Compilers Challenges for Heterogeneous Architectures

(or compilers challenges explained to hardware architects)

Henri-Pierre CHARLES

CEA DSCIN department / Grenoble

july 11, 2022

Intro : Bio

Mini CV

- Henri-Pierre CHARLES
- 93 PhD “De l’algorithmique parallèle à la micro-optimisation”
- Assistant professor during 17 years at “Université de Versailles Saint-Quentin-en-Yvelines”
- 2008 HDR : “Contributions à la génération de code d’applications multimédia sur processeurs spécialisés”
- CEA DACLE List Grenoble since 2010

Scientific evolution

i860 / 1990 cache & pipeline / linear algebra → loop unrolling in compiler

itanium / 2000 VLIW & caches / FFT → dynamic code specialization

cell / 2004 vector instructions / video compression → dynamic vector processing

Magali / 2010 5G algorithm → Modèle de programmation dataflow

CSRAM / 2016 IA ? / In Memory Computing → new computing model / new compiler “HybroGen”

Links

- Blog activity : `https://hpcharles.wordpress.com/`
- Twitter : `https://twitter.com/henri_31415r`
- Scholar : `https://scholar.google.com/citations?user=tAcrwjgAAAAJ&hl=fr`

Introduction : What Is CEA ?

W French_Alternative_Energies_and_Atomic_Energy_

Research centers :

- DAM Military applications division aka Weapon French Sovereignty
- DEN Energy division aka Energy French Sovereignty
- DRF Fundamental research division aka Research French Sovereignty
- DRT Technological division aka industrial French Sovereignty
  Between 20 to 30 % of governement funding
  Between 80 to 70 % of industrial or collaborative project (french or european)

DRT Research and Technology

- LETI micro/nano technologies and specializes in microsystems, biotech, photonics and nanoelectronics.
- LITEN cutting edge technologies related to energy and nanomaterials. Solar, carbon-free transports, biomass-hydrogen and nano materials-nanotechnologies.
- LIST systems and software-intensive technology and specializes in embedded systems, sensors and big data, and advanced manufacturing

Introduction : Acaces-Abstract

Abstract

- Compilers are rock solid piece of software (mostly) since the 80'. During the "Dennard scaling" era the compilers domain challenges where to integrate new applications needs and integrate micro-architecture hardware evolution (pipelines, vector ALU, cache hierarchies, special instructions, etc).
- The effect of the end of the "Dennard scaling" (around 2006) has implied that performance evolution should be found in other directions. One major trend is based on hardware heterogeneity (big.LITTLE, CPU+DSP, CPU+FPGA, CPU+GPUs, in memory computing, ...).
- In other hand applications become more and more dynamic and put pressure on memory hierarchy (indirect access, sparse computation). Dense computation can nearly reach the peak performance on modern processor (see TOP500 / Linpack), sparse computation can only use few percent of the peak performance.
- One solution could rely on innovative low level code generation schemes that will be shown in this course.
- Research opportunity will be shown during this course.

Introduction : Acaces-Plan

This course shows

- Some evolutions in term of computing architecture
- Traditional compilers strategies for different programming languages, and how to play with it
- Some success and failures in term of performance from compilers,
- Compiler strategy to adapt code for dynamic applications and what are the programmer opportunities,
- How to interact between researchers on computer architecture and application domain.
- How to emulate new architectures with QEMU and how to write a QEMU plugin to measure performances
- How to play with a DSL for heterogneneous systems

Introduction : Acaces Prerequist

Knowledge to attend this course

- Know how to program in C
- Basic knowledge of a computer architecture

Benefits

- Knowledge of current challenges in compilation
- Fine knowledge of compiler internal
- Example of possible approach in research
- Applications examples : dynamic compilation, compilation for in memory computing

Introduction : What Is This Course ?

Not a formal course on compilation

- Some observations on computing architectures and compilers
- Some technicals explanation
- Some scientific problems
- Some reseach proposals

Learn by example

- How to build a full compiler
- Original binary execution modes
- Invent new instruction model level for in memory computing

Introduction : Important Message

Computer Architecture and Compilation

- A lot of potential
- New metric, energy, social impact, circular economy
- Reach physic limits
- Always new application

What to do ?

- Invention in micro architecture
- Invention in compilation domain
- Invention in code execution scheme
- Link everything together
- Share tools
- Share ideas
- Make collaborations

Introduction : Henessy Patterson Agile Dev.

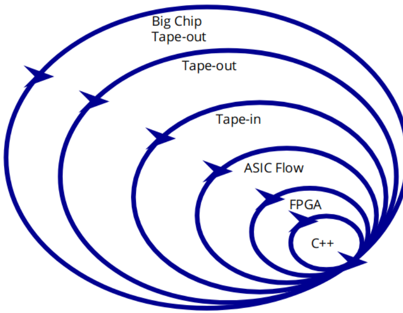
Inspiration

- Hennessy Paterson ACM lecture
- Use software methodology for HW development
- Does not work so easily. Because of ?

Illustration

Agile Hardware Dev. Methodology

Lee, Y., Waterman, A., Cook, H., Zimmer, B., Keller, B., Puggelli, A., ... & Chiu, P. F. (2016). "An agile approach to building RISC-V microprocessors." *IEEE Micro*, *36*(2), 8-20.



Small chip tape-out 100 chips 1x1mm @ 28nm is affordable at $14,000!

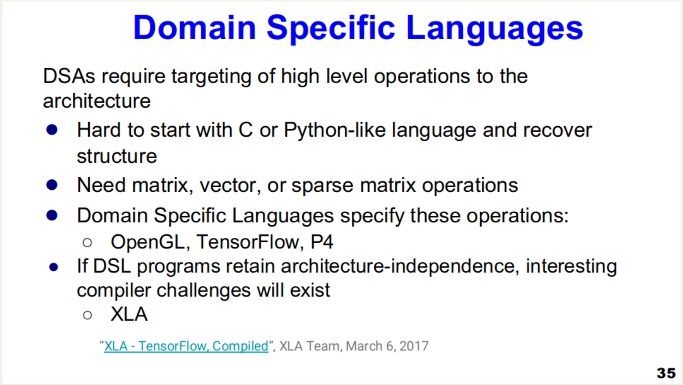
AWS FPGA F1 instance ⇒ develop new prototypes using cloud (nothing to buy)

57

Introduction : Hennessy Patterson DSL approach

Domain Specific Language

- Hennessy Paterson ACM lecture
- Domain Specific language ease the link between hardware and software



Domain Specific Languages

DSAs require targeting of high level operations to the architecture

- Hard to start with C or Python-like language and recover structure
- Need matrix, vector, or sparse matrix operations
- Domain Specific Languages specify these operations:
  - OpenGL, TensorFlow, P4
- If DSL programs retain architecture-independence, interesting compiler challenges will exist
  - XLA

"XLA - TensorFlow, Compiled", XLA Team, March 6, 2017

35