Fusion Engineering and Design 128 (2018) 82–85

Contents lists available at ScienceDirect

Fusion Engineering and Design

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

Improvement of GPU parallel real-time equilibrium reconstruction for plasma control

Y. Huang[a], B.J. Xiao[a,b], Z.P. Luo[a,*], Q.P. Yuan[a]

[a] Institute of Plasma Physics, Chinese Academy of Sciences, Hefei, China
[b] School of Nuclear Science & Technology, University of Science & Technology of China, China

ARTICLE INFO

Keywords:
Equilibrium reconstruction
Plasma control
EAST
GPU parallel computation

ABSTRACT

Improvements of GPU parallel real-time equilibrium reconstruction code P-EFIT are presented. P-EFIT is based on the EFIT framework, but built with the CUDA™ architecture to take advantage of massively parallel Graphical Processing Unit(GPU) cores to significantly accelerate the computation. Newly designed architecture and applied technique make P-EFIT integrate into plasma control system(PCS) as a sub-algorithm. Efficient parallel strategy and algorithms allow P-EFIT to achieve good computational performance. P-EFIT can complete one equilibrium iteration and transfer real-time diagnostics, control signals and equilibrium data in 375us with 129 × 129 spatial grid. P-EFIT has high spatial resolution, customized modules and internal current profile reconstruction for real-time plasma control in EAST.

1. Introduction

Experimental equilibrium reconstruction provides information such as plasma boundary, current density and safety factor profiles for tokamak operation and research. EFIT reconstructs equilibrium by finding the Grad-Shafranov solution that is the least squares best fit to the experimental measurements [1]. It efficiently combines equilibrium and fitting iterations to search for an optimum solution and has been widely used in many tokamaks worldwide. However, the full algorithms of EFIT is computation intensive to be used in real-time control particularly for the high spatial grid resolution. Its real-time version, called RT-EFIT [2] is currently used in DIIID, EAST, KSTAR, NSTX and MAST [3–6]. Based on CPU OpenMP parallel computation, the code called GPEC [7] and LIUQE [8] have been developed and applied in ASDEX-U [9]. Additionally, a parallelized version EFIT code based on GPU for magnetic reconstructions, P-EFIT has been developed, which significantly accelerate the whole computation process [10]. P-EFIT efficiently takes advantage of massively parallel GPU cores and significantly accelerates the EFIT reconstruction algorithms and it has been successfully implemented for plasma control in EAST [11].

P-EFIT is based on the EFIT framework but instead of using a multi-core CPU platform, it takes advantage of massively parallel GPU cores to significantly accelerate the computation. The equilibrium reconstruction algorithm consists of several sequential middle-scale matrix multiplications which need to be iterated. Parallelizing these algorithms require many computational cores and a large amount of communication between cores. There are standard linear algebra routines in CUDA™ library, but the size of matrix in plasma equilibrium reconstruction algorithms is too small to achieve good performance by directly using them. When developing modules in P-EFIT, these customized parallel algorithms should be carefully designed with full consideration on the needs of numerical algorithms and the GPU capacity. Efficiently distributing hundreds of GPU cores and minimizing the communication among them are basic principle. Some optimizations for middle-scale matrix multiplication are described in [10]. A fast Grad-Shafranov solver based on eigenvalue decomposition solve the block tri-diagonal block linear system in parallel on the GPU as described in [12]. Efficient parallel strategies and algorithms allow P-EFIT to achieve good computational performance regardless of different current representations and diagnostics [13]. P-EFIT supports Multiple-Input Multiple-Output (MIMO) plasma shape control experiments [14,15] and includes POlarimeter-INTerferometer (POINT) diagnostic for real-time plasma current density and q profile reconstruction in EAST [13].

Advanced plasma control for EAST steady state operation requires more accurate and detailed real-time plasma equilibrium reconstruction. P-EFIT needs to be improved to have higher spatial resolution [16], more diagnostics and integration into PCS. Framework and algorithms in P-EFIT should be updated for universality and integration. By newly designed architecture and interface, PCS can directly manage P-EFIT which is integrated into PCS as a sub-algorithm, the equilibrium data and control signals are transferred through PCI-E. With the

* Corresponding author.
E-mail address: zhpluo@ipp.ac.cn (Z.P. Luo).

https://doi.org/10.1016/j.fusengdes.2018.01.043
Received 22 June 2017; Received in revised form 2 January 2018; Accepted 17 January 2018

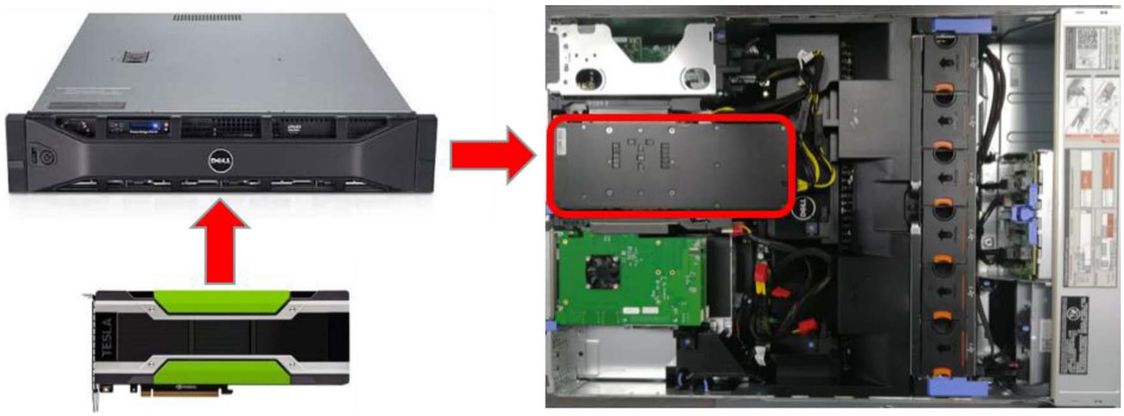

Fig. 1. The hardware architecture of integrating P-EFIT into PCS: installing Tesla P100 GPU into PCS real-time computing node.

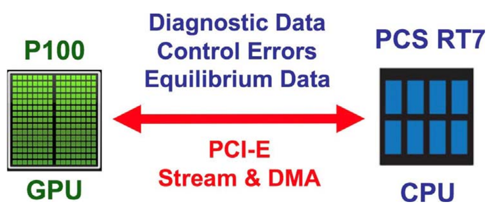

Fig. 2. The software architecture of integrating P-EFIT into PCS: PCS invokes P-EFIT directly, data is transferred between PCS and P-EFIT through PCI-E.

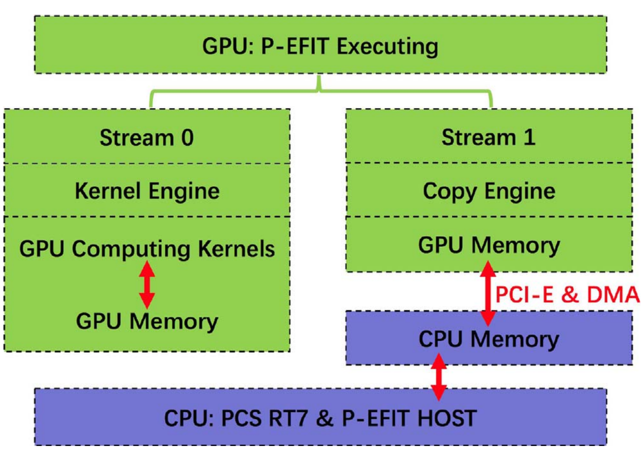

Fig. 3. Two streams are distributed on GPU, stream0 drives kernel engine for GPU parallel computation, stream1 drives copy engine for transferring data through PCI-E.

development of GPU hardware, optimization is carried out for P100 GPU. Robust parallel strategy and algorithms allow P-EFIT to achieve good computational performance, including equilibrium calculation, global parameters, geometry parameters, profile calculation and data transmission. After integrated in PCS, P-EFIT can complete one equilibrium iteration and transfer real-time diagnostics, control signals and equilibrium data in 375us with 129×129 spatial grid on P100 GPU. With all these improvements, P-EFIT can provide precise, detailed real-time plasma equilibrium reconstruction for sophisticated plasma control in EAST.

2. Improvement in P-EFIT algorithms and architecture

P-EFIT was an independent system to PCS, a reflective memory network (RFM) between them was built to share the real-time data [11]. However, this implementation has some disadvantages when extending P-EFIT's capacity to allow more applications in plasma control. Firstly, P-EFIT was integrated with PCS as a sub-system, the only communication method between two systems is through RFM network. When emergency situations occur, PCS cannot manage P-EFIT directly, it would make the systems unstable. A more reliable way is that integrating P-EFIT into PCS as a sub-algorithm, PCS can invoke and manage P-EFIT as a thread directly. Secondly, more applications in plasma control require much more data transferring in real-time, the data transmission speed of RFM cannot fulfill the requirement. After integrating P-EFIT into PCS as a sub-algorithm, PCI-E could be used in transferring mass of real-time data between the CPU and GPU. Lastly, more modules in P-EFIT need to be developed to meet the real-time control requirements.

2.1. Integrated into PCS as a sub-algorithm

EAST new PCS has one host and one real time computing node, the real time computing machine is Dell R730 server with two Intel Xeon E5-2667 eight-core CPUs with frequency 3.2 GHz on which 64-bit Linux Redhat 6.7 operation system is installed [17]. After installing Tesla P100 GPU hardware into PCS real-time computing node as shown in Fig. 1, the first challenging problem is how to make P-EFIT be invoked by PCS directly. EAST PCS is deployed by using C programming language and P-EFIT is deployed by using CUDA-C language [18]. Although CUDA-C language is the C language with new CUDA extension, it is hard to adjust either source code into the other's. The implementation is realized through compiling and wrapping P-EFIT source code into static library which could be linked when compiling PCS source code. The architecture is shown as Fig. 2, GPU processes equilibrium calculations, PCS distributes one CPU core called RT7 as P-EFIT's host. All the data is transferring between CPU and GPU's memory through PCI-E, the Stream and DMA techniques to accelerate data transmission speed will be introduced in the following sections. By this way, P-EFIT is integrated into PCS as a sub-algorithm which could be invoked and managed by PCS directly.

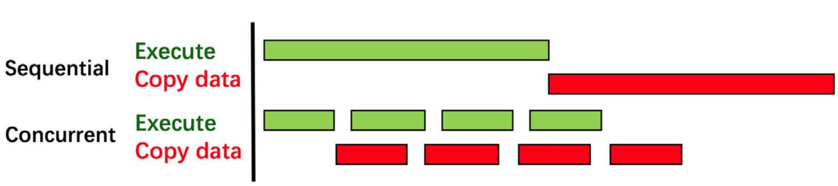

Fig. 4. Compared with sequential data transmission, concurrent data transmission can overlap most of data transmission and copy time.