Archeology for the future
Ada on OpenVMS

GÉRARD CALLIET – PIA-SOFER

AEIC PARIS 2025

Agenda

- Authorizing – Who?
- The **Events**? – VMS and Ada Fate
- The Economic **Context** of Sustainable Development'
- Quick Overview of the **Facts**:
  - What it is? What Has Been Done? What is in Progress?
- The **Thinking**
  - Transform the steps behind into back steps
- Contribute in the Specific Context of Ada and VMS

Who?

- Among Those who ended up in IT From Elsewhere (1984)
- In my case: Philosophy, for its Greater Rigor than Mathematics
- About 30 years of Working with COBOL, Dreaming of Becoming a System Engineer on VMS
- A Break at the CNAM to Become an Engineer:
  - Discovery of Theoretical Computer Science
  - The Big Course by T. A. Hardin and V. D-G. Viguié Between CAML and Ada
- SO: I must be humbl as an engineer, even if as a philosopher, I know I'm right

On what events?

- Finally (2015), an Interesting Project: Porting the VMS/Ada Application (metro line 14) to Itanium.

- Time trap 1:
  - HP decided to end support for VMS in 2013
  - Overcome by VSI's takeover of VMS.
- Time trap 2:
  - Adacore ended support for Ada on VMS/Itanium in 2015
  - Overcome by rebuilding from FSF sources.

VMS and Ada fate

- Two considerations of similarity of fate between VMS and Ada are found:
  - How can Ada and VMS escape time traps?
  - Moving from universal recognition to a certain anonymity.
  - Successful rebounds

- SO: Something interesting could be done involving Ada and VMS

The Economic Context for Sustainable Development

- The time there are a changing
  - we have entered a period where very long, sober maintenance cycles are required
  - the question of recycling appears in all areas
- How about it for IT?

Ada + VMS + new trend for sustainability => something to do

- Understanding the rebounds
- Going on with Ada and VMS

- Supporting an universal availability of Ada on VMS
  - On VAX
  - On Alpha
  - On Itanium
  - On x86
  - …

The Facts

- What is it?
- What as be done?
  - Bys us: Rebuild of GNAT Ada on VMS Itanium
  - By VSI: Journey of port VMS to x86
- What is in progress?

What it is? The Stone Age (1) (paleolithic)

- *[1957-1998]* DEC: Digital Equipment Corporation, the MIT, Ken Olsen and the mini-computers
- *[1977-2000]* VAX: Virtual Address Extension (from 16 (pdp) to 32 bits)
  - Second lifes on emulators (Stomasy, Avt Ware, SIMH) until today
- *[1977-…]* VMS: Virtual Memory System
- *[1980-…]* VAX/VMS: The Indestructible Alliance
- **DEC Ada**: Quality implementation (Ada 83), great debugging

What it is? The Stone Age (1) (paleolithic)

- LRM augmented

7 The predefined integer types include the type INTEGER. An implementation may also have predefined types such as SHORT_INTEGER and LONG_INTEGER, which have (substantially) shorter and longer ranges, respectively, than INTEGER. The range of each of these types must be symmetric about zero, excepting an extra negative value which may exist in some implementations. The base type of each of these types is the type itself.

DEC Ada provides the following predefined integer types. Additional integer types are declared in the package SYSTEM. See section 13.7a.4 for more information.

Predefined type	Range of values	DEC Ada systems on which it applies
LONG_INTEGER	-2^{31} .. $2^{31}-1$ (or –2,147,483,648 .. 2,147,483,647)	OpenVMS VAX
	-2^{63} .. $2^{63}-1$	Alpha
INTEGER	-2^{31} .. $2^{31}-1$ (or –2,147,483,648 .. 2,147,483,647)	All
SHORT_INTEGER	-2^{15} .. $2^{15}-1$ (or –32,768 .. 32,767)	All
SHORT_SHORT_INTEGER	-2^{7} .. $2^{7}-1$ (or –128 .. 127)	All

What it is? The Stone Age (2) (neolithic)

- *[1992-…]* GNAT Ada: Ada saved by NYU, ACT, AdaCore
- *[1992-2005]* Alpha: 64 bits RISC Architecture
  - Second lifes on emulators (Stromasys, Avt Ware,…)
- *[2001-2020]* Itanium: A consortium for EPIC architecture
- *[200?-2015]* GNAT Ada Pro supported on VMS itanium by AdaCore

- *[2013]* HP announces end of life for VMS
- *[2015]* End of support of GNAT Ada on VMS by AdaCore

What it is? Modern times

- *[2014-…]* VSI resumes support of VMS (Alpha, Itanium i-n) and begin the port of VMS on x86
- *[2016-…]* Rebuilt of GNAT Ada on VMS by pia-sofer & AdaLabs

- *[today]* VMS is available on x86
- *[today]* **Ada could be supported on all VMS hardware**

Restart of GNAT Ada on OpenVMS Itanium

- GNAT Ada for Itanium is one of the targets in the gcc/ada tool chain.
- This is a Canadian-style build.
  - Build = x86_64-linux. Host and Target = ia64-hp-openvms.
- The project would have been impossible without the collaboration between:
  - David Sauvage, Adalabs: Ada and Linux expert
  - Gérard Calliet, pia-sofer: VMS expert
- The (re)build is available here: https://github.com/AdaLabs/gnat-vms

Restart of GNAT Ada on OpenVMS Itanium

- Reconstruct the work done by Adacore
- Find all components in their compatible versions
  - gcc 4.7.4 mpc-0.8.1 gmp-4.3.2 mpfr-2.4.2 binutils-2.23.1
- Reconstruct the working environment:
  - Debian's wheezy version
  - Use of debootstrap and schroot
  - Find the archive location: today http://**archive**.debian.org/debian
- Having the necessary VMS tools available, in the compatible versions:
  - includes, libraries, transfer vectors to system services, link, etc.

Restart of GNAT Ada on OpenVMS Itanium

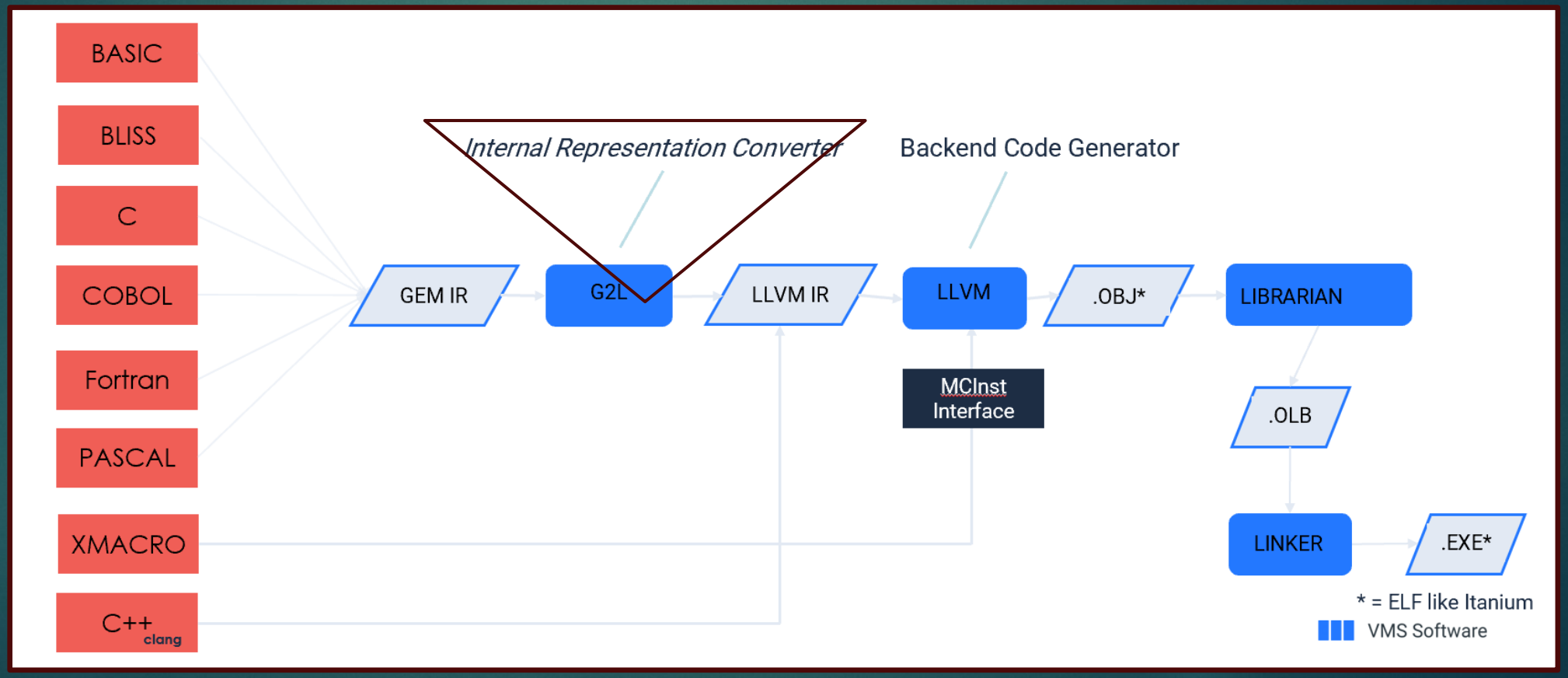
- Facilitators
  - Since VMS/Itanium, the standard elf format is used for objects.
  - Most adaptations for VMS (LD=LINK VMS, for example) are already present in the production chain, thanks to the work of Adacore.
- Difficulties
  - A number of details may be missing, and detecting the problem and finding the solution can take a long time.
  - We use the (configure + make) tool chain, which is not always easy to decipher.

Restart of GNAT Ada on OpenVMS Itanium

- The rebuild is successful, the compiler is used and supported.
- Go further.
  - Add debugging: Adacore had abandoned this, an attempt on our part has failed so far.
  - Use our workshop to have a GCC C and C++ for VMS Itanium.
  - Upgrade GCC.
    - Overcome the C version gap where GCC uses C++.
    - Without the support of work already done.

VMS journey from Itanium to x86: how?

The big picture



VMS journey from Itanium to x86: how?

- For legacy compilers: G2L Internal Representation Converter

- Calling convention: LLVM's X86_64_SysV +…
  - Argument count at all calls
  - Consistent EH data for register saves & restores
  - EH data for Alpha pseudo registers (mostly BLISS and Macro)

VMS journey from Itanium to x86: how?

- LLVM +...
  - Mixed pointer size linker relocations
  - Memory model changes
  - **.note** section generation for module name, compilation date/time, etc
  - AMD64 ABI additions for argument count in RAX register
  - Additional DWARF language tags
  - Additional EH unwind descriptors for Macro-32 VAX register emulation

VMS Journey to x86: When?

- Phase 0, on Itanium:
  - With the C++03, get a LLVM 3.4.2 code generator
  - With LLVM create cross-compilers (Itanium, x86) [for legacy compilers]
  - With the generated code Boot VMS on x86
- Phase 1, mixing tools on Itanium and Linux
  - Generate from Linux LLVM 10.0.0.1 on x86
  - With this LLVM generate native compiler [for legacy compilers]
  - With this LLVM get a clang 10 native compiler
- Phase 2, eveything on x86
  - Generate a native LLVM 20.1.1 and Clang 20
  - Needed: new version of Cmake

Ada on VMS / x86

- We have soon the complete LLVM workshop on VMS / x86
- We can use the Adacore project GNAT Ada on LLVM
- In progress:
  - Analysis of the use of LLVM by GNAT Ada
  - Analysis of the specificities on VMS LLVM x86:
    - Memory model
    - Calling standard
    - Libraries
- A lot of different strategies are possible:
  - Some cross-compilation
  - Some native builds
  - …

Universal Ada on VMS

- Is possible
- Meet real uses (on VAX, Alpha, Itanium, next day on x86)

- Always some step behind

- Which is not only a disability…

The Thinking

- Back steps
  - To a Temporal portability
  - The benefits of the Archives
  - Understanding The Step Behind
  - What about the Tailor-Made?
  - Time and Synthesis

Back Step: From Backward Compatibility to Temporal Portability

- The backward compatibility always followed in the "DEC World"
  - from pdp to x86
- This quality in the IT world has gradually evaporated
  - Transition from a quality to a luxury
- SO: why this difference in consideration between:
  - "Spatial" portability (between platforms)
  - And temporal portability
- Deciding not to chase the ever-new
- Reconsidering the role of time in computer design

Back Step: Rethinking the Role of the Archive

- Need for long-term archiving:
  - the Software Heritage Project is Worth Mentioning
- Having active communities interested in the archive:
  - for example, the SIMH sommunity
- Rethinking the museum as a place for intellectual renewal

- … Programming is Writing, Writing is Archiving

Back Step: Understanding of The Step Behind

- What to do When we're a Few Steps Behind Anyway
  - What Changes are Ahead of Us
  - Their Reasons
  - What is Preserved in these Changes
- (To be Able to Practically Adapt or Circumvent the Problem)

- … Rediscovering the Fundamentals by Analyzing the Dynamics of Change

Back Step: In defense of tailor-made solutions

- What Qualities should Programs have that Must Endure Over Time?
  - Their Usefulness
  - Their Gradual Adaptation to their Function over Time

- … There is a Relationship in our Domain Between Time and Excellence

Back Step:

- Every Synthesis is a Rebound
- It is Conceived by Considering the Role of Time

- … Which also Summarizes the Emergence of DEC, VMS and Ada, Rebound Logics

Contribute in the Specific Context of ADA and VMS

- Contribute Archiving
- Contribute Maintaining Old environments
- Contribute (veryLTS) Communities
- Find all the Existing Uses of Ada on VMS
- Make evolve GNAT Ada VMS Itanium, and the gcc workshop for VMS
- Contribute for the port of Ada on VMS x86

Thank you

- Contacts, references:
  - www.vmsadaall.org, contact@vmsadaall.org
  - www.pia-sofer.fr
  - www.vmssoftware.com
  - https://softwareheritage.org
  - https://github.com/AdaLabs/gnat-vms
  - https://github.com/AdaCore/gnat-llvm
  - https://llvm.org/devmtg/2017-10/slides/Reagan-Porting%20OpenVMS%20Using%20LLVM.pdf
  - Concepts Et Outils De Programmation. Le style Fonctionnel, le style impératif avec CAML et Ada, Thérèse Accart Hardin Véronique Donzeau-Gouge Viguié; InterEditions, 1992