



# KICKOFF TÂCHE 3

## NEW MODELS FOR ERROR ESTIMATION AND COMPOSITE ANALYSIS



- **Interne à Interflop**
  - Les analyses avec recompilation ont des besoins d'initialisation et de propagation dans les bibliothèques non instrumentables
  - Les analyses asynchrones ont des besoins de synchrone pour utiliser le débogueur plus efficacement: break if erreur >= ...
- **Besoins théoriques**
  - Détection et qualification fine des tests instables
  - Revenir en arrière sur un point de sauvegarde pour continuer l'analyse différemment
- **Besoins sur les études de cas (à compléter)**
  - ?
  - ?
  - ?

## COUPLES FRONT-ENDS / BACK-ENDS

front-ends

back-ends

valgrind  
pin  
unisim

synchrone sans addition (SA)  
fp-anr  
stochastique asynchrone (SA)  
verrou, verifcarlo

llvm sans shadow

stochastique asynchrone (AA)  
nsan

llvm avec shadow

symbolique synchrone (AA)  
herbgrind

valgrind avec shadow

exécution synchrone (AA)  
FpDebug

instrumentations C++ / C

shaman  
origins

interprétation

stochastique synchrone (AAA)  
cadna

abstrait synchrone (AAAA)  
fldlib

abstrait et symbolique synchrone (AAAA)  
formal\_debug

- **Changement dynamique de back-end dans un même front-end**
- **Dump mémoire (ou snapshot machine) à recharger dans un autre front-end**
- **Dump partiel de la mémoire avec possibilité d'interroger la mémoire du front-end stoppé**

- **découpage niveau LLVM, isolation des boucles à fort calcul**
- **nouvelle version**

- **Objectif: analyse semi-automatique d'une portion de code avec des hypothèses « minimales » et création d'un « résumé » dédié**  
**propagation des erreurs**
  - Niveau code source
  - Niveau code binaire
  - Capacités (théoriques) de synchronisation si les deux sont disponibles
- **Moyen: interface de débogueur pour indiquer progressivement ce qui est toujours négligeable et manipulation de formules graphiques**
  - Besoin d'un test et d'une coopération avec un autre front-end
- **Développement en cours**
- **API définie, mais à consolider**

## PROPOSITION D'ÉTAPES

- Définir des « challenges » partenarials
- Publier des preuves de concepts (qualité des challenges)
- Evaluer les preuves de concepts (PoC)
- Transformer les PoC pertinents en Minimal Viable Product (MVP)

# PARTENAIRE(S) × CHALLENGE × CAS D'ÉTUDE