Real-Time Outlier Detection Algorithm for Finding Blob-Filaments in Plasma

Lingfei Wu*, Kesheng Wu(Advisor)†, Alex Sim(Advisor)†, Andreas Stathopoulos(Advisor)*

*College of William and Mary, Williamsburg, Virginia, USA
†Lawrence Berkeley National laboratory, Berkeley, California, USA

I. Introduction

Magnetic fusion could be an inexhaustible, clean, and safe solution to the global energy needs. The success of magnetically-confined fusion reactors (like ITER [1]) demand steady-state plasma confinement which is challenged by the edge turbulence such as the blob-filaments. A blob-filament (or blob) is a magnetic-field-aligned plasma structure that appears near the edge of the confined plasma and has significant higher density and temperature than the surrounding plasma [2]. The blobs are important to study since they convect filaments of plasma outwards the containment wall, which results in massive heat loss from plasma, degradation of the magnetic confinement and erosion of the containment wall.

This work is motivated by several considerations. A magnetic fusion device operates in pulses, known as shots. A shot can typically last from a few seconds to hundreds which produce terabytes of data. There are three types of analysis in most of fusion experiments: in-shot-analysis, between-shot-analysis, and post-run-analysis. All existing blob detection methods address post-run-analysis challenges, but in this work, we focus on the more challenging first two cases to provide a real-time analysis so that the scientists can monitor the progress of fusion experiments and prevent catastrophic events. Another motivation is to integrate this work into the in transit data analysis framework over wide-area network for oceans part collaborators, named ICEE [6]. In this system, our blob detection algorithm is served to monitor the health of the fusion experiments in real time at Korea Superconducting Tokamak Advanced Research (KSTAR).

In this work, we propose a real-time outlier detection algorithm to efficiently find blobs in the fusion experiments or numerical simulations. The proposed algorithm is based on outlier detection with various criteria and a fast connected component labeling method to find blob components. We have implemented this algorithm with hybrid MPI/OpenMP and demonstrated the accuracy and efficiency of our implementation with a set of data from the XGC1 fusion simulations [7]. Our tests show that we can complete blob detection in a few milliseconds using a cluster at NERSC and achieve linear time speedup. We are integrating it into the ICEE system, and plan to test the algorithm in the KSTAR experiments. We also plan to develop a blob tracking algorithm based on this work.

II. Related Works

To study the impact of the size, movement and dynamics of blobs, various blob detection methods have been proposed to identify and track these structures. A plasma blob is most commonly determined by some threshold computed statistically in the local plasma density signal [2]. However, the exact criterion has varied from one experiment to another, which reflects the intrinsic variability and complexity of the blob structures. In [3], an image analysis for the identification of blobs has been presented based on a gas puff imaging (GPI) diagnostic images. Due to noise and lack of a ground truth image, this approach can be sensitive to the setting of parameters and hard to use generic method for all images. Recently, several researchers [4][5] have developed a blob-tracking algorithm that utilizes a contouring method and image analysis software on the raw GPI camera data. This method is close to our approach but not suitable for real-time blob detection since they compute time-averaged intensity to normalize the local intensity.

III. A real-time blob detection approach

A. *Outlier detection algorithm for finding blobs*

In this section, we illustrate our outlier detection algorithm to determine blobs to study their characteristics. A sequence of sample processed frames can be in situ raw data either from real experiments or from numerical simulations. The structure of the turbulence is analyzed by firstly normalizing the local density $I(r, z, t)$ to the initial time frame density $I(r, z, 1)$ and computing associated quantities in the regions of interests. The resulting triangular mesh is refined to a higher resolution one with 4 times more triangles. In order to apply a proper predefined quantile, it is advised to perform exploratory data analysis to explore the underlying distribution of the data sets. Using the best fitted distribution, we apply two-step outlier detection to identify the relative high density regions with a specified confidence level. The density of the mesh points in the selected regions larger than certain minimum density criteria are considered as blob candidates. Then we apply an efficient connected component labeling algorithm adopted from [8] on a refined triangular mesh to find different blob components. A blob is claimed to be found if the median of a blob component satisfies the minimum median density criteria.

B. *A hybrid MPI/OpenMP parallelization*

Existing blob detection approaches cannot tackle the following two challenges from the large amount of data produced in a shot and the real time requirement. In our approach, we can complete our blob detection in a few milliseconds using a hybrid MPI/OpenMP parallelization with in situ evaluation. The key idea is to exploit many cores in the supercomputer by

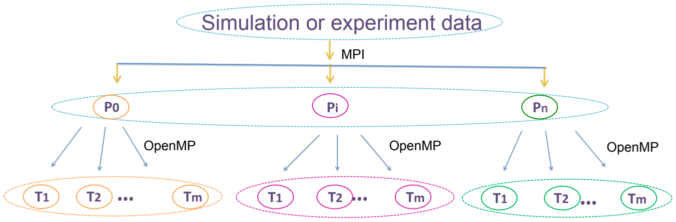
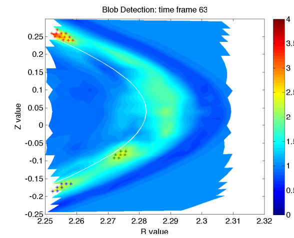
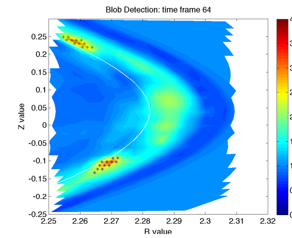


Fig. 1: hybrid MPI/OpenMP parallelization



(a) Time frame 63



(b) Time frame 64

Fig. 2: An example of the blob detection in two continuous time frames in the R (radial) direction and the Z (poloidal) direction. The separatrix position is shown by a white line and the different color stars denote blob components.

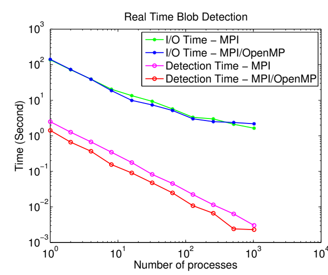
running MPI to allocate n processes to process each time frame and by leveraging OpenMP to accelerate the computations using m threads. Our hybrid MPI/OpenMP parallelization for blob detection is shown in figure 1.

IV. Experiments and Results

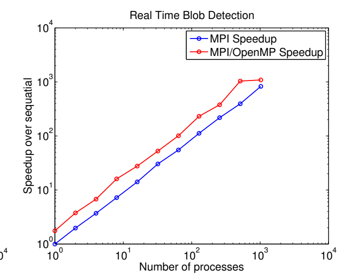
We implement our blob detection algorithm in C with a hybrid MPI/OpenMP parallelization. We have tested our implementation on the NERSC's newest supercomputer Edison with small simulation data sets (30GB) with 1024 time frames based on the XGC1 simulation [7] from the Princeton Plasma Physics Laboratory, which last around 2.5 milliseconds. Our goal is to achieve blob detection time close to 2.5 milliseconds for monitoring the process of fusion experiments in real time.

The blob detection results are shown in figure 2. Compared to the recent method in [4][5], our method does not miss blobs in the edge of regions of interests. Note that the criteria for outlier detection need to be tuned to achieve optimal performance for different fusion experimental data sets.

Our most encouraging preliminary result is that we can complete blob detection on the simulation data set described above in around 2 milliseconds with MPI/OpenMP using 4096 cores and in 3 milliseconds with MPI using 1024 cores. From figure 3, we can see that the hybrid MPI/OpenMP implementation is about two times faster than the MPI implementation when varying the number of processes from 1 to 512. With 1024 processes, both of them achieve close performance but the MPI/OpenMP one is slightly better. Also, we can achieve linear time scalability in blob detection time and slightly superlinear in I/O time. MPI and MPI/OpenMP implementations accomplish 800 and 1200 times speedup when the number of processes scale to 1024, respectively. We have been able to control analysis speed by varying the number of processes. One of future plans is to exploit a different number of threads to tune the best performance.



(a) Time



(b) Speedup

Fig. 3: Blob detection time, I/O time and speedup with MPI and MPI/OpenMP varying number of processes

Acknowledgment

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 and partially supported by NSF under a grant No. CCF 1218349, and by DOE under a grant No. DE-FC02-12ER41890. The author would like to thank Scientific Data Management Group at LBNL, and our collaborators in PPPL and ORNL for their contributions to this work; in particular, Michael Churchill at PPPL for providing his MATLAB blob detection code, and Jong Choi at ORNL for the integration of our blob detection code with the ICEE fusion analysis framework.

References

[1] ITER web page, "The International Thermonuclear Experimental Reactor (ITER) project", http://www.iter.org/, November 2004.

[2] D. A. D'Ippolito, J. R. Myra, and S. J. Zweden, "Convective transport by intermittent blob-filaments: Comparison of theory and experiment", *PHYSICS OF PLASMAS* 18, pp.1-48, 2011.

[3] Nicole S. Love and Chandrika Kamath, "Image analysis for the identification of coherent structures in plasma", *Proc. SPIE 6696, Applications of Digital Image Processing XXX*, San Diego, August 2007.

[4] W.M Davis, M. K. Ko, et al, "Fast 2-D camera control, data acquisition, and database techniques for edge studies on NSTX", *Fusion Engineering and Design*, Volume 89, Issue 5, May 2014, Pages 717720.

[5] J.R. Myra, W.M Davis, D. A. D'Ippolito, et al, "Edge sheared flows and the dynamics of blob-filaments", *Nuclear Fusion*, Volume 53 (2013) 073013 (15pp).

[6] Jong Y. Choi, Kesheng Wu, Jacky C. Wu, et al, "ICEE: Wide-area In Transit Data Processing Framework For Near Real-Time Scientific Applications", *4th SC Workshop on Petascale (Big) Data Analytics: Challenges and Opportunities* in conjunction with SC13, 2013.

[7] S. Ku, C.S. Chang, and P.H. Diamond, "Full-f gyrokinetic particle simulation of centrally heated global ITG turbulence from magnetic axis to edge pedestal top in a realistic tokamak geometry", *Nuclear Fusion* 49, 115021 (2009).

[8] Kesheng Wu, Ekow Otoo, and Kenji Suzuki, "Optimizing two-pass connected-component labeling algorithms", *Pattern Analysis and Application* (2009) 12:117-135.