

OCAML CODE OPTIMIZATION



Cette formation a été conçue par OCamlPro pour permettre aux développeurs OCaml de tirer la performance maximale des applications qu'ils écrivent. Basée sur l'implication directe de notre équipe dans le développement du compilateur OCaml, cette formation montre comment écrire le code pour qu'il soit optimisé au maximum, et comment utiliser au mieux toutes les optimisations fournies par le compilateur OCaml.

INFORMATION

Price: 1500 EUR/pers. excl. taxes

Duration: 2 days

Practice: 50%

Public

Développeurs



Pre-requisites

OCaml



Computers



Pedagogical Objectives

- (to be translated soon...)
- Base sur les performances des architectures moderne
- Runtime de OCaml
- Compilateur OCaml
- Outils
- Approches pour agir sur les performances
- Organiser l'optimisation d'un projet
- Algo et structures de données pertinentes / non pertinentes en OCaml

Langages

Français



English



Result Indicators

- Mean satisfaction score: 18.5/20
- Taux de réussite: 100%

CONTACT

✉ contact@ocamlpro.com

☎ +33 6 72 73 37 53

📍 21, rue de Chatillon, 75014, Paris, France

🏠 training.ocamlpro.com

🐦 @ocamlpro

Version: December 23, 2022

TRAINERS

Pierre Chambart (OCaml)

Pierre est ingénieur R&D chez OCamlPro depuis 2013, après une thèse de doctorat en méthodes formelles et un post-doc sur la programmation web en OCaml. Pierre est membre de la coreteam du compilateur OCaml, spécialisé dans la génération de code optimisé. Pierre est aussi l'un des développeurs principaux de la blockchain Tezos en OCaml.

Vincent Laviron (OCaml)

Vincent est ingénieur R&D chez OCamlPro depuis 2015. Après avoir travaillé sur l'adaptation de techniques d'interprétation abstraite à OCaml, Vincent a rejoint l'équipe travaillant sur Flambda, le compilateur optimisant pour OCaml, et aujourd'hui membre de la coreteam du compilateur OCaml.

TRAINING PROGRAM

(to be translated soon...)

Base sur les performances des architectures moderne

- Rapides bases d'assembleur (pour les exemples suivant)
- Architecture: Hiérarchie de mémoire/cache, Out-of-order, Prédiction, Effet des cache d'instructions décodée
- Conséquences: Interpreter les résultats des outils de sampling d'instruction (perf), Interpreter les microbench
- Système d'exploitation et appels système: Ce qui est cher, Ce qui est bloquant

Runtime de OCaml

- Cours assez exhaustif du fonctionnement du GC
- Coût des différentes parties du GC: algorithmiques pire et moyen, effets inhabituels, Cas d'exemple d'effets inattendus
- FFI (partie pertinente pour les perf): Allocations/invariants, Annotations [@...] / boxing, Racines

Compilateur OCaml

- Architecture: Suivre les transformations des différentes constructions à travers les passes
- Coût des constructions du langage
- Tour guidé des optimisations
- Différences avec les compilateur/runtime C-like

(continued on page 2)



Outils

- perf et outils dérivés
- Faire correctement des benchmarks: Analyser quel type de performances est recherché (vitesse/latence (moyenne/pire) ...), Analyser des statistiques de bench, Bench de non-régression
- Valgrind/callgrind/cachegrind...
- Le debugger gdb

Approches pour agir sur les performances

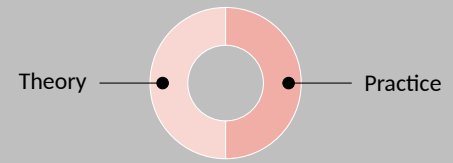
- Utiliser les optimisations / choisir des structures de code adaptées: Contrôler les optimisations, Vérifier qu'elles s'appliquent
- Traduire des parties en C-like: Analyser la pertinence, Utiliser efficacement la FFI
- Trouver, reconnaître et corriger des anti pattern
- Corriger des fuites mémoire et sur allocations: Réduire des lifetimes, Liste des types de fuites

Organiser l'optimisation d'un projet

- Estimer quels sont les performances atteignable pour un programme: Segmenter le programme et séparer l'analyse, Comparer avec la réalité
- Trouver ce qui a un sens à optimiser: Prioriser les choses les plus utiles, Quoi ne pas optimiser

Algo et structures de données pertinentes / non pertinentes en OCaml

THEORY VS PRACTICE





EVALUATING PROGRESS

We make the progress of our trainees and its evaluation a core aspect of our courses. Indeed, guaranteeing the durable acquisition of the skills at hand is key, especially for the newer comers. To that extent, we will have trainees undergo tailored group works, exercises and hands-on practice which modalities can all be customised to your specific needs.

At the end of the course, you will have an opportunity for feedback to help us improve upon our methods. This is crucial as we believe there is always room for learning on both sides of the desk and no opinion other than yours matters more for us to do so.

CONSIDERING RQTH(RECOGNITION OF HANDICAPPED WORKER STATUS)

If people with disabilities are part of the course, do reach out to us so we can adapt the training accordingly.

PEDAGOGICAL RESSOURCES

The ressources are written by the OCamlPro team prior to the courses. Documents are generally written in english and can be translated to french if need be.

FUNDING RESORTS IN FRANCE: OPCO AND CPF

The funding of the training by the OPCO is possible as OCamlPro has received the Qualiopi certification. The two following conditions must then be attained:

- The funding must cover all expenses relative to the course.
- The funding agreement must get to us five days prior to the training session at the latest.

INTER CORPORATION TRAININGS

Les horaires pour nos formations inter-entreprises in-situ sont :
Start - 9:30AM Lunch Break - 12:00PM to 01:00PM End - 05:30PM