



# **crofiler : Analyser la compilation de code C++**

Hadrien Grasland

2022-11-08

# Contexte

- Mars 2020 : La bibliothèque de tracking Acts **migre sur Github** pour faciliter la collaboration hors CERN
  - Problème : **Crashes CI intermittents**, à endroit variable
  - Disparaissent quand on descend à -j1 ? Contournement !
- Mai 2020 : Coupable identifié = **consommation RAM**
  - 4 fichiers cpp nécessitent >6 Go + 2 fichiers >4 Go
  - Les temps de compilation se comptent en minutes
  - Reste à comprendre pourquoi...

# Sources de données

- **Moniteur système** : Indique le fichier cpp, mais pas la raison
  - Analyse automatisée par `cmakeperf` → `visualisation`
- **Templight ?** Nécessite un clang patché, compliqué à utiliser
- Corrélation RAM/temps observée → **Analyse temporelle !**
  - **Rapport compilo ?** Juste des passes, peu intéressant
  - **Profileur ?** Indique le bout du compilo lent, pas la raison
  - Heureusement, clang  $\geq 9$  nous donne **-ftime-trace**

# Démo

[https://acts-project.github.io/metrics/metric/compile\\_max\\_rss/](https://acts-project.github.io/metrics/metric/compile_max_rss/)

# -ftime-trace

- Vient d'un dev Unity3D, actuellement spécifique à clang
- **Historique** de compilation avec données à grain fin
  - Valeur des **paramètres** (ex : types/fonctions traités)
  - Vue **hiérarchique** : on sait pourquoi le traitement est fait
- Sortie au format Chrome Trace Format
  - Diverses visualisations (chrome://tracing, Speedscope...)

# Démo

Visualisation directe time-trace

# Des visualisations perfectibles

- Centré **séquence temporelle**, pas goulot d'étranglement
- Noms de fonctions « **manglés** » illisibles (côté backend)
- Gèrent mal les **noms de types/fonctions longs**
- On aimerait aussi pouvoir **corrélérer des activités liées**
  - Coûts frontend + backend de compilation d'un template
  - Même template instantié avec des paramètres différents
- **crofiler** vise à faire tout ça + améliorer l'ergonomie

# Démo

crofiler



# Ce qui reste à faire

- Raffiner la simplification des noms d'entités C++
- Corrélation entre activités liées
- Améliorer encore l'ergonomie et les perfs d'exécution
- Consolider l'infra (intégration continue, jeu de test plus grand)
- Et faire de la pub !

**Merci de votre attention !**

<https://github.com/HadrienG2/crofiler>