



Master 2 internship

Implementation of Parallel Portable Data Interface (PDI) plugins for efficient CPU/GPU data management

Maison de la Simulation offers a Master 2, or *projet de fin d'études* internship to implement PDI plugins in order to manage IO from CPU/GPU hybrid applications.

Contacts: [Julien Bigot](mailto:julien.bigot@cea.fr) ✉ julien.bigot@cea.fr ☎ +33 (0)1 69 08 01 75
[Yushan Wang](mailto:yushan.wang@cea.fr) ✉ yushan.wang@cea.fr ☎ +33 (0)1 69 08 60 70

Location: [Maison de la Simulation](#) (CEA Saclay, 91191 Gif-sur-Yvette CEDEX)

Duration: 4–6 month

Context

With the increasing complexity of numerical simulation codes, new approaches are required to analyze the data they generate. This requires to couple up-to-date data analysis libraries with the existing highly optimized numerical simulation legacy codes around the world. The PDI Data Interface code coupling library has been designed to fulfill this goal.

The **PDI Data Interface** is a library designed and developed for process-local loose coupling in high-performance simulation codes. PDI supports the modularization of codes by inter-mediating the exchange of data between the main simulation code and independent modules (plugins) based on various libraries. It is developed in modern C++ and offers C, Fortran and Python application programming interfaces.

PDI offers a reference system similar to Python or C++ `shared_ptr` with locking so as to ensure coherent access by coupled modules. It provides a global namespace (the data store) to share references and implements the *Observer* pattern to enable modules to react to data availability and modifications. It implements a metadata system that can be used to specify a dynamic type for references based on the value of other data (eg. array size based on the value of a shared integer). Codes using PDI declarative API expose the buffers in which they store data and notify it of significant steps in the simulation. Third-party libraries such as HDF5, SIONlib or FTI are wrapped in PDI plugins. A YAML file is used to interleave application code with plugins use and additional code without having to modify the original application.

PDI is actively involved in many numerical simulation models. It has been successfully tested on the [Adastra¹](#) supercomputer under the Grand-Challenge project. In this project, we identified a possible optimization of PDI in the case where data is generated on GPU. Currently, PDI does not provide plugin for GPU. If we want to expose GPU data via PDI, we need to perform the data transfert between CPU and GPU. Then we can pass to PDI the host data. What we would like to achieve is to let PDI manage the host/device data transfert.

Work-plan

During the internship, you will integrate the PDI development team and work on the implementation of PDI plugins for efficient CPU/GPU data management. You will also work with the PDI team to introduce new features in PDI as they are required for other specific use-case.

¹11th of the top 500 supercomputers worldwide

As a preliminary step, you will acquire expertise on the PDI library interface and internals by working on simplified use-cases and on small library improvements. Once familiar enough with the library, you will work on the development of PDI plugins.

In a first step, you will work on the plugin of plain text output. This step will allow you to understand the code structure of a PDI plugin.

In a second step, you will develop the GPU data plugin for PDI. You need to identify the locality of data and efficiently expose the data to PDI with minimum data transfer. You will study different programming interfaces such as HIP², Kokkos³, CUDA, etc.

For each step, you will validate your developments on parallel machines and super-computers before integrating your contributions to the main branch of PDI.

Work environment

At laboratory [Maison de la Simulation](#) you will join a group including engineers and scientists focusing on all aspects of high-performance computing (HPC). You will have the opportunity to collaborate with users of PDI in order to introduce new features in the PDI plugin family. As a member of the PDI team, you will also have the opportunity to exchange with the developers of other HPC codes for enrich your skills in HPC code development. To validate your developments, you will have access to the top European supercomputers (Aadastra, Jean-Zay, etc).

Skills

The successful candidate will master the following skills and knowledge:

- team-work and integration in an international team (English language interactions)
- software engineering and library design,
- proficiency in C++-11+ & knowledge of FORTRAN,
- parallel computing (including the MPI library).

In addition, the following skills and knowledge will be considered a plus:

- programming languages: C & Python,
- HPC parallel IO libraries such as HDF5 or NetCDF
- experience with library build and packaging tools (CMake, spack, etc.).

²HIP is a C++ Runtime API and Kernel Language that allows developers to create portable applications for AMD and NVIDIA GPUs from single source code.

³Kokkos Core implements a programming model in C++ for writing performance portable applications targeting all major HPC platforms.