

# Coq- Approx

Computation & Benchs

native/vm\_compute

## Quelle(s) version(s) ?

- ▶ **native-coq** (basée sur le trunk),  
Attention : mais pas mise à jour quotidienne  
git clone git ://github.com/maximedenes/native-coq.git
- ▶ **Coq-interval** version trunk  
changements dans la coqlib (inversement de paramètres :-(  
problème non encore résolu de normalisation :  
lbnf (Float (-2) (-249)) (Float 2 (-249))
- ▶ **Ssreflect** version compatible  
avec ... les 2 autres
- ▶ Avec – un peu – de chance ça tombe en marche !

# Les Benchs

- ▶ premiers tests sur *exp*
- ▶ fichiers origines : temps équivalents  
(machine un peu plus rapide entre 25% et 50% de gain)
- ▶ `vm_compute` → `native_compute` : ça ne marche pas.
  - ▶ incompatibilité avec certains modules (**bug!**)
  - ▶ solution : couper les fichiers en 2 : header + modules de calcul
  - ▶ exécution en temps... très long (feature?)

## Les Benchs (2)

- ▶ réduire le nombre d'appels à compute.  
(valable aussi pour vm\_compute)
- ▶ un unique calcul ( liste des résultats)
- ▶ reste le problème d'afficher les résultats :  
bench\_exp\_250\_120\_down temps de calcul : 57s, temps  
d'affichage : plusieurs heures (bug!!)
- ▶ solution temporaire : couper la liste en n (20) sous-listes
- ▶ gain par rapport à vm\_compute : 50% sur le calcul et 0% sur  
l'affichage
- ▶ le rapport calcul sur affichage reste de 1 pour 4.
- ▶ bench\_exp\_250\_120\_down :  
native (57s) 5m44 , vm (137) 6m43, original 8m30.

## Les Benchs (3)

### Les benchs avec calcul (benchGoal)

- ▶ même stratégie : un unique calcul booléen
- ▶ résultat le *and* de tous les résultats intermédiaires.
- ▶ plus de problème d'affichage (calcul pur)
- ▶ benchgoal\_tm\_exp\_120\_250\_245\_Up :  
native (50s) 1m3 , vm (129) 2m18, original 4m14.
- ▶ benchgoal\_tm\_exp\_120\_500\_455\_Up :  
native (101s) 1m51 , vm (248) 4m16, original 6m34.

# Conclusion

- ▶ `native_compute` 2 fois plus rapide que `vm_compute` (sans changer les structures de données)
- ▶ gain certain en réduisant le nombre d'appels (valable pour `vm_compute` aussi)
- ▶ Questions : Quel output ?
- ▶ Que retirer de cette expérimentation pour le papier ?

MERCI