

Francesco Zappa Nardelli

Software engineer @ Meta

PROFILE

Programming languages specialist working at boundaries with system programming and computer architectures, with extensive experience of large-scale systems.

EDUCATION

2013 - HDR, computer science, École normale supérieure, France

2003 - Ph.D, computer science, Université Paris 7, France

2000 - M.Sc, Computer science, Università di Pisa, Italy

EXPERIENCE AND MAIN PROJECTS

2019 - TODAY: SOFTWARE ENGINEER, META

HACK TEAM, PROGRAMMING LANGUAGES & RUNTIMES

- Extended Meta's Hack language to make it a better fit for Meta's large framework use cases. Designed and implemented:
 - lightweight dependent types to encapsulate I/O and database accesses with uniform typed getters and setters;
 - safe support for trait diamond inheritance, removing MLocs of codegen code;
 - friend modules to protect framework internals;
 - partial classes for interoperability between manual and generated code, enabling eg. Instagram server side code port from Python to Hack.
- Designed and implemented fast method call tracing and mocking in HHVM, Meta's Hack runtime, by jitting self-modifying code: 10% unit test speedup in CI, 30% in local executions.
- Designed, implemented and deployed Meta's build system for server-side code, to specify and enforce isolation between production and internal code. Relied upon by Meta's prod continuous push.
- Optimized and formally verified concurrent queue implementations from Meta's XROS kernel.

2015 - TODAY: PROFESSOR, ÉCOLE POLYTECHNIQUE, FRANCE (PART-TIME)

Teaching the Computer Architectures and Operating Systems course, Master level.

2016 - 2019: SENIOR RESEARCH SCIENTIST (DR), INRIA, FRANCE

2004 - 2016: RESEARCH SCIENTIST (CR), INRIA, FRANCE

- On Weak Memory Models:
 - Defined the de-facto standard x86-TSO memory model of the x86 architecture, now relied upon by language standards and mainstream compiler implementations.
 - Studied the correctness of compiler optimisations in the C++ and C memory models. Identified unexpected issues in the language standards, and implemented tools to perform random testing of compiler optimisers, identifying bugs in gcc.
 - Formally proved the correctness of a compiler for a large subset of C on top of the x86-TSO memory model.
- On Debug Information
 - Designed and implemented tools to validate and synthesise DWARF unwind tables from stripped binaries and to speedup ~20x DWARF-based unwinding.
- On Language Design
 - Reverse engineered and formalised the Julia subtyping relation, relied upon by Julia runtime to implement multiple dispatch. Built a reference implementation of the algorithm, used for differential testing of Julia runtime. Led to Julia v1.0 release.
 - Designed LikeTypes, a gradual type system for dynamic languages that captures common programmer mistakes and can be implemented efficiently. Initially prototyped on top of JavaScript and V8, now supported by Meta's Hack language.

RESOURCES

- My publications and academic software are available at <https://fzn.fr>
- Meta's Hack language is open source and available at <https://hacklang.org>