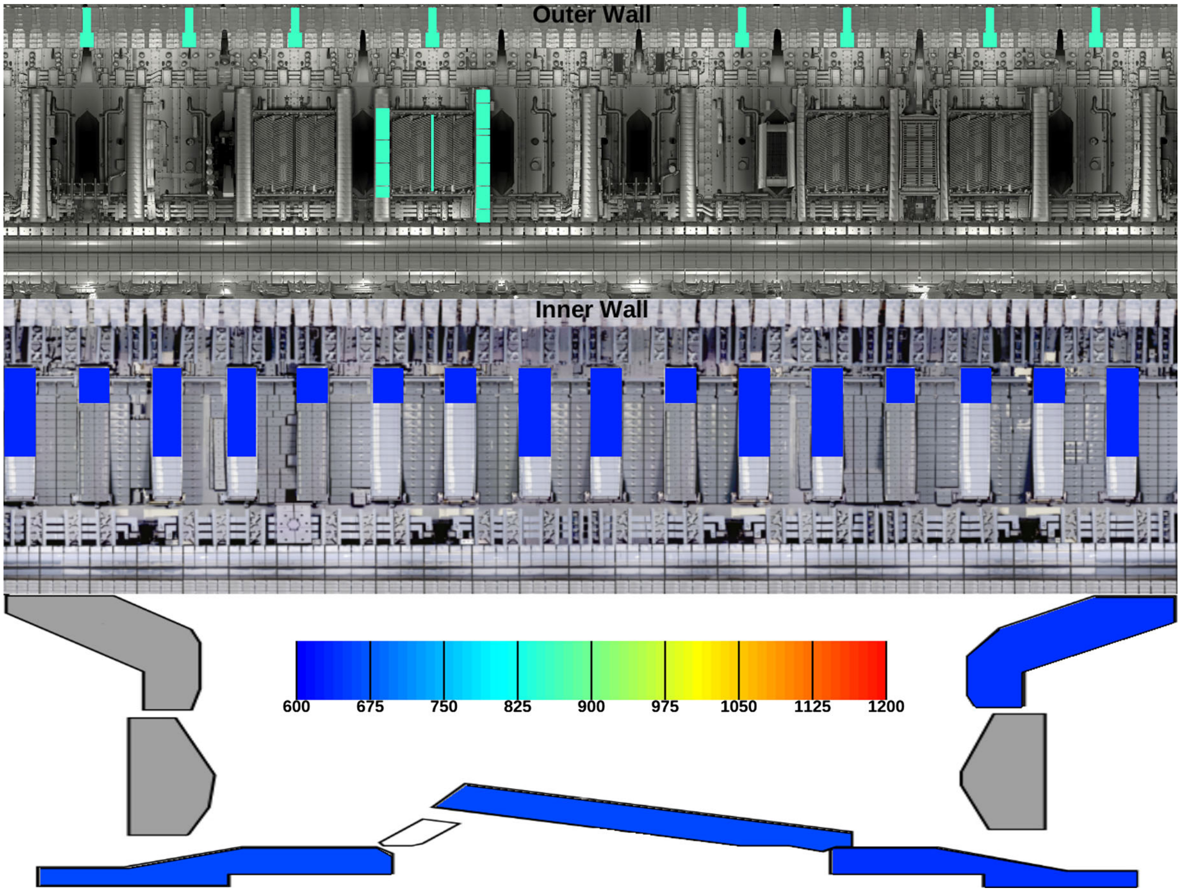

FIG. 4. Real-time temperature map display—http/ajax based polling @ 4 Hz without compromising real-time performance.

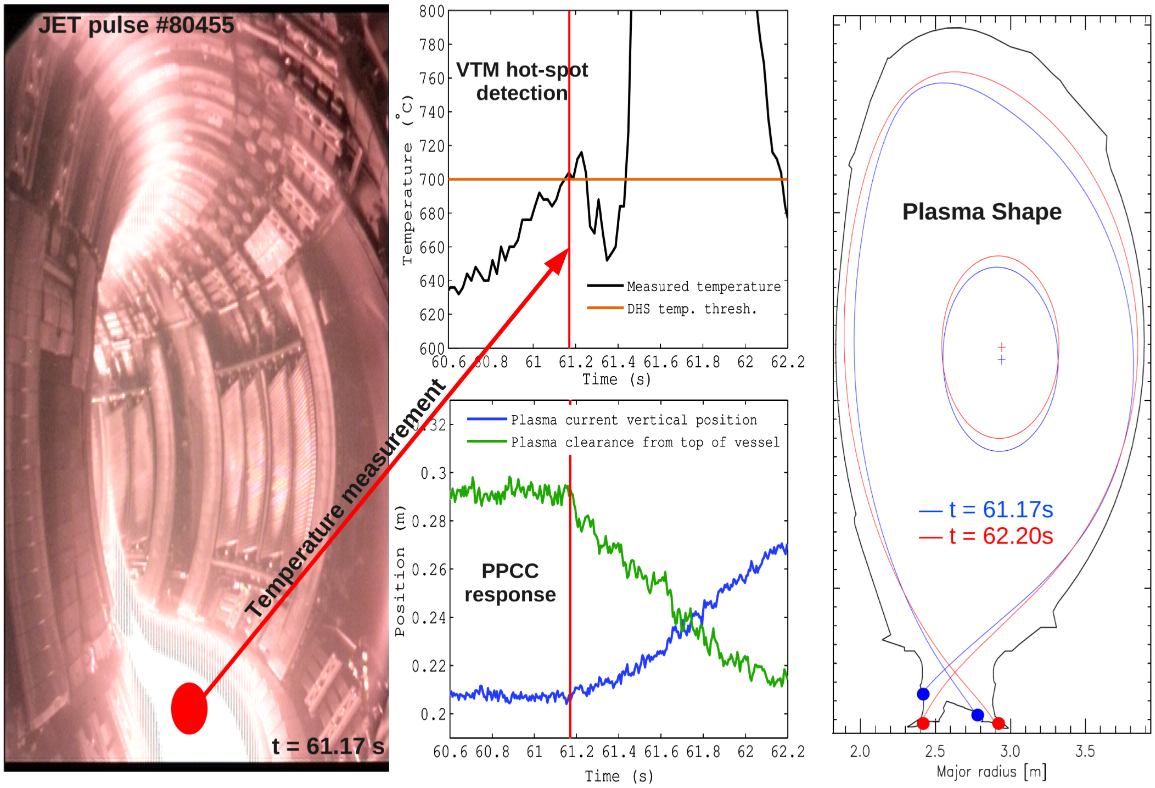

FIG. 5. Preliminary results—upon the detection of a thermal event the plasma is driven into a configuration that attempts to reduce the heat loads on the affected regions.

the alarms and WS temperatures to the RTDN; (ix) *data collection GAM:*—records DDB signals to be collected for posterity; (x) *web statistics GAM:*—http display of DDB signal statistics.

To provide a convenient and intuitive real-time display of the vessel temperature map, an Ajax-based http service (see Fig. 4) has been developed. Planar images of both JET's inner and outer wall plus the so-called divertor region are used as canvas for painting each LTs' temperature using the indicated color code. This service has been tested with a single browser client polling at 4 Hz and no impact was observed in the system's real-time performance.

V. PRELIMINARY RESULTS

As a proof of concept, dedicated plasma pulses were performed to test the end-to-end response of the wall protection apparatus. Simulated camera measurements based on visible light were provided to the VTM and reference pulses were used to place the VTM's alarm triggering temperature thresholds at convenient levels. Figure 5 illustrates the achievement of pulse 80455. A particular plasma configuration is being sustained until a thermal event is identified at 61.17 s. The shape of the plasma boundary being sustained is shown by the blue line of the contour plot on the right. Note the positions where the lines strike the bottom (so-called divertor) region of the machine. These are the points where the larger heat fluxes are deposited. At 61.17 s a divertor hot spot (DHS) is detected by the VTM (top middle plot) based on temperature measurements from a camera (image on the left). Upon the detection of the hot spot, protective action is performed in order to drive the plasma into a configuration that will attempt to mitigate a further temperature rise. The transition into a different plasma configuration can be seen in the bottom middle plot showing the plasma current's centroid moving upwards and the gap between the plasma boundary and the top of the vessel being reduced. Finally, at 62.20 s, the plasma is in a configuration (red curve on the contour plot on the right) where the strike points have moved to different positions therefore easing the heat load on the previous locations.

VI. CONCLUSIONS

The VTM has been demonstrated to perform the role it was designed for in the context of the wall protection project. The nontraditional configuration of its real-time capability, using a plain Linux vanilla kernel together with the user-space MARTe-based implementation, makes it an interesting proof that with little effort it is possible to have an application meeting real-time requirements. The fact that its Ajax-based visual temperature monitoring service, acting as a continuous additional source of interrupt and CPU loads, has no impact in the system's real-time performance takes this matter to an even higher level demonstrating the potential of the multicore approach.

This system is now fully integrated into the standard machine protection ensemble and is fundamental to JET operations.

ACKNOWLEDGMENTS

This work was supported by EURATOM and carried out within the framework of the European Fusion Development Agreement. The views and opinions expressed herein do not necessarily reflect those of the European Commission [12].

[1] G. Matthews *et al.*, "JET ITER-like Wall—Overview and experimental programme," in Proceedings of the 13th International Workshop on Plasma-Facing Materials and Components for Fusion Applications, Rosenheim, Germany.

[2] M. Jouve *et al.*, "Real-time protection of the ITER-like Wall at JET," at the 13th International Conference on Accelerator and Large Experimental Physics Control Systems (2011).

[3] A. V. Stephen *et al.*, "Centralised Coordinated Control To Protect The JET ITER-like Wall," at the 13th International Conference on Accelerator and Large Experimental Physics Control Systems (2011).

[4] R. Felton *et al.*, "Real-time plasma control at JET using an ATM network," at the 11th IEEE NPSS Real-Time Conference, Santa Fe, 1999.

[5] T. Bellizio, G. De Tommasi, P. McCullen, A. C. Neto, F. Piccolo, F. Sartori, R. Vitelli, and L. Zabeo, IEEE Trans. Plasma Sci. **38**, 2465 (2010).

[6] A. C. Neto, F. Sartori, F. Piccolo, R. Vitelli, G. De Tommasi, L. Zabeo, A. Barbalace, H. Fernandes, D. F. Valcarcel, and A. J. N. Batista, IEEE Trans. Nucl. Sci. **57**, 479 (2010).

[7] A. Neto *et al.*, "Exploitation of Modularity in the JET Tokamak Vertical Stabilization System," at the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), 2011.

[8] D. Alves *et al.*, Fusion Eng. Des. **86**, 1034 (2011).

[9] D. F. Valcárcel *et al.*, IEEE Trans. Nucl. Sci. **58**, 1490 (2011).

[10] P. Carvalho, P. Duarte, T. Pereira, R. Coelho, C. Silva, and H. Fernandes, IEEE Trans. Nucl. Sci. **58**, 1427 (2011).

[11] A. Neto *et al.*, IEEE Trans. Nucl. Sci. **58**, 1482 (2011).

[12] F. Romanelli *et al.*, in *Proceedings of the 22nd Fusion Energy Conference, Geneva, Switzerland, 2008* (IAEA, Vienna, 2008). [All members of the JET-EFDA Collaboration (JET-EFDA, Culham Science Centre, Abingdon, U.K.) appear in the appendix of this paper.]