

ILC Software Transition to 64 bit Architectures

Harald Vogt, DESY

Outline:

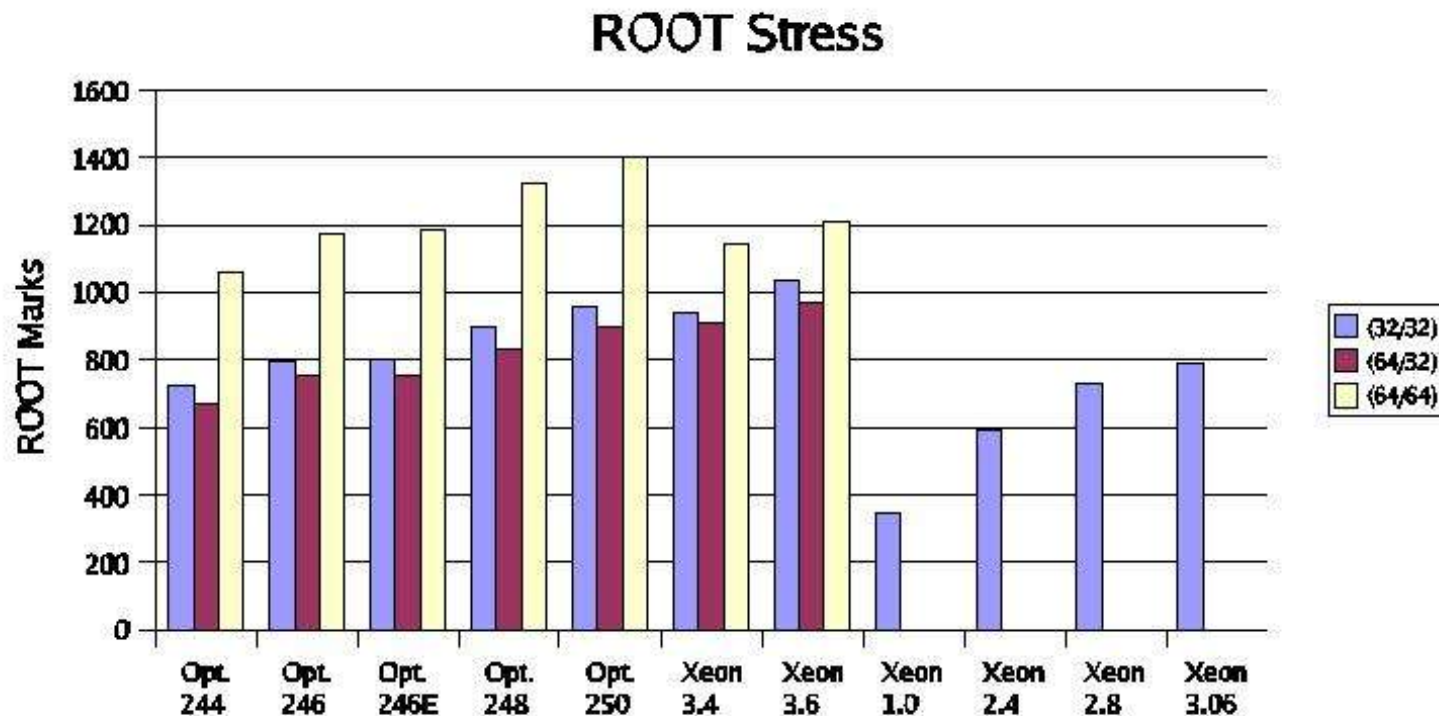
- ◆ Why 64-bit transition
- ◆ Available 64-bit Linux architectures
- ◆ 32 bit applications support under 64-bit systems
- ◆ Developers tools (compiler, debugger)
- ◆ Porting applications to 64-bit
- ◆ ILC software specifics
- ◆ The CERNLIB problem
- ◆ Conclusions

Why transition to 64 bit ?

- ◆ More and more 64-bit systems on the market
- ◆ Mainstream processor development (*Intel, AMD, PowerPC?*)
- ◆ Linux OS (Windows) supports 64-bit architectures
- ◆ Larger address space, larger file length
- ◆ Architecture improvements (registers, addressing schemes)
-> improved performance

Performance improvements

CMS Root Benchmark



64-bit mode gives gains on Opterons of about 40%

64-bit Linux Platforms

Architecture	uname -m	Size	Endian	Libpath	Miscellaneous
Alpha	alpha	LP64	little	lib	
AMD64	x86_64	LP64	little	lib64	executes x86 code natively
IPF	ia64	LP64	little	lib	executes x86 code via emulation
MIPS64	mips64	LP64	both	lib64	executes MIPS code natively
PowerPC64	ppc64	LP64	big	lib64	executes PowerPC code natively
Sparc64	sparc64	LP64	big	lib64	executes Sparc code natively
PA-RISC64	parisc64	LP64	big	—	only kernel support, no 64-bit user land, executes 32-bit PA-RISC code natively
zSeries (s390x)	s390x	LP64	big	lib64	executes s390 code natively

Data types (AMD64/LP64)

The following table shows the number of bytes used for main basic C89 types in the LP64 data model:

<u>Type</u>	<u>i386</u>	<u>AMD64/LP64</u>
int	4	4
long	4	8
pointer	4	8
float	4	4
double	8	8

64-bit and 32-bit applications together

- ◆ Supported by: *AMD64 (Opteron), Intel EM64T, MIPS64, Sparc64, PowerPC64*
- ◆ But: on *Sparc64, PowerPC64* no significant 64-bit applications
- ◆ 32 bit applications are executed natively
- ◆ 2 library paths are required therefore:
 - for 32-bit -> paths ending with */lib*
 - for 64-bit -> paths ending with */lib64*
- ◆ The dynamic linker will select the appropriate libs

64-bit Linux Developers tools

- ◆ GCC Compiler Suite (but also Intel and Portland should be usable)
- ◆ Gnu debugger
- ◆ Java 2 Platform Standard Edition (AMD64/EM64T) JDK 5.0 Update 5 for Suse and Redhat (Scientific Linux) distros
see: <http://java.sun.com/j2se/1.5.0/ReleaseNotes.html>
- ◆ The gcc compiler option `-m32` can be used to compile 32-bit applications

◆ *Porting applications to 64-bit*

- ◆ The LP64 data model requires correct function prototypes in C (if no prototype is used - return value is int).
- ◆ Reconsider carefully how arguments are passed and are expected in the function called (*int - long mangling*)
- ◆ In Fortran77 variables containing pointers must be declared as `INTEGER*8`.
- ◆ This is a serious problem for porting parts of `CERNLIB` to an 64-bit architecture (`ZEBRA`).

64-bit ILC Software

- ◆ ILC software code in Fortran77 and C requires a revision to make them portable.
- ◆ LCIO (Fortran interface): usage of data types long and INTEGER*8.
- ◆ SIMDET: usage of cernlib (pythia, kernlib, but not ZEBRA) -> no problem.
- ◆ BRAHMS: based on Geant3 (includes ZEBRA and for the interactive version also PAWLIB) -> serious problem.

The CERNLIB problem

- ◆ Pointer usage and pointer arithmetic is used heavily especially in the **ZEBRA** and **PAW** part of **PACKLIB**.
- ◆ Pointers are stored in **INTEGER** (4 Bytes) variables
- ◆ For x86-64 architectures (AMD64/Intel EM64T) this problem could be solved.
http://www-zeuthen.desy.de/linear_collider/cernlib_2005.html
- ◆ Because of a compiler bug in gcc/g77 one has to use the **-fno-f2c** compiler option.
- ◆ This requires that all code used for an application has to be compiled with that option.
- ◆ Brahms including LCIO I/O is running now on x86-64 architectures (performance gain 15-20%).

Conclusions

- ◆ 64-bit architectures are the future computing platforms
- ◆ They provide larger address space and file sizes
- ◆ Multi-language (Java, C++, C, Fortran) applications have been adapted to 64-bit architectures
- ◆ No problems seen for Java and C++
- ◆ For C and Fortran code the transition to 64 bit addresses requires strict prototyping
- ◆ This is done in LCIO but it is not done in "old" software (cernlib)
- ◆ It has been solved and all official ILC software is now ready to run on 64-bit architectures