



Réunion Interflop Lot1

08/06/23

Bruno Lathuilière

EDF R&D



Ce qu'on avait dit

Objectives:

- a Propose an operational workflow
- b Promote a modular, sustainable and open platform with a common exchange specification between the modules while minimizing the impact on performance
- c Provide users with a software platform, an operational test environment that will be used after the project to promote the InterFLOP approach

Work program:

- a T[0-3]: Set up a co-design environment: github, general project architecture, test-case with continuous integration, bug report management
- b T[0-12]: Update existing tools so that they will be compliant with the proposed specifications

Deliverables:

- a [D1.1 @3] Documentations / Specifications
- b [D1.2 @12] Website / working environment
- c [D1.3] New version of VERROU@6, VERIFICARLO@6, CADNA@10, FPANR@12, PROMISE@12, FLDLIB@12 compliant with the proposed specification

Ce qu'on avait dit sur les formats d'échange

- ① Interface 1 : **Frontend / backend**. As this interface is highly critical in terms of performance, its specification will be done in 2 steps :
 - ▶ Specification for **native storage type** (with adaptation of Verrou and Verificarlo D1.3). This specification part will be an input to dynamorio frontend [Task2]. The new version of Verrou and Verificarlo [D1.3] will be an early input of Task6.
 - ▶ Specification extension for **software storage type** part (with adaptation of FPANR, CADNA and FLDLIB [D1.3])
- ② Interface 2 : **Code output / analysis tools**. This interface must be able to treat several kinds of outputs (scalar, vector and meshed field) for several kinds of applications (Statistical analysis [Task5], visualization [Task5], Feedback criteria...).
- ③ Interface 3 : **Instrumentation / feedback**. This interface should expose a common framework for exploratory analysis (with adaptation of PROMISE and verrou delta-debug localization) based on a 2-step scheme (search space generation and delta-debug like algorithms based on feedback criteria). This specification will be an input to Tasks 2,3,4, and 5

La plateforme

- ◆ ✓ : Site Web
- ◆ ✓ : Github
- ◆ ✗ : Intégration continue
- ◆ ✗ : Cas test

Suite à la difficulté de converger nos outils rapidement, on a décidé de passer la diffusion via la mise en place d'une image docker.

La première image qui contient Verificarlo, verrou, Cadna a été poussée sur ghcr.io :

```
docker pull ghcr.io/interflop/interflop:v0.1
```

Reste à faire :

- ◆ Un README.md d'introduction
- ◆ Ajout des outils restants
- ◆ Gestion des outils graphiques

Interface 1: Front-end/backend (synchrone)

◆ Difficultés :

- ▶ Interface statique et dynamique
- ▶ Gestion des backends avec des appels à la libc
- ▶ Mémoire partagée
- ▶ Impact sur les performances
- ▶ Opérations vectorisées

◆ Etat :

- ▶ Proposition très prometteuse (avec implémentation) de Yohan Chatelin
- ▶ Intégration délicate (encore des doutes sur les performances et le processus de compilation) à faire dans verrou
- ▶ Relecture/merge coté verifcarlo

◆ Questions ouvertes :

- ▶ Interface vectorielle
- ▶ Organisation du dépôt

Interface 1: Front-end/backend (asynchrone)

L'implémentation de Mathys : <https://github.com/Thukisdo/NSan-interflop-runtime/blob/master/src/backends/MCASync.cpp>
constitue de première preuve de concept.

Difficulté :

- ◆ Maintenance et intégration dans llvm

A-t'on les moyens de lancer le sujet du point de vue de la standardisation de l'interface d'ici la fin de l'ANR?

Interfaces codes : 2-3

Interface : Code output / analysis tools

Premiers outils d'analyse post-mortem mais sans travail de normalisation :

- ◆ verrou_plot_stat et post_verrou_dd
- ◆ VFC-CI

Interface : Instrumentation / feedback

Implémentation du Delta-Debug commune aux approches asynchrones :

- ◆ verrou_dd_line
- ◆ verrou_dd_sym
- ◆ verrou_dd_synchro
- ◆ vfc_ddebug

A priori la mise en commun des implémentations de delta-debug dédiées à la localisation asynchrone avec celle de PROMISE n'est pas pertinent.

Discussion : objectifs et renoncements

Objectif 1 : Faire l'image docker avec tous nos outils

Objectif 2 : Terminer l'interface frontend/backend dans le cas asynchrone

Critère de réussite :

- ◆ 2 backends présents dans tous les frontends (Verificarlo, Verrou, Pene, UniSIM-vp) avec les mêmes sources
- ◆ une publication

Objectif 3 : Définir le périmètre d'une interface de feedback (ie. delta-debug) et mettre en commun les sources.

Objectif 4 : Faire l'état des lieux des outils d'analyse post-mortem, et si pertinent mettre en commun.

Objectif 5/renoncement : Procédure de test commune

Renoncement : Standardisation de l'interface synchrone?