



# Status of LHCb applications on 64-bit platforms

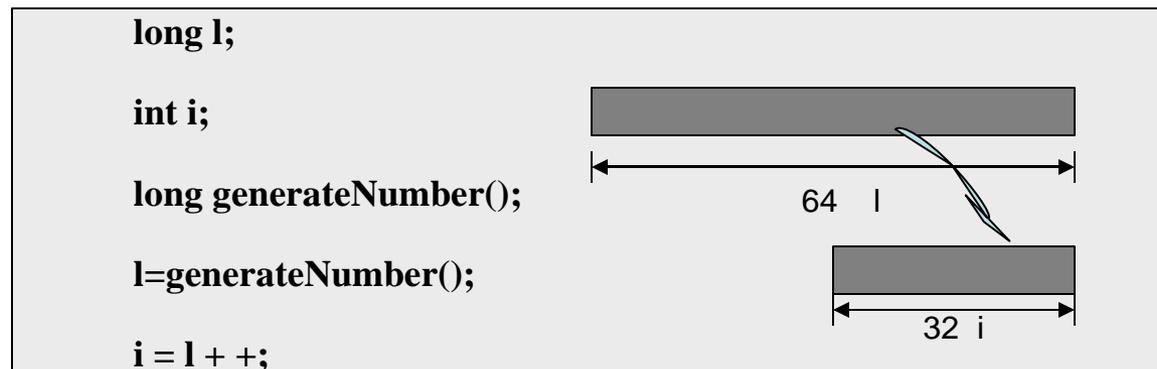
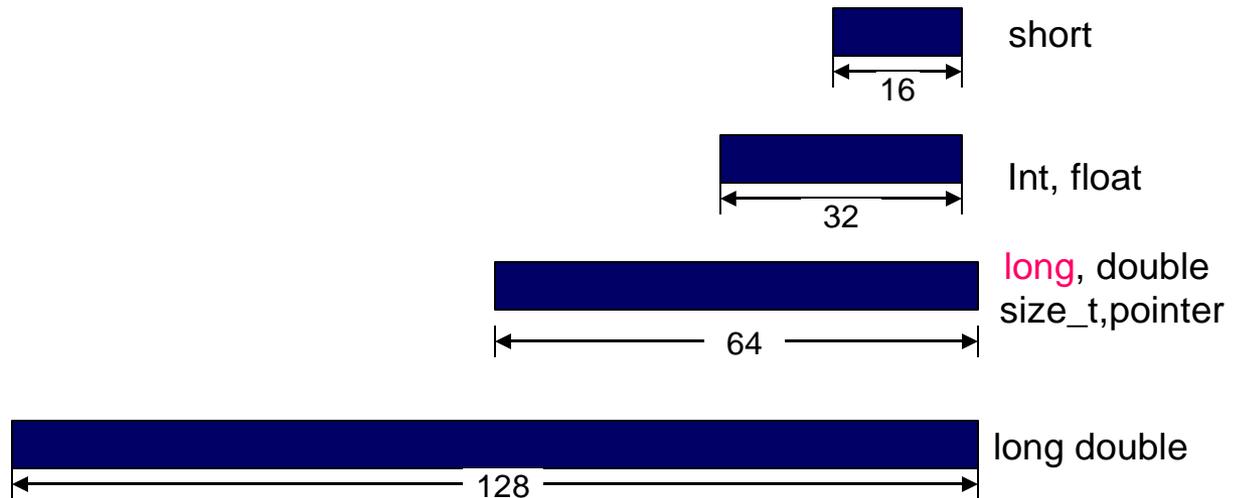
Rosa M. Garcia Rioja  
Openlab

# Contents

- 64-bit platforms
  - Source Code issues
- LHCb applications
  - Global status
  - Issues found
  - Future problems, actions and solutions
- Conclusions



# Data size



# Data size

- Do not assign long or pointers to int
- ***size\_t***, ***time\_t***, and ***ptrdiff\_t*** are 64-bit  $\Rightarrow$  Do not assume they are interchangeable with integer
- Use ANSI const instead of #def hexadecimal variables

	32 OS	64 OS
#define OFFSET1 0xFFFFFFFF	-1	4,294,967,295
#define OFFSET2 0x100000000	0	4,294,967,296

const signed int OFFSET1 = 0xFFFFFFFF;



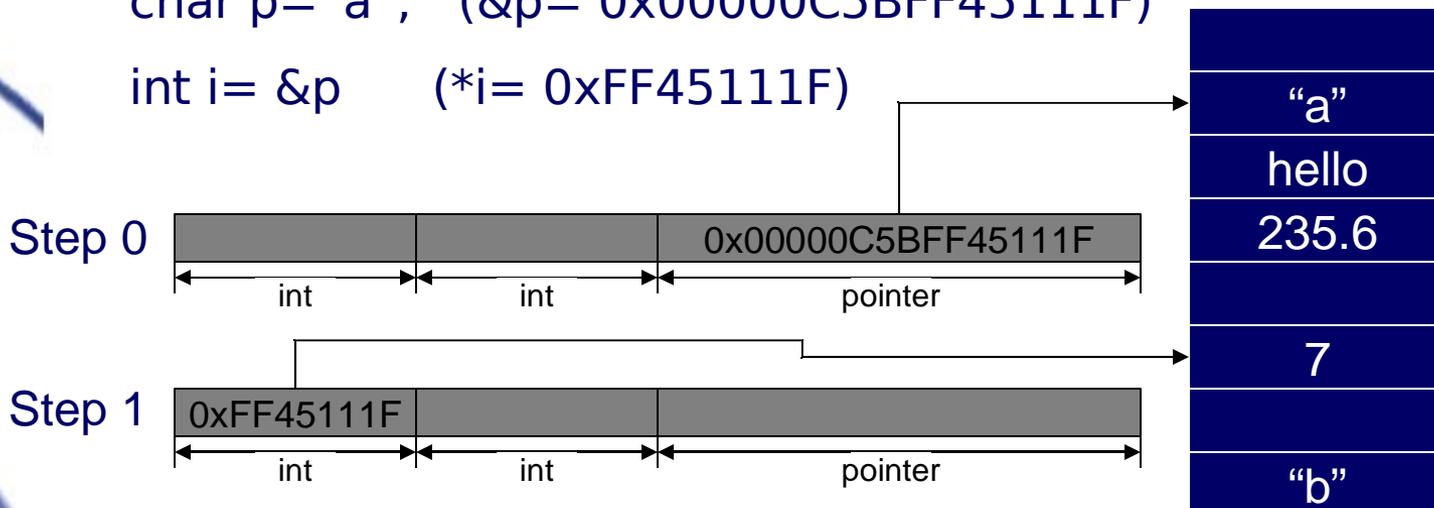
# Pointers

- Pointer sizes

- Pointer size is 64 bit.
- Wrong cast  $\Rightarrow$  Value of the pointer will be truncated
- Problems with int (linux) and long and int (windows)

char p="a"; (&p= 0x00000C5BFF45111F)

int i= &p (\*i= 0xFF45111F)



# Pointers

- Do not code with native C types that change in 64-bit OS.
- Use Macros or type definitions
- Use polymorphic types

Linux

type-cast pointer	<i>intptr_t, uintptr_t</i>
counting numbers	<i>long, size_t, ssize_t</i>

Windows

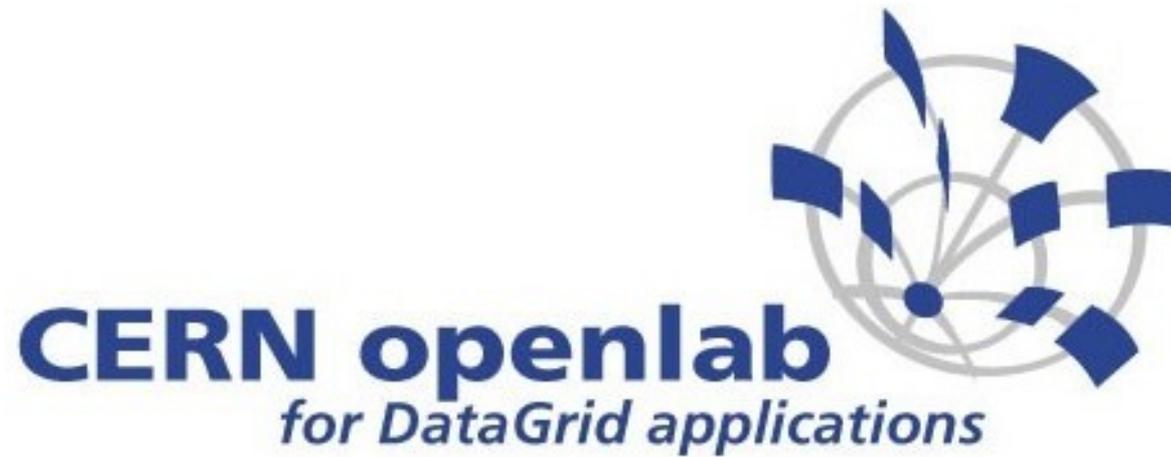
type-cast pointer	ANSI	<i>intptr_t, uintptr_t</i>
	Windows 2000 (64-bit)	<i>LPARAM, WPARAM, LRESULT, INT_PTR, UINT_PTR, DWORD_PTR, LONG_PTR, ULONG_PTR</i>
counting numbers	ANSI	<i>size_t, ssize_t</i>
	Windows\ 2000 (64-bit)	<i>int3264, SIZE_T, SSIZE_T</i>



# Some tips

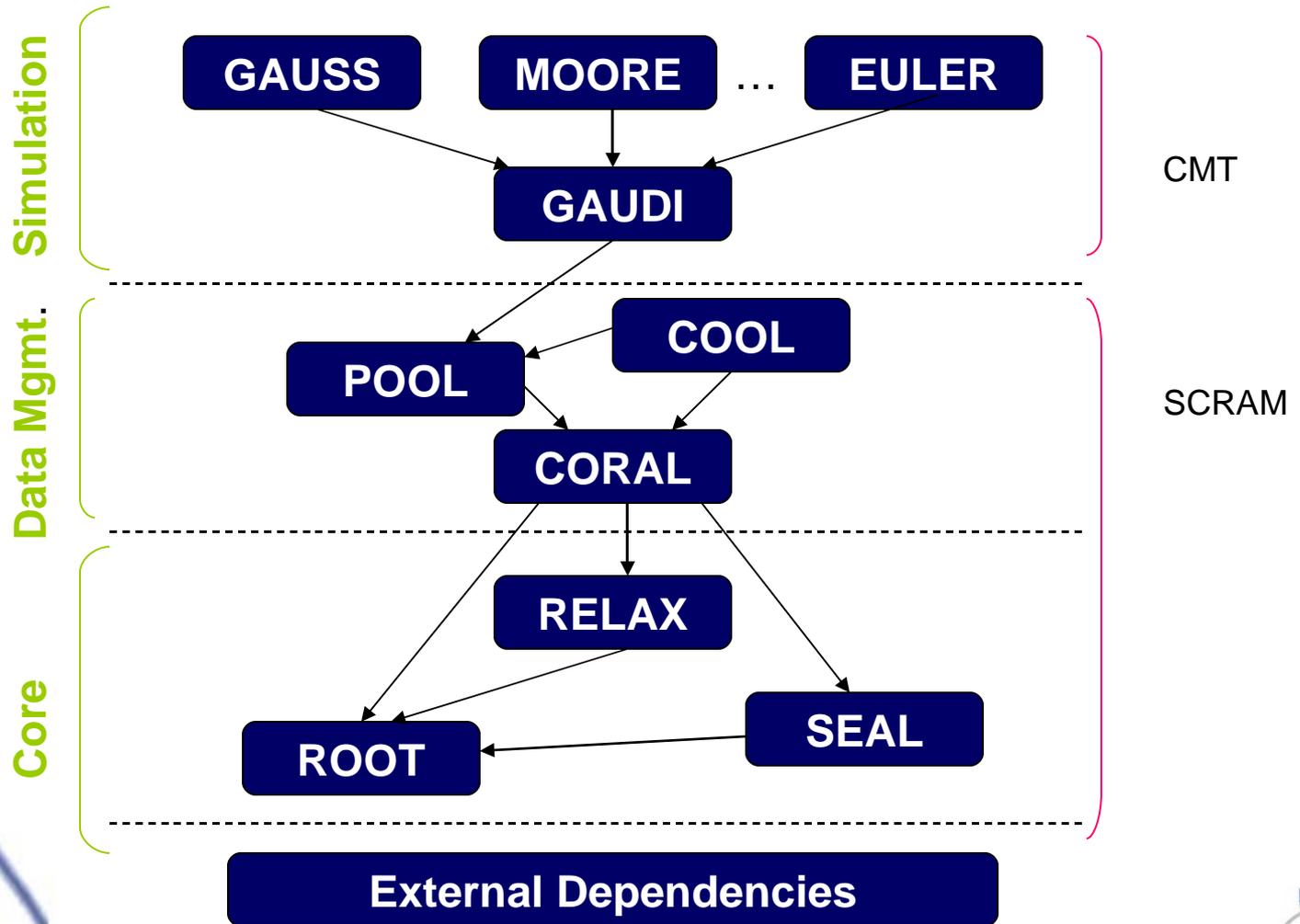
- Do not mix different data models
- Use define types or macros to isolate the code from the architecture
- Use pre-processor flags for different architectures implementations:  
\_\_LP64\_\_, \_\_M\_IA64, \_\_WIN64, \_\_WIN32
- Use specific functions to access data sizes and system parameters





# LHCb applications

# LHCb applications



# SEAL

- SEAL → Software infrastructure, basic framework libraries and tools for LHC experiments
- Merge to ROOT 5.0.8
  - Packages: Dictionary & Reflection
  - Maintain those packages for previous versions
    - Neither update, nor change or port



# SEAL packages

MathLib	Minuit	FML	MathCore
Scripting	PyROOT	PyLcgDict	PyBus
Dictionary	LcgDict	Cintex	Reflex
	Dictionary	Reflexion	
Framework	SealKernel	SealServices	
Foundation	SealBase	SealUtil	SealZip
	SealIOTools		PluginManager



RELAX



ROOT



# SEAL

- External libraries:
  - afs (careful with the right architecture)
  - Local (rebuild them & keep directory structure)
- Properly initialize environment variables
- `scram`
  - `eval `scram runtime -sh``
  - `scram b release-reset-arch`
  - `Scram build`
  - `qmtest run testsuitName`

# POOL & CORAL

- Provides a general persistency store for the LHC experiment to store events and associated metadata.
- Partially ported
  - 64 bit clean at end of March
- New application CORAL
  - Allow access to different Data structures in a transparent way.
  - 64 bit clean



# GAUDI

- Framework of simulation
  - Event Modeling
  - Persistency is crucial
  - Only one package depends on POOL → the critical one
  - Not clean



# GAUDI object diagram

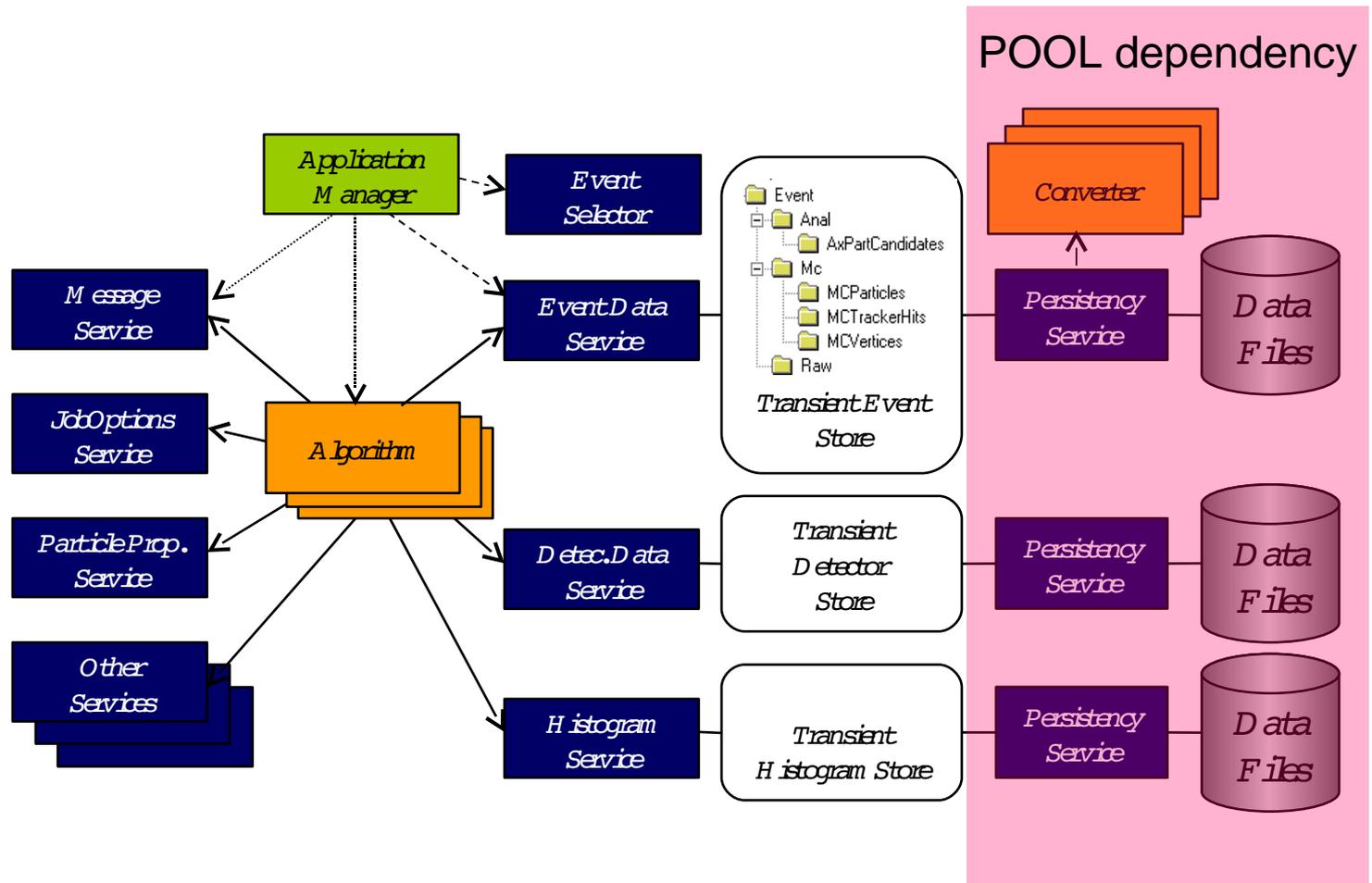


Image from: <http://lhcb-comp.web.cern.ch/lhcb-comp/Frameworks/Gaudi/GaudiTutorial.htm>



# GAUDI

- Pool is not yet clean  Main external dependency
- Build with cmt
- Set environment
- Cmt broadcast gmake
- Packages not using POOL can run on 64-bit architectures

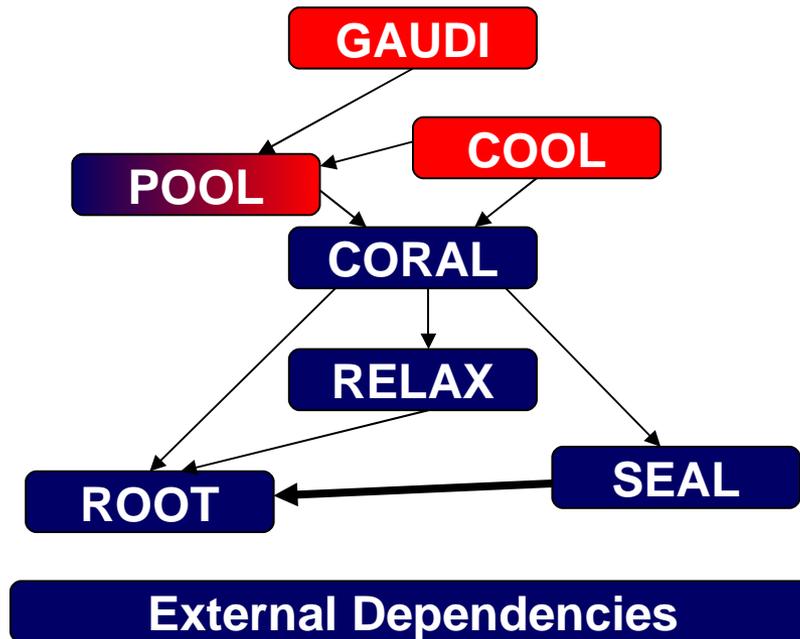


# Some external dependencies

- Some tests failing or patches needed:
  - GSI, Python, gccxml, Swig, Anaphe
- CLHEP includes `-fPIC` in makefile
- Boost
- ROOT



# Global Status



 Not clean for 64 bits

# Issues found

- External dependencies:
  - Some adapted by CERN and available only for 32-bit archs.
  - No tests to verify they really work on 64 bit archs.
- No homogeneous build tools
  - Autoconfig, make, scram and CMT
- Debuggers /memory tracers tools for 64-bit or complex applications using different languages together.



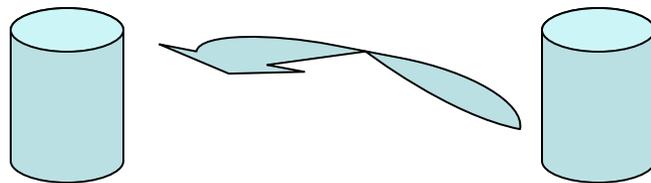
# Issues found

- Typical problems porting code from 32 to 64 bit architectures
  - Pointers, int, long ...
  - Assembler code inside the C++ code
  - Specific architectural parameters defined as constants
- Patches unapplied



# To be solved

- Check client and server architectures



Buffer 4\*long (32 bits)

Buffer 4\*long (64 bits)

- Typedefs
- A class mapping the types and architectures  $\Rightarrow$  discuss in Architects Forum



# Conclusions

- Now 64 bit architectures are being considered inside the developer community
- Most of the LHCb stack is ready
- There is some expertise on porting
- Some problems still need to be solved.



