



HAL
open science

InterFLOP, Interoperable Tools for Computing, Debugging, Validation and Optimization of Floating-Point Programs

David Defour, François Févotte, Stef Graillat, Fabienne Jézéquel, Wilfried Kirschenmann, Jean-Luc Lamotte, Bruno Lathuilière, Yves Lhuillier, Eric Petit, Julien Signoles, et al.

► To cite this version:

David Defour, François Févotte, Stef Graillat, Fabienne Jézéquel, Wilfried Kirschenmann, et al.. InterFLOP, Interoperable Tools for Computing, Debugging, Validation and Optimization of Floating-Point Programs. ISC-HPC 2021 DIGITAL, Jun 2021, Online, France. <hal-03245586>

HAL Id: hal-03245586

<https://hal.science/hal-03245586v1>

Submitted on 1 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

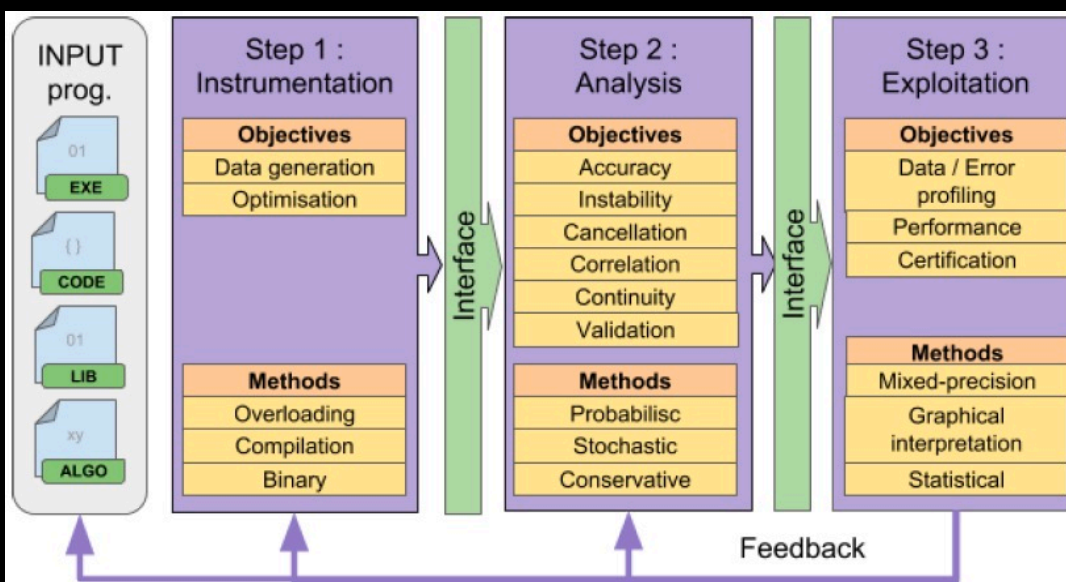


1. MOTIVATIONS

“Floating-Point arithmetic is considered an esoteric subject by many people”. *D. Goldberg*

The era is becoming increasingly complex as

- News apps are rising (AI, Drug simulation)
- New formats are becoming available (BF16, FP16, FP24, FP128, Unum, Posit, FlexPoint, FPNR...)
- New units, new operators and implementations (MatrixUnit, SpecialFunc Unit, Interpolation, Compressor...)



2. OBJECTIVES

Set a common platform integrating major tools of the French Floating-Point community to tackle the FP challenges and recent evolutions of software and hardware. We propose new analyses and combinations of existing ones to address the challenge of providing a quick and precise numerical diagnosis requiring little user expertise. InterFLOP will collect and combine information on numerical instabilities, catastrophic cancellations, unstable tests, build various statistical analyses of program executions at minimal overhead.



InterFLOP

ANR-20-CE46-0009
(2020-2024)

Funded by The French National research Agency



D. DEFOUR, F. FEVOTTE, S. GRAILLAT, F. JEZEQUEL, W. KIRSCHENMANN, J.-L. LAMOTTE, B. LATHUILIERE, Y. LHUILLIER, P. de OLIVEIRA, E. PETIT, J. SIGNOLES, D. SOHIER, F. VEDRINE.

3. DESCRIPTION

Task 1: Specification of the platform

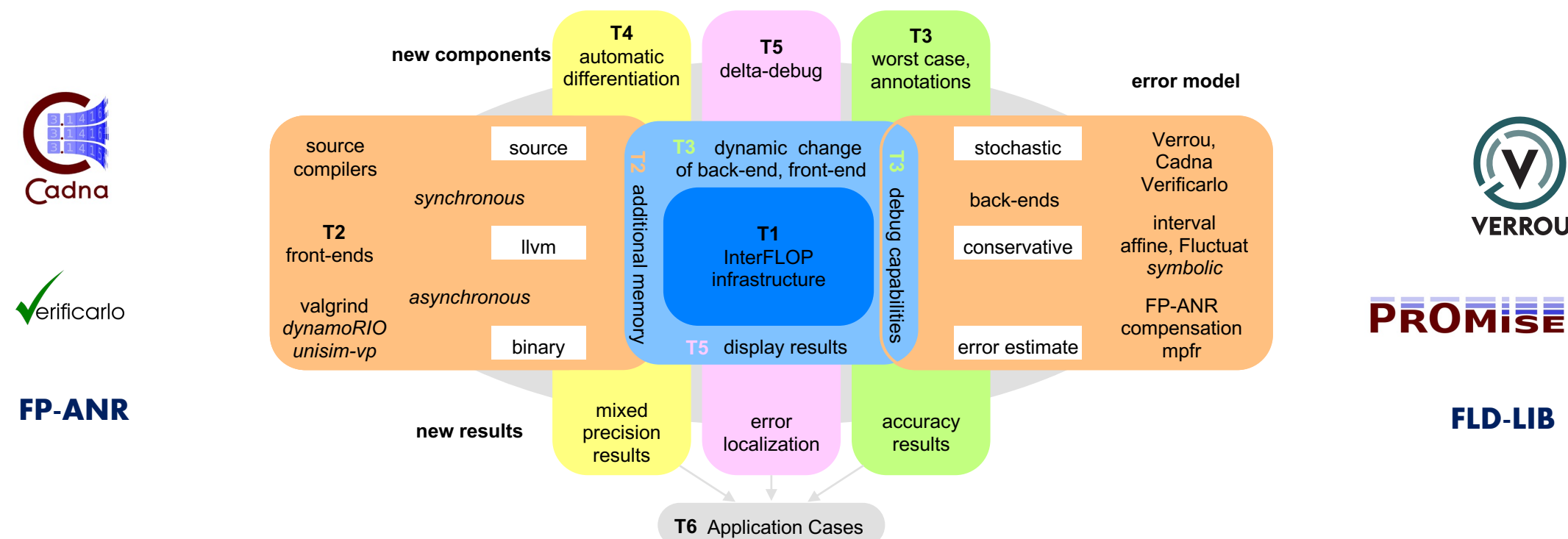
Propose an operational workflow. Define the type and the format of the data exchanged between each module to minimize bandwidth and memory usage while maximizing the amount of useful exchanged information. Promote a modular, sustainable and open platform with a common exchange specification between the modules while minimizing the impact on performance.

Task 2: Front-end and mechanism to collect information

Enable mixed analysis between the different FP arithmetic. The choice of the arithmetic will come from the availability of different back-ends: floating-point like FP-ANR, Monte-Carlo – MCA, stochastic – CESTAC and Taylor based or affine arithmetic.

Task 3: Models for error estimation and composite analysis

Define and implement composite analyses to illustrate the added value of the software chain. Promote an approach based on (i) an efficient search for instabilities (ii) guarantees of robustness / absence of instabilities on some parts of the code, and (iii) additions of generated code annotations and enabling dynamic changes of the analysis mode.



FP-ANR



PROMISE

FLD-LIB

Task 4: Precision auto-tuning and verified computing

Validate the accuracy of numerical results, automatically tune the precision to achieve the desired accuracy for the result and propose new compressed format based.

Task 5: Post-processing and statistical analysis of the results

Tackles the problem of post-processing and analyzing the results of the InterFLOP chain through three axes: statistical analysis, instabilities tracking and visualization.

Task 6: Application Cases

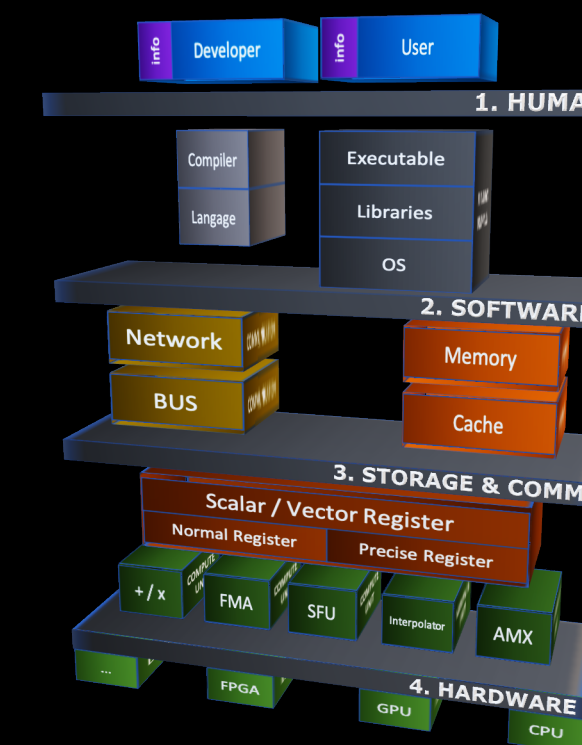
Provide a feedback on the results from the other tasks with regards to their use in industrial applications and propose new analysis methodologies. Considered applications: Yales2, AVBP, Abinit, Slang, EPX, Code_Aster, Telemac, quantitative analysis.

4. EXPECTED RESULTS

- **Common platform** available at: <https://github.com/interflop>
- **Adapt the granularity level of inspection and type of analyze to the application and user's need**
- **Automatic exploration of precision**
- **Statistical and visual analysis**
- **Validated on real applications**

5. CONSORTIUM

8 Complementary partners which bring their own expertise, tools and methodology in software analysis, compilation, Numerical Simulation, Statistics, Computer arithmetic, Parallelism, Computer architecture to make this project successful.



6. CONCLUSION

Takes the problem of numerical bug detection, software verification and validation to a new level, necessary to address issues that will be encountered with larger problems, new architectures, and new representation formats.

Numerical bug detection will be aided and guided through a unique interface at every step of the lifecycle of a software starting from its prototyping, testing, installation and operation.

Industrial and academic users could then evaluate basic compositions and develop customized ones for their own needs. Such composite analyses mixing execution (for speed on large codes), analysis (for precise diagnosis) and auto-tuning (to propose automatic enhancement) will be pioneer in the field and will be enriched with statistical and visual analysis.