ITER PHYSICS DEPARTMENT

Call for Proposals for HLST

Proposal for the Use of High Level Support Team resources

Abstract

(150 – 300 words)

Experiment and modelling are essential components of the fusion energy science research program with a key overarching goal being to reach a level of scientific understanding to enable an accurate predictive capability of burning tokamak plasma. There are many prominent and urgent scientific issues impacting the burning plasma program and the successful operation of the ITER experiment. The parallel simulation codes that address these questions are becoming more and more complex: they are composed of a growing set of sub-models, use more and more complex numerical and parallelization schemes, contain a rather large number of lines of code and accept an increasing number of input parameters. Tests should be devised to *detect bugs* that might lead to crashes as early as possible so as to ease their correction. Additionally, the *performance* of these codes has to remain at a high level so as to get results in a reasonable time on the available computers.

In this project, we will consider two simulation codes: JOREK (Edge Localized Modes and disruptions), GYSELA (Ion Temperature Gradient driven turbulence in the core). The goal is to port their Non-Regression Testing suite on Helios. These tests should be automated and triggered either each time the code repository is modified or periodically (each week-end typically). In this proposal, the work done will permit to increase the *confidence level* in two parallel codes that are already used in production on Helios. Furthermore, the monitoring, over time, of the *runtime* of representative runs on Helios, will improve the *performance* of these codes because developers will receive notice as early as possible of the performance degradation.

Project Title	*Continuous Integration tool for Parallel Applications on Helios supercomputer*
Project Acronym (up to 8 characters)	*CIPAH*

Project coordinator:

Name of Coordinator:	Guillaume latu
Institution:	CEA, France
Street Address:	
City:	13108 St-Paul-lez-Durance
Country:	France
Email:	Guillaume.latu@cea.fr
Phone:	00 33 4 42 25 63 57

Principal Investigator(s) [other than coordinator]:

Institution:	CEA, France
Names of Investigators:	Julien Bigot
Street Address:	Maison de la Simulation, Bat 565
City:	91191 Gif-sur-Yvette cedex
Country:	France
Email:	Julien.bigot@cea.fr
Phone:	01 69 08 01 75

Please duplicate the table above if Principal Investigators from more than one Research Unit

Requirements for the present largest run of the GYSELA code

Total amount of CPU hours	7 000 000 core hours
Architecture(s) where application is already used	1)INTEL Xeon 2)INTEL Itanium 3)IBM SP6 4) AMD Opteron
Number of CPUs	65 000 cores
Memory requirements	28 GB per node
Storage requirements	7 TB
Pure MPI or mixed communication (OpenMP+MPI)	Mixed OpenMP + MPI
Own code / 3rd party code	Own code
Code publicly available (yes/no)?	Yes
Library requirements	HDF5, MPI, OpenMP
Special requirements	
Site name(s) where application is already used	CINES, CCRT, HPCFF, NERSC, ...
Expected usage of the IFERC computer (yes/no)?	Yes

Technical Improvement or adaptation work done so far

1. Do you apply in parallel for similar support from other institutions?

   No

2. Has your code/project already received support (especially as part of a previous HLST call) related to improvement of its computational capabilities?

   Yes

Request for work

a) Indicate nature (type) of HLST support being requested
   1) Setup the coupling between Continuous Integration platform (ci.inria.fr or another Jenkins-like tool) and Helios supercomputer,
   2) Launch automated parallel test case on Helios machine with this platform.

b) Indicative level of support (in ppm[1])

   6 ppm

Involvement of the project proponents

Indicate the effort (in ppm[1]) of the projects proponents to be given (in parallel to the HLST work) to the execution of the project

2 ppm

Potential Impact

Indicate the estimated benefits that the HLST support activity will have on the software and physics modelling capabilities

With contribution of the High Level Support Team, the confidence level/robustness will be increased on several parallel simulation codes. The performance of the applications will be *improved* in average (the automated test will ensure that the runtimes of the application do not increase over time). There will be an improvement of the robustness and reliability of these parallel codes that address challenging issues for modelling tokamak plasma. This approach could also be considered for other parallel codes from the fusion community after the completion of this project.

[1] Note that 1ppy=12ppm

Detailed Project Description (max. 1-2 pages)

The parallel simulation codes JOREK & GYSELA are becoming more and more complex: they are composed of a growing set of sub-models, use more and more complex numerical and parallelization schemes, contain a rather large number of lines of code and accept an increasing number of input parameters. Tests should be devised to detect bugs that might lead to crashes as early as possible so as to ease their correction. To guarantee accurate results, the parallel algorithms and numerical schemes should be crosschecked if possible and have their own verification procedures. Additionally, the performance of these codes has to remain at an acceptable level so as to get results in a reasonable time on the available computers. Apart from the challenges that these parallel codes are targeting, their maintenance, and the definition of strategies to assess their robustness and reliability is a complex problem in itself.

To use the simulation code as a tool to answer to some questions, it is quite crucial to be able to trust the results of a simulation. It means that one has to have confidence in the sub-models, the numerical schemes, the algorithms, the code lines, the set of valid input parameters, etc. In this HLST proposal, the work done will permit to increase the confidence level in these codes. And also it will allow maintaining this confidence level over time by the way of Non Regression Testing. Software verification and validation is a continual effort to ensure that the tools meet all design requirements and that computed results provide a high level of accuracy and performance. For complex simulations, various individual models should be verified and validated; needless to say it can have a big impact on simulation results.

A recent work (collaboration IRFM/CEA with Maison de la simulation/CEA and IDRIS computing facility) has been to setup a minimal set of unit tests and regression testsIn a first step, we have used the Jenkins framework with limited computing power (on one node with two cores and 4GB of memory). Then, the second step consisted to couple Jenkins with a medium size parallel machine (Poincaré) located at IDRIS/France. The present HLST project is the third step: coupling Jenkins with the Helios supercomputer to run larger parallel simulations regularly on bigger configuration. The Jenkins framework currently used is located at https://ci.inria.fr, but if security issues or other reasons prevent from using this framework, one will consider installing a Jenkins instance/service close to the Helios machine for example.

An initial set of Non Regression Tests will be provided for the two codes: JOREK, that already runs on Poincaré machine (IDRIS computing facility/France). The requested work that will be made in the HLST team will be to:

1) Preparatory work
   a) port these tests on Helios supercomputer,
   b) evaluation of deployment and security issue on Helios
2) Deployment
   a) setup a strategy to launch the automated tests when required, at each commit on the software repository or regularly (e.g. each week-end).
   b) launch performance tests (small strong/weak scaling tests) to monitor/report the execution time in order to ensure that execution time does not grow over time.

Extensions to this project that are interesting if there is still sufficient time: enrich the set of tests performed on each code (Non regression Tests, Verification tests, …). This activity should be done *in close collaboration* with the responsible of each code.