Development of Solver-Market: A tool for comparing GPU-enabled Algebraic Multigrid libraries

November 26, 2025

Context

GPU enabled linear algebra libraries are at the core of many HPC scientific applications such as TRUST. Indeed, large linear systems eventually arise from the discretisation of physical systems. Algebraic Multi-Grid (AMG) methods are widely used for the resolution of these systems because of their excellent scalability property ($\mathcal{O}(n)$ with n the dimension of the matrix). The linear system resolution often represents the majority of the simulation duration (Typically 70% in the case of TRUST). Writing optimized AMG solvers is an extremely difficult task, requiring expertise in numerical linear algebra, software engineering as well as a fine knowledge of GPU architecture / optimization. Thankfully, many libraries that implement GPU-enabled AMG methods have been developed by both scientific teams and GPU vendors. These libraries are typically used as external packages in simulation

Library Name	**GPU compatibility**	**Development team / Institution**
AMGx	Nvidia only	Nvidia
rocALUTION	AMD only	AMD
MueLu (Trilinos)	Nvidia & AMD	SNL
BoomerAMG (Hypre)	Nvidia & AMD	LLNL
GAMG (PETSc)	Nvidia & AMD	ANL
Ginkgo	Nvidia & AMD	UTM
Amgcl	Nvidia & AMD	Denis Demidov
PSCToolkit (AMG4PSBLAS)	Nvidia & AMD	IRC

Table 1: Popular AMG and direct solver libraries with GPU support and their developers.

codes and save an immense chunk of development time to the users. However, testing a AMG library into a simulation code remains a very difficult task for the non-AMG-specialist. In particular, we identify 3 main difficulties:

1. API: One needs to understand the API of the library to correctly add it to a code. All of this work for perhaps disappointing performances.
2. Installation: One needs to understand how to install the library with often many variants to handle on various hardware.
3. Knowledge and tuning: One needs to learn enough about the numerical method that the library provide, as well as it's specific parameters to try and get good performances.

As a result, teams may choose a specific library for simplicity of use, or out of habits, instead of pure performances. This internship aims at simplifying the testing process of GPU AMG libraries by tackling the 3 difficulties mentioned above by developing solver-market. In particular, solver-market should be:

1. A centralized set of "input-decks" (sample codes that read matrix-market formatted matrix files https://math.nist.gov/MatrixMarket/, and perform a resolution to tackle the API challenge),

2. packaged via the spack package manager https://spack.io/ to tackle the installation challenge,
3. along with a python translator that unifies the input parameters of the several input decks, as well as coupling with a tuning tool such as GPTune https://github.com/gptune/GPTune to tackle the knowledge and tuning problem.

This tool will be crucial for CEA applications as we are preparing our codes for the arrival of the first french exascale super-computer "Alice-Recoque". We strongly consider open-sourcing the project once it is more mature.

Moreover, a library that performs poorly at time t can be improved by the development team later on, and these improvements are easily missed by users. We also want to set up continuous integration to track the performance evolution of all libraries as releases come out.

State of the project

The project has already started thanks to our current intern Ivan Huard (EPITA). We already made some technical choices and progress:

- The complexity of installing the various libraries is already handled with the spack package manager.
- Our scope is set on current portable, spack-ready AMG packages, namely MueLu, BoomerAMG and GAMG.
- We have successfully developed a first version of the translator as well as several input decks.

Job description

This project will be hosted within the LCAN laboratory where the TRUST platform is developed. First, the candidate will get familiar with the solver-market code-base and learn how to use spack as well as about AMG methods by playing around with the input decks. Then, depending on the state of the project, one or several challenge will be tackled:

- Developing an expertise in one or several AMG libraries.
- Performing proper parameter tuning on workstations and various French public super-computers (Adastra, Jean-Zay) for both matrices from TRUST and Arcane (another CEA code) using GPTune.
- Completing the parameter "translator" if necessary.
- Troubleshooting the installation of solver-market on French public super-computers.
- Setting up continuous integration both for the verification of the good functioning of solver-market and for the non-regression of the performances.
- Adding more promising libraries to solver-market namely PSCToolkit and Ginkgo.
- Writing a documentation relative to the developments made during the internship.

What we want to see

Expect to be asked about your track record of coursework or school/personal projects related to each skill you mention in your CV or during the interview. You do not need at all to know about all the mentioned skills ! Advice: show us your curiosity and quickly inquire some of the items, or elements of this job offer that you are not familiar about ! Relevant experiences are:

- Python and at least one parallel programming framework (OpenMP, MPI, CUDA, Kokkos).
- Spack, cmake.

- One of the cited libraries above.
- Numerical analysis, in particular numerical linear algebra.
- Parameter tuning tool (GPTune for instance).

General information

- Start date: Any. **Note:** Working at CEA requires an investigation that might take up to 3 months.
- Duration: $\geq$ 6 months
- Localization: CEA Saclay (bus stop "CEA Porte nord")
- Affiliation: CEA/DES/ISAS/DM2S/SGLS/LCAN
- Salary: Fixed by the CEA salary grid:

Level	**Salary**(€/ month)
BAC+3	850
BAC+4	1300
BAC+5	1400

Table 2: CEA internship salary grid

- Benefits: Company restaurant, private buses from Paris to the laboratory
- Supervisor / Contact: Rémi Bourgeois: `remi.bourgeois@cea.fr`