

Analyse et amélioration des performances d'un logiciel C++ pour la simulation de plasma

Sid TOUATI*et Blaise FAUGÉRAS†

Janvier 2016

Sujet adaptable pour un ou deux étudiants en M1

Description du sujet

Lieu

INRIA-Sophia, équipe-projet AOSTE et CASTOR.

Prérequis

Maîtrise du langage de programmation C++.

Description du logiciel EQUINOX

Nous travaillons sur un projet de recherche qui a pour objectif d'améliorer radicalement les performances d'une application, appelée EQUINOX. Cette application de simulation permet la reconstruction de l'équilibre du plasma dans un Tokamak, application développée par l'équipe CASTOR. EQUINOX est implémentée au CEA Cadarache dans le cadre de l'upgrade du Tokamak Tore Supra en WEST et a récemment été sélectionnée par l'agence F4E (Fusion for Energy) dans le cadre de l'approche élargie d'ITER pour être implémenté au Japon pour le Tokamak JT60-SA. L'application EQUINOX est codée en C++, langage populaire et réputé en programmation orientée objet permettant une modularité et une architecture logicielle maintenable à long terme. En revanche, si les performances observées de l'application sont suffisantes pour une analyse a posteriori de l'expérience physique, elles ne sont pas suffisantes pour un usage véritablement temps réel, qui est déterminant pour que les physiciens puissent contrôler le profil de la densité de courant plasma pendant l'expérience. En terme de simulation, le temps d'exécution actuel d'EQUINOX est de plusieurs dizaines de millisecondes, alors qu'un usage temps réel nécessiterait de baisser le temps d'exécution à 1 milliseconde. Arriver à ce gain de performances est l'objectif de notre projet de recherche. L'étudiant en M1 fera un premier travail expérimental et exploratoire pour la suite du projet.

Plan du stage: Le programme du projet est le suivant :

*Professeur à l'UNS, Sid.Touati@inria.fr

†Ingénieur de recherche au CNRS, Blaise.FAUGERAS@unice.fr

1. Profilage et analyse des performances de l'application EQUINOX pour détecter les goulots d'étranglement.
2. Analyse statistique de la variabilité des performances.
3. Etudier et appliquer des méthodes d'optimisation de code avec g++. Le compilateur choisi est de préférence un logiciel libre.