# OCAML CODE OPTIMIZATION

This training was designed by OCamlPro to enable OCaml developers to extract maximum performance from the applications they write. Based on our team's direct involvement in the development of the OCaml compiler, this training shows how to write code that is optimally optimized and how to best use all the optimizations provided by the OCaml compiler.



# **INFORMATION**

Price: 2000 EUR/pers. excl. taxes Duration: 2 days Practice: 50%

#### Public

Developers

Pre-requisites OCaml Computers

#### Pedagogical Objectives

- Performance on Modern Architectures
- OCaml Runtime
- OCaml Compiler
- Tools
- Approaches to Improve Performance
- Organizing Optimization in a Project
- Relevant / Non-Relevant Algorithms and Data Structures in OCaml

#### Langages

French English



#### **Result Indicators**

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

## CONTACT

- contact@ocamlpro.com
- **\$** +33 6 72 73 37 53
- **Q** 21, rue de Chatillon, 75014, Paris, France
- ftraining.ocamlpro.com
- O @ocamlpro

Version: February 12, 2025

# TRAINERS

#### Pierre Chambart (OCaml)

Pierre has been an R&D engineer at OCamlPro since 2013, following a PhD in formal methods and a postdoc on web programming in OCaml. Pierre is a member of the OCaml compiler core team, specializing in optimized code generation. He is also one of the main developers of the Tezos blockchain in OCaml.

#### Vincent Laviron (OCaml)

Vincent has been an R&D engineer at OCamlPro since 2015. After working on adapting abstract interpretation techniques to OCaml, Vincent joined the team working on Flambda, the optimizing compiler for OCaml, and is now a member of the OCaml compiler core team.

# **TRAINING PROGRAM**

#### **Performance on Modern Architectures**

- Quick assembler basics (for the following examples)
- Architecture: Memory/cache hierarchy, Out-of-order execution, Prediction, Effects of decoded instruction caches
- Consequences: Interpreting results from instruction sampling tools (perf), Interpreting microbenchmarks
- Operating system and system calls: What is expensive, What is blocking

#### **OCaml Runtime**

- Comprehensive overview of the garbage collector (GC) operation
- Cost of different parts of the GC: Worst-case and average algorithmic costs, unusual effects, Example cases of unexpected effects
- FFI (performance-relevant aspects): Allocations/invariants, Annotations [@...] / boxing, Roots

#### **OCaml Compiler**

- Architecture: Tracking transformations of various constructs through compiler passes
- Cost of language constructs
- Guided tour of optimizations
- Differences with C-like compilers/runtimes

#### (continued on page 2)

# OCAML CODE OPTIMIZATION (PAGE 2)



#### Tools

- perf and derived tools
- Proper benchmarking: Analyzing the type of performance sought (speed/latency (average/worst)...), Analyzing benchmark statistics, Non-regression benchmarks
- Valgrind/callgrind/cachegrind...
- The gdb debugger

#### **Approaches to Improve Performance**

- Using optimizations / choosing appropriate code structures: Controlling optimizations, Ensuring they apply
- Translating parts into C-like code: Analyzing relevance, Effectively using the FFI
- Identifying, recognizing, and fixing anti-patterns
- Fixing memory leaks and over-allocations: Reducing lifetimes, List of leak types

#### **Organizing Optimization in a Project**

- Estimating achievable performance for a program: Segmenting the program and separating analysis, Comparing with reality
- Identifying meaningful optimizations: Prioritizing the most useful aspects, What not to optimize

# Relevant / Non-Relevant Algorithms and Data Structures in OCaml

# THEORY VS PRACTICE



# **USEFUL INFORMATION**



# **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 HANDI-CAPPED 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 OCamIPro 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