Fusion Engineering and Design 128 (2018) 1–6

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

Implementation and applications of a new real-time framework for plasma control system in HL-2A

W. Zheng[a,b], F. Xia[c], C. Yang[a,b], M. Zhang[a,b,*], L.Y. Chen[c], G. Zhuang[a,b], Y. Pan[a,b]

[a] State Key Laboratory of Advanced Electromagnetic Engineering and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[b] School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[c] Southwestern Institute of Physics, Chengdu, Sichuan, China

ARTICLE INFO

Keywords:
Tokamak
Plasma control system
Real-Time framework
HL-2A

ABSTRACT

Plasma Control Systems (PCS) that contain various functions and algorithms are applied in different tokamak devices. A real-time framework which allows various control functions and algorithms to be developed and deployed is very important to a PCS. A modular-based Multi-thread Real-time Framework (MRTF) is developed by HL-2A for this purpose. A multi-thread scheduling solution is provided by the framework to run the modules in PCS, and also to synchronize the modules and make them shared with each other. CPU affinity has been made for every thread in MRTF. In order to integrate control modules, an abstraction interface for PCS functions is defined by the framework. Several tests have been made to assess the performance of the real-time framework. A plasma real-time equilibrium reconstruction module is developed and integrated in MRTF. The module can calculate the flux distribution of plasma profile by the grid of 129*129 within 700 us on HL-2A experiment. The framework has been implemented in HL-2A experiment and achieved expectations. More control functions and strategies of PCS will be integrated by the framework in the future.

1. Introduction

HL-2A [1] is the first tokamak with the divertor and elongated plasma cross-section in China. Nowadays, more and more control functions and algorithms are integrated in Plasma Control Systems (PCS) in HL-2A. That means a real-time framework which allows various real-time control functions to be developed and deployed more smoothly is required by the HL-2A PCS.

Several solutions have been adopted in tokamaks around the world. A modular real-time control software framework called the Multithreaded Application Real-Time executor (MARTe) [2–4] has been developed by JET. Also, a simple and light weight framework [5,6] for real-time control has developed in J-TEXT [7]. Inspired by those frameworks, a new Multi-thread Real-Time Framework (MRTF) for Plasma Control System is developed by HL-2A.

2. Requirement of HL-2A PCS

It should be noted that HL-2A PCS [8,9] is a component of the HL-2A control system. Fig. 1 demonstrates that the fast controllers include the diagnosis system, the PF system and the heating system. The Reflective Memory (RFM) is employed in the real-time data transmission between the above controllers. For the slow control of all the sub systems, a PLC network is implemented in HL-2A. In order to carry out the high performance computing, an Nvidia Tesla K20x GPU card is used in PCS. With the GPU, the PCS can accelerate parallel computing, such as real-time reconstruction [10].

In addition, HL-2A PCS requires high real-time performance, in particular, most of the control strategies in HL-2A PCS are carried out with 1 kHz. It means that a real-time deterministic performance by 1 ms is required for the MRTF. In order to implement and integrate the control strategies in HL-2A PCS, the MRTF will provide the expandability and portability.

3. Structure of MRTF

It should be noted that the MRTF is developed under Red Hat Enterprise Linux (RHEL), which complies with the ITER CODAC fast controller standard. The framework employs the abstraction and modular designs to integrate the control functions of PCS. Meanwhile, the multi-thread techniques are implemented to improve the real-time performance of the framework. The structure of the real-time framework is shown in Fig. 2. A multi-thread framework is developed in the top layer, which is responsible for the creation, management and

* Corresponding author at: State Key Laboratory of Advanced Electromagnetic Engineering and Technology, Huazhong University of Science and Technology, Wuhan 430074, China.
E-mail address: zhengwei@hust.edu.cn (M. Zhang).

https://doi.org/10.1016/j.fusengdes.2018.01.018
Received 3 November 2016; Received in revised form 24 December 2017; Accepted 3 January 2018
Available online 28 January 2018

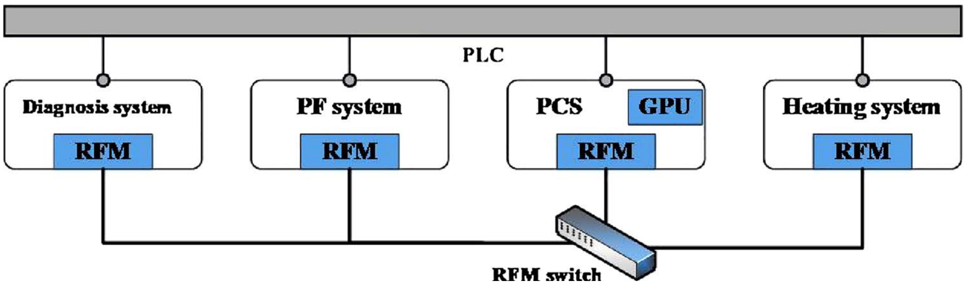

Fig. 1. The links between PCS and other system in HL-2A.

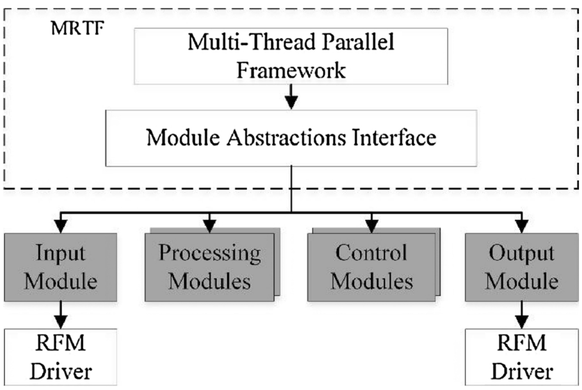

Fig. 2. Definition of the structure for the real-time framework.

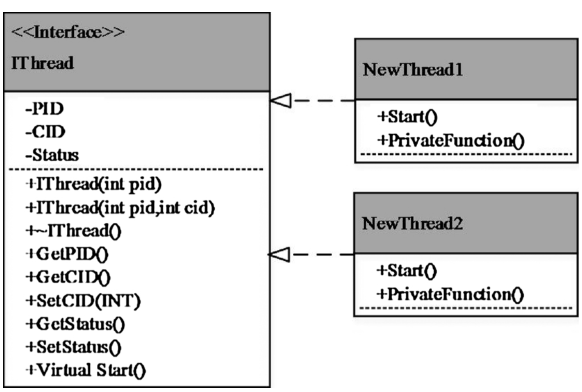

Fig. 3. The abstract interface for real-time framework.

synchronization of application threads. To uniform the modules in PCS, an abstraction interface is designed. In addition, several modules have been defined by the interface, which includes the input module, output module, processing modules and controlling modules. It should be noted that the modules implementing the interface are invoked by the threads via the interface. To communicate with other systems in the tokamak, the RFM driver is integrated in the input/output module.

With the above structure, the real-time framework is flexible and expandable. In particular, a module can be added, removed and modified in the framework without changing other elements. The modules are reusable components that are running separately. The threads only need to invoke the abstraction interface without knowing anything about the implementation of the modules.

4. Implementation of MRTF

The implementation of MRTF consists of thread scheduling, abstraction design and modular integration. A solution of integrating the control functions of PCS is proposed in the framework.

4.1. Interface for threads

In this paper, an abstraction interface is proposed to define the common attributes and functions of the threads, which simplifies the creation and management of threads in MRTF. As shown in Fig. 3, the attributes of the interface contain thread ID, status, core ID, etc. At the same time, the functions involve the CPU affinity configure, and get/set the attributes. A virtual function named "start" is declared in the interface, which indicates that the threads inherited from the interface have to define the function to carry out the execution. New threads can be easily constructed and all of the threads have a uniform specification based on the interface.

The main thread and sub threads are developed using the MRTF with the interface described above. As shown in Fig. 4, the main thread is executed first, which will initialize the global variables and create other sub threads in sequence. After all of the threads are carried out, the main thread will monitor the status of each thread during the cycle.

4.2. Multi-threads CPU affinity

In order to improve the real-time deterministic performance of the threads in the MRTF, CPU affinity is set to the threads to assign them to different cores. A Linux system library (sched.h) is used in the function applied by the interface of threads as mentioned in Section 4.1. Threads can be simply assigned to cores in CPU. A strategy has been made to assign application threads of MRTF to cores. As shown in Fig. 5, non-real-time threads (main thread and configure thread) of MRTF are centrally assigned to core 1 in the CPU block. Real-time threads are assigned separately in the other cores in the CPU blocks. The input

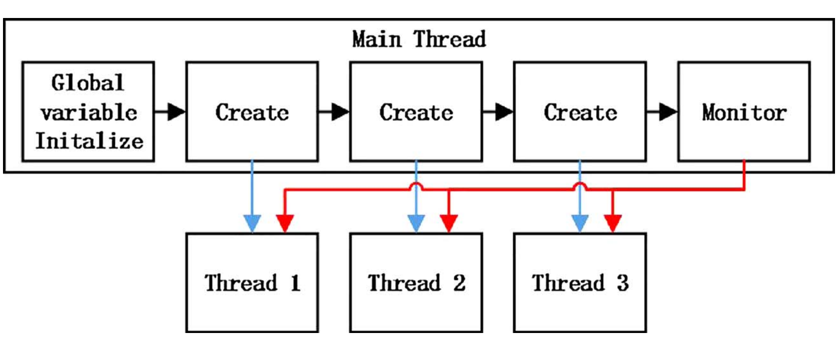

Fig. 4. The multi-threads creation in the MRTF.