Sylvain Henry

Curriculum vitae

Education

2013 · Ph.D in Computer Sciences
@ University of Bordeaux / INRIA, 4 years
Topic: "Programming Models and Runtime Systems for Many-Core and Heterogeneous Architectures".

2009 · Computer engineering diploma
@ ENSEIRB-MATMECA, 3 years
High-performance computing (HPC) specialization

2006 · Preparatory course for "Grandes écoles"
@ CPGE Camille Guérin (Poitiers), 2 years

2004 · French scientific baccalauréat
@ Lycée Pilote Innovant (Futuroscope), Jaunay-Clan, 3 years
French baccalauréat S (scientific) with mention ("Bien"), equivalent to A levels.

Teaching experience

2013 · TCP/IP Networking
@ ENSEIRB-MATMECA Info (year 2)
Course (27 hours)

2013 · GPGPU Programming
@ ENSEIRB-MATMECA Info (year 3)
Course (8 hours) – Practical sessions (8 hours)

2013 · Compilation
@ ENSEIRB-MATMECA Info (year 2)
Practical sessions (16 hours)

2012 · System & Networking
@ ENSEIRB-MATMECA Info (year 2)
Project (32h): Development of a "mpirun"-like application launcher with additional distributed shared memory support

2012 · Functional programming
@ ENSEIRB-MATMECA Info (year 1)
Project (30h): Lisp automatic solver for a "minesweeper" game

2012 · Unix, Bash, C
@ ENSEIRB-MATMECA Elec (year 1)
Practical sessions (8 hours)

2011 · C, OS (Linux), STM
@ ENSEIRB-MATMECA Telecom (year 2)
Project (64h): Development of a software transactional memory (STM) kernel module for Linux (tested within QEMU).

M. Sylvain Henry
52 avenue de Bordeaux
86490 Beaumont, France
sylvain@haskus.fr
+33 (0) 6 70 94 86 76
Perso: hsyl20.fr # github.com/hsyl20
Pro: haskus.org # github.com/haskus

Birthday: March 14, 1986

Software development skills

Domains of interest
- System programming
- Type-level programming
- Web applications (also targeting tablets)
- Many more: compilers, ray-tracing, Unicode...

Programming languages
- Expertise: **Haskell, C**
- Experience: **X86 asm, HTML/CSS/JavaScript**
- Past experience: Scala, Java, D, C++, OCaml, SQL, Delphi/Pascal, Lisp, XML...
- Poor experience: Ruby, PHP, Lua...

Frameworks, tools, etc.
- Contributions to **GHC compiler**
- System programming with Linux
- HPC: **OpenCL, CUDA, MPI, OpenMP**
- GIT, LaTeX, POV-Ray...

Communication skills

- French: native
- English: fluent
- Notions of German, Spanish and Japanese

Hobbies

- Playing music (drums, guitar)
- Reading
- Programming (Haskell, Arduino...)
- Dancing (lindy hop)
- Biking and hiking
- Travels:
  - **2018** · Philippines
  - **2018** · Poland (Krakow)
  - **2017** · Spain (San Sebastian)
  - **2013** · USA (Boston)
  - **2009** · Japan
  - **2009** · Hungary
  - **2003** · Germany

Previous jobs and internships

2016-2019 · · · Self-employed — Haskell, C, HTML/JavaScript, x86-64, Arduino

Since 2016 I have been working on my own projects and contributing to GHC (more details in the following sections).

2014-2015 · · · Postdoc at Intel Exascale Computing Research Lab — Haskell, C, Lua, x86-64

@ Versailles, 2 years (January 2014 - December 2015)

- Value profiling module in C and Lua for MAQAO (Modular Assembly Quality Analyzer and Optimizer): use binary instrumentation (x86_64 ISA) to characterize loops (e.g. cycles per iteration) and function calls (e.g. detect specialization or memoization opportunities).

- Haskell interface for MAQAO and HappStack based Web interface to navigate into disassembled binary files.

2009-2013 · · · Ph.D at INRIA Bordeaux – Assistant professor at ENSEIRB-MATMECA — C, OpenCL, Haskell

@ INRIA Bordeaux Sud-Ouest, 4 years (September 2009 - November 2013)

- OpenCL implementation (in C) called SOCL on top of the StarPU runtime system (runtime system written in C for heterogeneous architectures), now included into StarPU's distribution.

- Prototype front-end for StarPU using implicit parallel functional programming to create task graphs (in Haskell).

- Runtime system prototype in Haskell for heterogeneous architectures (independent of StarPU): ViperVM, presented at FHPC'13 workshop.

2009 · · · Internship at Thales Avionics — C, IBM CELL/BE

@ Thales, Pessac, 7 months (February 2009 - August 2009)

- Adaptation of a low-level high-performance communication middle-ware used in radars to the IBM CELL/BE processor – development in C on a cluster composed of 8 PlayStation 3 (using YellowDog Linux OS).

2008 · · · Internship at MediaLog — Delphi 5, FireBird

@ MediaLog, Mérignac, 3 months (June 2008 - August 2008)

- Optimizations for a client-server medical application (in Delphi 5): stored procedures for FireBird, etc.
- Implementation of a mechanism to ensure coherency of a patient form in the case of concurrent modifications.

2007 · · · Internship at AEI — PHP, SyncML

@ AEI, Bordeaux, 1 month (July 2007)

- Development of a contact synchronization module – using SyncML and PHP – for Vtiger CRM (Customer Relationship Management software).

Current activities and projects

System · · · Framework for system programming - haskus.org/system — Haskell, x86-64

Framework for system programming in Haskell on x86-64 architecture and Linux: native support for system formats (ELF, dwarf, cpio); fast Linux syscall wrappers as GHC primops; interface for some Linux subsystems (DRM, input).

Battuo · · · Web application - battuo.com — Haskell, HTML, JavaScript

Web application to practice drums. I am the developer, the editor and the server maintainer

Haskus · · · Haskus packages - docs.haskus.org - github.com/haskus — Haskell

All my development is open-source here except for Battuo server application

GHC studio · · · GHC interactive frontend - https://youtu.be/sPu5UOYPKUw — Haskell, Web

This is a proof-of-concept of a GHC frontend using GHC API. The aim would be to use this kind of interface to debug and optimise Haskell programs and GHC itself.

Arduino · · · MIDI filter and MIDI xylophone — Arduino C, MIDI

Lately I have been using Arduino for pet projects:

- a MIDI filter: https://haskus.org/arduino/midi

- a MIDI xylophone: work in progress

Contributions to GHC

Latest — An up-to-date and more detailed list can be found on https://haskus.org/ghc

2019-02 — Speed up compilation of large Strings literals #16190

Make GHC use ".incbin" assembler directive for large strings (better performance)

2019-02 — Represent Cmm literal strings as ByteString #16198

Improve performance of GHC

2019-01 — Add Template Haskell support for assembly files #16180

Template Haskell can be used to insert files into the compilation chain. Until now it only supported C, Objective C, C++, Objective C++ and objects (.o). I added the support for assembly files (.s).

2018-06 — Add support for built-in Natural literals into Core #14170

I have added support for Natural literals into Core with some constant folding rules. Type-level naturals desugar into Natural and benefit from this.

2018-04 — Enhanced constant folding #9136

I have added support for many new constant folding transformations into Core, especially for nested expressions.

2016-12 — Remove GHC's static flags #8440

GHC was using "static flags", i.e., flags whose values are accessed through "global variables". I converted them into dynamic flags. Some of them still use some kind of global variables but at least they can be set before the first use of the GHC API.

2016-11 — Scrutinee constant folding (Core optimization) #12877

I added a new optimization into Core: constant folding through case expressions.

2016-11 — New "-fhide-source-paths" flag #12807 #12851

I added a -fhide-source-paths flag to GHC to reduce the line noise when compiling many modules.

2016-11 — Fix float out bug #12901

Using pattern synonyms and view patterns in the GHC codebase triggered a bug because the float-out optimisation was allowed to float-out levity polymorphic expressions.

2016-11 — Fix link errors with GOLD linker #12816

2016-11 — Fix RTS linker bug with 64-bit symbol tables #12827

2016-10 — Check foreign primop imports #12355

Foreign primop imports were not checked if the given entity string was empty. The latter triggered an error in the assembler phase (jump to an empty label). I fixed this by using the function name as a label when the entity string is missing (the ccall convention has this behavior, in accordance with the Haskell 2010 report).

2016-10 — Uninstall GHC signal handlers #4162

GHC wasn't properly uninstalling its signal handlers. It is problematic when using the GHC API so I fixed it.

2016-09 — Fix foreign call argument overwriting #11792 #12614

GHC was sometimes generating bad assembly code for passing arguments to foreign calls (call into C code for instance): the arguments were corrupted.

2016-04 — Rework PThread based RTS ticker #11965

The RTS ticker thread wasn't properly disabled when in STOPPED state. I designed an algorithm using double-checked locking that was then implemented by Ben Gamari to solve this issue.

2016-02 — RTS: use timerfd on Linux #10840

The runtime system used alarm signals to implement green threads scheduling. Signals are problematic because they can interrupt system calls (syscalls) and the user-code can mess up with them. I modified the RTS so that it uses the timerfd syscall on Linux instead of alarm signals.

2015-11 — Fix "link info" ELF section #11022 #10974

GHC stores information about each build in a ELF section but it wasn't following the ELF specification while doing so ("note" sections have to follow some rules that weren't followed). I fixed this by making GHC follow the spec.

GHC used "readelf" program to read the contents of this section back. It was problematic, especially with the previous fix so I replaced it with a direct extraction of the ELF section by using Data.Binary.Get

2015-10 · · · Detect use of foreign primops in GHCi #10462

GHCi cannot use foreign primops but this wasn't checked, leading to GHC panics so I implemented the check.

2015-06 · · · Fix RTS linker allocation #9314

GHC's runtime system linker needs to allocate memory in the low 2GB of the memory (so that 32-bit relative addresing can be used). GHC was using one mmap call per object in archive files (.a) wasting a lot of memory space (mmap allocates full memory pages even for small objects) and time (mmap call is costly). I implemented an allocator named M32 allocator that loads several objects into the same memory page.

2015-05 · · · Allow execution of static programs in initramfs #7695 #10298

GHC can compile statically linked programs. However, the produced binary was infinitely looping if it couldn't find the iconv files (used to convert between character sets). I stumbled on this while trying to use a static binary into an initramfs image. I helped in fixing this.

Other contributions

2019-03 · · · Stack: add support for Git repositories with submodules #4581

Publications and writings

Latest · · · Latest writings are on my blog: hsyl20.fr

2014-08 · · · Toward Automatic OpenCL Multi-Device Support

Conference - Euro-Par 2014 - August 2014 - Porto (Portugal)

With Alexandre Denis, Denis Barthou, Marie-Christine Counilh and Raymond Namyst

2013-11 · · · Programming Models and Runtime Systems for Heterogeneous Architectures

Ph.D Thesis - November 14, 2013 - University of Bordeaux

2013-09 · · · ViperVM: a Runtime System for Parallel Functional High-Performance Computing on Heterogeneous Architectures

Workshop - FHPC'13 - September 23, 2013 - Boston, MA (USA)

2013-08 · · · SOCL: An OpenCL Implementation with Automatic Multi-Device Adaptation Support

Report - Inria Research Report - August 22, 2013

With Alexandre Denis, Denis Barthou, Marie-Christine Counilh and Raymond Namyst

2012-12 · · · Programmation multi-accélérateurs unifiée en OpenCL

Journal - Techniques et Sciences Informatiques 31, 8-9-10 - December 2012

With Alexandre Denis and Denis Barthou

2011-05 · · · Programmation multi-accélérateurs unifiée en OpenCL

Conference - RenPAR'20 - May 2011 - Saint Malo (France)