Proposal n°2010PA3215

Next cut-off date for evaluation: Tuesday 1 December 2015 at 11:00:00 CET

General information

Type of proposal: Preparatory type **B – Code development and optimization by applicant (without PRACE support).**

Start date: 1st January 2016

The start date should be either one and half or two months after the announced date for evaluation cut-off.

Project name: FOODIE, Fortran Object-Oriented Differential-equations Integration Environment

Research field: Mathematics and Computer Science

Template documents for the mandatory final report :
(as described at http://www.prace-ri.eu/Information-for-PRACE-Awardees)

- PRACE_-_FRF_PC_TypeA.doc
- PRACE_-_FRF_PC_TypeB.doc
- PRACE_-_FRF_PC_TypeC.doc

Project leader (personal data and contact)

Gender: Male
Title: Dr
First name: Stefano
Last name: Zaghi
Initials:
Date of birth: 12101979
Nationality: Italian
e-mail : stefano.zaghi@cnr.it

Please give your professional e-mail address.
E-mail addresses such as Gmail and Hotmail are not accepted.

Phone number: +39 3497730036
Fax number:

Project leader (organisation and job title)

Job title: Researcher
Website: http://www.insean.cnr.it/
Organisation name: CNR-INSEAN
Department: CFD
Group:
Address: Via di Vallerano 139
Postal code: 00128
City: Roma
Country: ITALY

Organisation with a research activity: Yes

Employment contract of the project leader is valid at least 3 months after the end of the allocation period: Yes

For commercial companies,

- **Is the head office of the organisation in Europe?** –
- **% of R&D activity in Europe as compared to total R&D activity :** –

Contact person for all correspondence

Name: Stefano Zaghi
E-mail: stefano.zaghi@cnr.com

Collaborators

Organisation name: University of Miami
Department: Department of Ocean Sciences
Group:
Address: 4600 Rickenbacker Causeway Rosenstiel School Of Marine and Atmospheric Science Marine Science Center Miami, 33149 FL United States
Postal code: 33149 FL
City: Miami
Country: UNITED STATES

Dr Milan Curcic

Job title: Postdoctoral Associate
Gender: Male
Title: Dr
First name: Milan
Last name: Curcic
Initials:
Date of birth: 01021970
Nationality: Serbian
e-mail : milan@orca.rsmas.miami.edu
Phone number: +1 3054214806
Fax number:

Organisation name: Sourcery Institute
Department: Sourcery Institute
Group:
Address: 482 Michigan Ave.
Postal code: 94707 CA
City: Berkeley
Country: UNITED STATES

Dr Rouson Rouson

Job title: President
Gender: Male
Title: Dr
First name: Rouson
Last name: Rouson
Initials:
Date of birth: 01021970
Nationality: American
e-mail : damian@sourceryinstitute.org
Phone number: +1 510 600 2992
Fax number:

Organisation name: University of Maryland
Department: CROCCO LAB
Group:
Address: 3172 Glenn L. Martin Hall, College Park
Postal code: 20742 MD
City: Washington
Country: UNITED STATES

Dr Izaak Beekman

Job title: Graduate Research Assistant
Gender: Male

Title: Dr
First name: Izaak
Last name: Beekman
Initials:
Date of birth: 01021970
Nationality: American
e-mail : izaak@izaakbeekman.com
Phone number: +1 3012449367
Fax number:

Requested computer systems

Computing center	Machine
BSC *Barcelona Supercomputing Center*	☐ MareNostrum
BSC *Barcelona Supercomputing Center*	☐ MareNostrum Hybrid Nodes
CINECA	☑ Fermi
Gauss/FZJ *Juelich*	☐ Juqueen
Gauss/HLRS *High Performance Computing Center Stuttgart (HLRS)*	☐ Hornet
Gauss/LRZ *Leibniz-Rechenzentrum*	☐ SuperMUC
GENCI/CEA *Commissariat à l'Énergie Atomique*	☐ Curie Fat Nodes (FN)
GENCI/CEA *Commissariat à l'Énergie Atomique*	☐ Curie Hybrid Nodes
GENCI/CEA *Commissariat à l'Énergie Atomique*	☑ Curie Thin Nodes (TN)

1. Summary of the project *(for Types A, B, C)*

To be published in the PRACE website. Maximum 500 words.

The present PRACE-Preparatory Access (type B) project is aimed to the parallel scaling and overall performance measurements of the FOODIE (meaning Fortran Object-Oriented Differential-equations Integration Environment) library and to the eventual development/optimization efforts necessary to achieve satisfactory performances.

The (numerical) solution of Differential Equations (DEs) problems is of paramount relevance, DEs system being an ubiquitous mathematical formulation of many physical phenomena (such as those involved in fluid dynamics, chemistry, biology, evolutionary-anthropology). FOODIE is a library aimed to numerically solve DEs problems by means of a clear, concise and efficient abstract interface. FOODIE has manifolds aims: to provide a (comprehensive) set of built-in numerical schemes that are accurate, robust, validated and efficient and to allow easy application of these schemes to (almost) all DEs problems by means of an effective Abstract Calculus Pattern. The key idea is to allow the same solver-implementation to be applied to all DEs problems thus avoiding the re-implementation of the DEs solver for each different DEs problem: code re-usability is consequently maximized, FOODIE being a general robust framework. Besides, the same framework also allows rapid development of new DEs solvers due to the high abstraction level of the library itself.

FOODIE shows an interesting novelty feature: to the authors knowledge, FOODIE is the first pure Fortran code providing a totally abstract environment for solving DEs problems. FOODIE exploits new features of Fortran

2003/2008 standards, thus it being worth for state of the art.

FOODIE library has proven to be accurate and robust and it shows good performance on small-scale multi-core architectures such as "personal multi-core workstations". Extensive parallel benchmarks have been with OpenMP paradigm and with the new Fortran 2008 co-arrays (CAF). However, FOODIE aims to be ported on HPC facilities, therefore its performances must be measured on cutting-edge HPC facilities employing hundreds to thousands CPUs. In this regards, the high level of FOODIE abstraction could constitute an unknown that must be addressed, eventually requiring API optimizations.

2. Scientific case of the project *(for Types A, B, C)*

Explain the scientific case for which you intend to use the code(s). Maximum 500 words.

Modern Fortran standards (2003+) have introduced support for Object-Oriented Programming (OOP). Exploiting new features like Abstract Data Type (ADT) is now possible to develop a library providing an awesome environment for the numerical integration of Differential-Equations such as Ordinary and Partial Differential Equations (ODE, PDE).

FOODIE is tailored to the systems arising from the semi-discretization of PDEs, but it is not limited to them.

The FOODIE environment allows the (numerical) solution of general, non linear differential equations system of the form

(1) dU/dt = R(U,t)

that is coupled with the initial conditions

(2) U0 = U(t=0)

In (1) "U = U(t)" is a general "integrand" state (vector) and "R(U,t)" is a general (eventually non linear) "residual function". Considering the simplest Euler numerical solver for (1) problem it can be written as:

(3) U(n+1) = U(n) + Dt * R(U(n),t=tn) for n=0, .., N

In (3) "U(n)" is the solution corresponding to the time step "t = tn = Dt *n", "Dt" being the time step used for the numerical integration of (1). Implementing (3) for concrete problems usually means re-write (3) for each concrete "U/R(U)" pairs thus involving many potential issues: this workflow in general leads to a not clear implementation where the meaningful, conciseness and clearness of (3) is lost.

On the contrary, as scientist we want to translate our numerical formulae as easier as possible into computer codes: in particular we want that the concrete implementation mimics as much as possible the high-level mathematical language. For example we want that the Fortran implementation of (3) is (for all DE problem) like

(4) U = U + Dt * U%t(t=t)

where "U%t" is the residual function. To achieve this purpose, FOODIE exploits the Fortran 2003/2008 Object Oriented Programming (OOP) features. As a matter of facts, FOODIE provides an Abstract Calculus Pattern (based on a simple yet powerful Abstract Data Type for the DE integrands).

Summarizing, FOODIE library has two main scientific cases:

+ for developers devising new schemes for the numerical integration of differential equations (DE):

* provide a concise, clear, robust and comprehensive abstract environment by means of which:
* express the solvers formulae with a very high-level language, it being close as much as possible to their natual mathematical formulations; this ensures:
- clearness, conciseness and robustness;
- fast-developing;
+ for clients that must solve a differential equations system:
* provide a simple, standard API for many built-in DE solvers out-of-the-box, thus allowing:
- fast-solution of new problems;
- robustness: the same DE solver is applied to different problems, i.e. cross-validation;

3. Computer resources requested *(for Types A, B, C)*

Total storage required (Gbyte) (only available during the duration of the preparatory access project)	50.00
Maximum amount of memory per core (Mbyte)	1.00

4. Please provide the details listed below for the main simulation application *(for Types A, B, C)*

Name and version

FOODIE, v0.1.2 or higher

Webpage or other reference

https://github.com/Fortran-FOSS-Programmers/FOODIE

License
If the code is open source please, fill out *open source* for this query.

FOSS licensed.

FOODIE is an open source project, it is distributed under a multi-licensing system:

+ for FOSS projects:
* GPL v3;
+ for closed source/commercial projects:
* BSD 2-Clause;
* BSD 3-Clause;
* MIT.

5. Describe the main algorithms and how they have been implemented and parallelized *(for Types A, B, C ; Maximum 250 words)*

FOODIE is based on Abstract Calculus Pattern concept, see https://github.com/Fortran-FOSS-Programmers/FOODIE/wiki/High-Level-Programming.

It presently provides more than 20 DE solvers. The algorithm of each solver is totally abstract and unaware of the actual parallel model adopted for solving the concrete DE problem of the case: the parallelism is hidden into the concrete extension of the integrand abstract type.

The present project for the measurements of the FOODIE parallel scaling is aimed to prove that FOODIE can be used into HPC frameworks without performance penalties with respect standard non abstract approaches. To this purposes FOODIE is provided with extensive parallel tests. Currently there are OpenMP paradigm based tests and Co-Array Fortran (CAF) ones. Pure MPI library based tests will also be employed if issues will arise with the other two kinds of tests.

6. Current and target performance
(for Types A, B, C ; including the points below. Maximum 250 words)

- Describe the scalability of the application and performance of the application
- What is the target for scalability and performance? (i.e. what performance is needed to reach the envisaged scientific goals)

FOODIE has currently proven to have satisfactory speedup (almost close to the theoretical linear scaling) up to 10 cores order of magnitude. This project is aimed to test FOODIE speedup up to hundreds (to thousands) of cores. In particular, we want investigate the eventual performance penalties (in the HPC framework) arising from the FOODIE abstraction and in the case they will not be negligible we want optimize the current API to make FOODIE HPC-enabled.

7. Confidentiality *(for Types A, B, C)*

Is any part of the project covered by confidentiality? No

If YES, please specify which aspect is confidential and justify:

8. Describe the I/O strategy regarding the parameters indicated below *(for Types A, B, C)*

8.a) Is I/O expected to be a bottleneck?

IO is not expected to be a bottleneck.

The tests are designed to stress only the computational intensive operations, whereas the IO operations are required only once per test (at its end, for checking the correctness of the results).

8.b) Implementation: I/O libraries, MPI I/O, netcdf, HDF5 or other approaches:

The FOODIE parallel tests will need:

+ Fortran compiler supporting modern standard (2003+); preferred compilers are: GNU gfortran 5.2+, Intel Fortran v15.0+; we would also like to test FOODIE with IBM XL Fortran;

+ OpenMP-enabled Fortran compiler; the compilers previously listed satisfy this requirement;

+ Co-Array Fortran (CAF) enabled compiler; the compilers previously listed satisfy this requirements; moreover, we would like to test the OpenCoarrays project, see http://www.opencoarrays.org/

+ MPI library; in the case CAF tests will be unsatisfactory the MPI model is adopted; in this regards OpenMPI or MPICH libraries will satisfy this requirement;

8.c) Frequency and size of data output and input:

Very low frequency for IO operations. The size of IOs will be up to 1 GB per test.

8.d) Number of files and size of each file in a typical production run:

1 IO/core per test. A typical production run using N cores will generate N files in the case of pure CAF/MPI test or N/OMP_threads in the case of mixed MPI/OpenMP tests.

9. Main performance bottlenecks *(for Types B, C. Maximum 250 words)*

The FOODIE solvers deal (and know) with only the integrand Abstract Data Type. This is a very flexible approach, but it introduces some operations of allocation/de-allocation of dynamic memory that can be avoided with a non abstract (less powerful) approach. For the small-scale parallel tests already done, this issue seems to be irrelevant for the overall performance, but the scenario could change in the HPC contest where the allocation/de-allocation operations could become expensive.

10. Describe possible solutions you have considered to improve the performance of the project
(for Types B, C. Maximum 250 words)

There are few approaches that can minimize the number of dynamic memory allocation/de-allocation operations, but they should be complicated (and maybe failings). Moreover, some features arising from the current level of abstraction could be lost.

11. Describe the application enabling/optimization work that needs to be performed to achieve the target performance
(for Types B, C. Maximum 250 words)

Hopefully no work will need to obtain target performance. In the worst case, the FOODIE level of abstraction will be reduced in flavor of more speedup.

12. Which computational performance limitations do you wish to solve with this project?
(for Types B, C. Maximum 250 words)

We would like to prove that FOODIE does not introduce any performance limitations into the HPC contest. If this will be proven, FOODIE will be integrated into other numerical codes (mainly CFD ones) that will be object of eventual PRACE projects.

13. Describe the impact of the optimization work proposed? *(for Types B, C. Maximum 250 words)*

- Is the code widely used?
- Would the code be used only within this original research project?
- Would the code be used for other similar research projects with minor modifications?
- Would the code be used in many research projects of the research field indicated in the proposal?
- Would the modification be easy to add to the main release of the software?

+ FOODIE is not currently widely used;

+ FOODIE, if this preparatory access will prove that its performance are satisfactory, will be integrated into other research projects;
+ FOODIE is designed to be easily extended/modified into order to be used/integrated into other projects,
+ FOODIE can used in any mathematical/numerical projects involving the numerical solution of Differential Equations problems (including any Engineering to Physical fields);
+ FOODIE is highly modular, its extensions should be easily up-streamed.

14. Describe the request plans for work with support from PRACE experts *(for Type C only)*

14.a. Describe the level of collaboration with PRACE experts you have planned for and how much effort (person months) have you reserved for this?

Specify a rough estimate for the amount of person months this work entails :

14.b. Describe the optimization work you expect to be done with the support of a PRACE expert for your project

14.c. Please specify the amount of PRACE experts person months required to support your project (1-6 PMs):

☑ **I certify that I have read, understand, accept and comply with the terms and conditions of *PRACE Preparatory access – Call for proposals* available at http://prace-ri.eu/PRACE-Preparatory-Access**

Those terms include the ones reproduced hereinafter for the sake of clarity:

The users commit to:

a. Provide to PRACE within the period established in the guide for applicants a final report, using the proper PRACE template, with the results obtained through the access to the PRACE Research Infrastructure, as well as a qualitative feedback on the use of the resources.
b. Acknowledge the role of the HPC Centre and PRACE in all publications which include the results above mentioned. Users shall use the following (or equivalent) wording in such acknowledgement in all such papers and other publications:

« We acknowledge PRACE for awarding us access to resource [machine name] based in [country] at [site] »

Where technical support has been received the following additional text should also be used:

« The support of [name of person/people] from [organisation name], [country] to the technical work is gratefully acknowledged. »

c. Allow PRACE to publish the mentioned report as of one year from the termination of the allocation period.
d. Commit to collaborate with PRACE, upon its request, in the preparation of dissemination material.
e. The applicant commits to not use the project results for military purposes.