

ITANIUM[®] SOLUTIONS

A L L I A N C E

SUCCESS IS YOURS FOR THE TAKING. JOIN US.

Completing a Successful Migration

Porting from 32 to 64 Bit: Step-by-step

- **Decision making/planning**
 - In some cases 32 bit is all you need !
- **Porting strategy**
 - Port everything ?
 - Integrate 32-bit components by IPC
- **Identify dependencies**
 - Are your tools, libraries, drivers all ported?
 - Do you have the required operating system, hardware?
 - Linux 32 versus Linux 64 differences
- **64bit Code Clean**
 - Use the compiler to help in 64-bit porting (-wp64)
 - ANSI-compliant coding
 - One source base only
 - Use pre-processor definition where really necessary
- **Fix potential run-time issues**
 - Debugging
 - Memory alignment
- **Optimization**
 - Use tools like Intel® Vtune™ to optimize for target architecture

Defining a Successful Ported Application

- Application runs on the new platform
- One source code base only
- Third-Party dependencies will be supported for new platform too in the future
- Application performs well on the new platform
- Application is sold on the new platform

Key to a Successful Migration Start

- **PLANNING!**
- Product experts need to be involved
- Allow time for testing
- Allow time to learn new tools
- Do forget to plan to migrate any build environment/tools
- Third party requirements
- Know the environment and requirements **BEFORE** you start
- Allow time to update documentation
- Aren't 32 bit enough for your application ?

Sometimes 32-bits is Enough!

- Minimize work by leaving some apps 32-bits:
 - Text editors
 - Line-Drawing “Drafting” applications
 - Many compiler-like apps

Useful Rule-of-Thumb: 1.5 “bits” of capacity per year

1. Measure your largest data space’ (file size, in-memory data, net transmission, etc.) capacity
2. Multiply by 1.5^{10} (≈ 58)

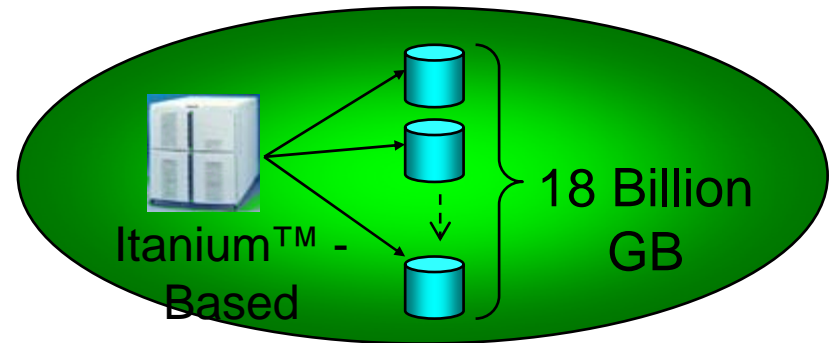
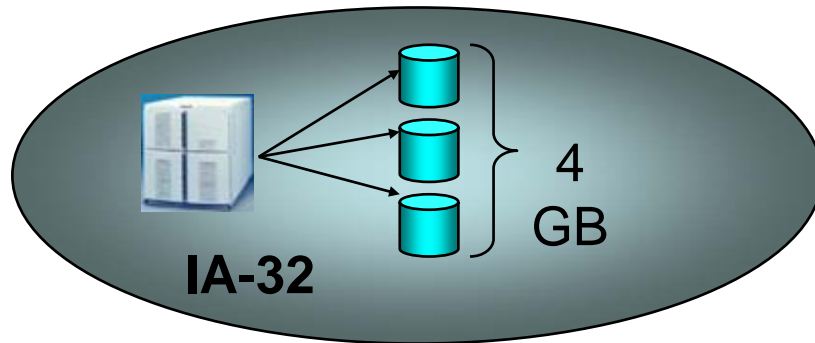
If the result is *still* $\leq 2^{32}$, the app can stay 32 bits

- native, fast Itanium® instructions (currently only HP-UX)
- as competitive, already-built x86 binaries
 - But then performance still might make a move to 64bit useful !

Do you *really* need a 64-bit editor ?

64-bit Address Space

- Benefit from 64-bit addressing
 - Virtual address space of 2^{64} bytes
 - For server & database applications
 - For huge data-set manipulation code



The physical address space is smaller : 50 Bits on Itanium® 2 Processor

The operating system might limit (virtual) size of memory segments to something less than 2^{64}

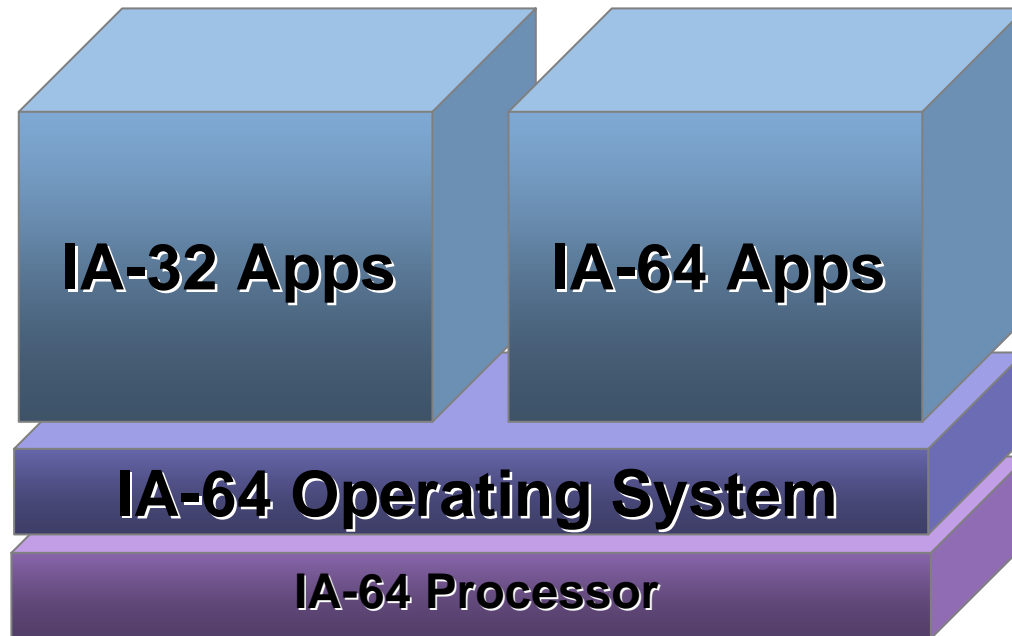
However we live in a REAL world ...

- The physical address space is smaller
 - 2^{52} for Itanium® 2 Processor
 - Much larger than on x86-64 systems (typically 2^{36} up to 2^{40})
- The Itanium® system in the market today have lower limits again for physical amount of memory
 - Typically $\geq 2^{36}$
 - But big enough for huge (512/1024/ ... processor) Itanium® 2 SMP systems
- The operating system might define further limits ...

Limits of SUSE Linux Enterprise Server 9, kernel version (version 2.6.5-7)

Platform	x86	Intel® Itanium® Processor Family	X86-x64 (Intel EM64T / AMD x64)	IBM® S/390® (31 bit)	IBM® zSeries®	IBM® POWER™
<i>Kernel related items</i>						
CPU bits	32	64	64	32	64	64
max. #CPUs (practical)	4-32	4-512	8	16	32	64
max. #CPUs (theoretical)	128	512	8	32	64	32 (iSeries), 128 (pSeries)
max. user-/kernelspace	3 GB	2 EB (full 64 bits)	512 GB	2 GB	N/A	2 TB
max. RAM (practical)	48 GB	4 TB	512 GB	2 GB	256 GB	512 GB
max. RAM (theoretical, practical)	64 GB	1 PB	1 TB	2 GB	4 TB	1 PB
max. swap space	up to 32 * 64 GB					

32 Bit Compatibility in Architecture

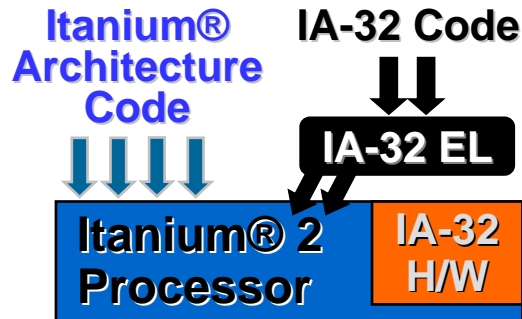


- Full compatibility with the x86/87 instruction set
- Includes SSE-1 and Intel® MMX™ Technology instructions
- IA-32 supported in processor hardware
- Support for inter- & intra-proc. Itanium® / IA-32 transitions

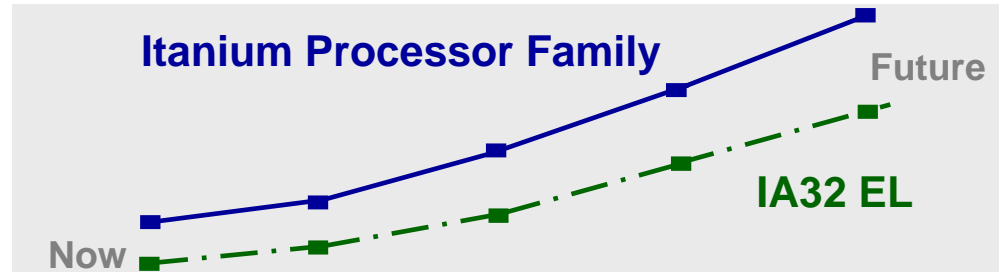
IA-32 Execution Layer

- IA-32 Execution Layer (EL-32) supports IA-32 applications running on Itanium® 2-based systems by software emulation

IA-32 EL Is an Advancement Over IA-32 On-die Hardware



IA-32 EL Performance Scales with Future Itanium® Processors¹



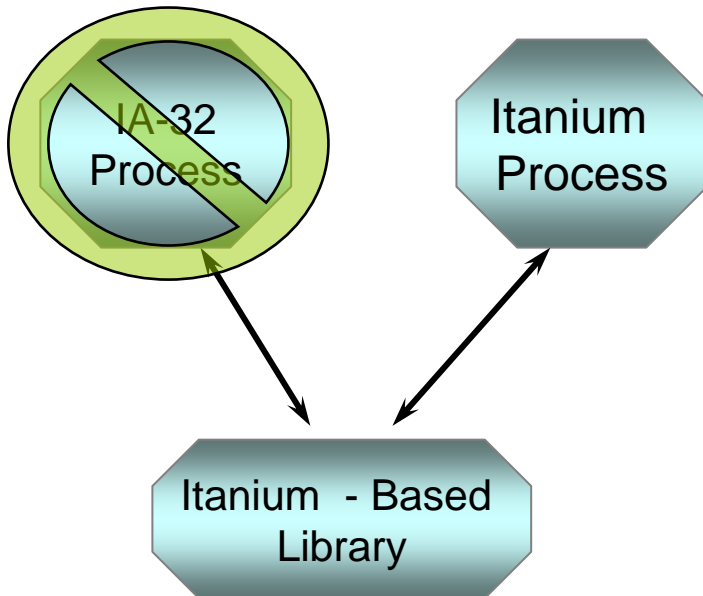
¹ Data extrapolated from measurements using frequency scaling and assuming incremental optimization. All products, dates, and figures are preliminary and are subject to change without notice.

- Supports Instructions Set extensions (SSE-2, SSE-3)
- Approach similar to Java JIT (Just-In-Time compilation)
- Availability
 - Microsoft Windows 2003 Server, SP1 or as a separate download package
 - SUSE SLES 8.x, Redhat 3, 4
- Performance
 - Estimate 32-bit application performance on Itanium 2 processor 1.5 GHz, 6 MB similar to Xeon™ processor MP 1.5GHz (performance varies by application)
 - Delivers ~ 50% to 70% of native Itanium architecture performance

Combining 32Bit and 64Bit Code

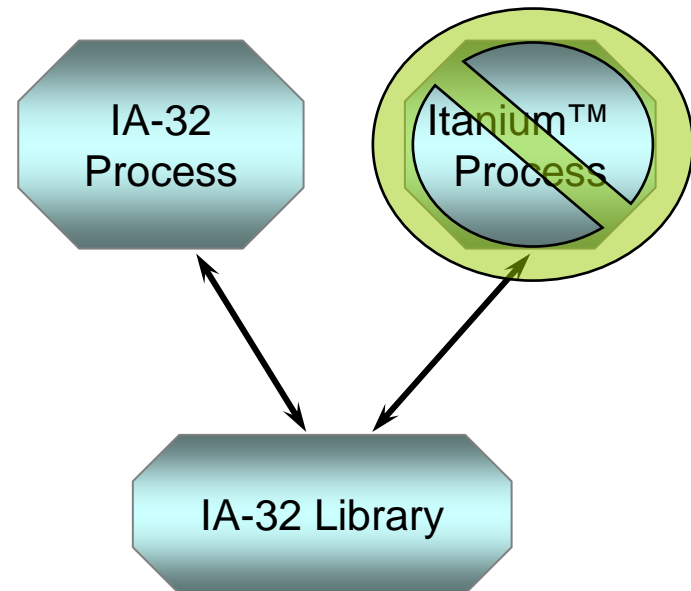
● Itanium library port

- The library can only be used by Itanium apps.



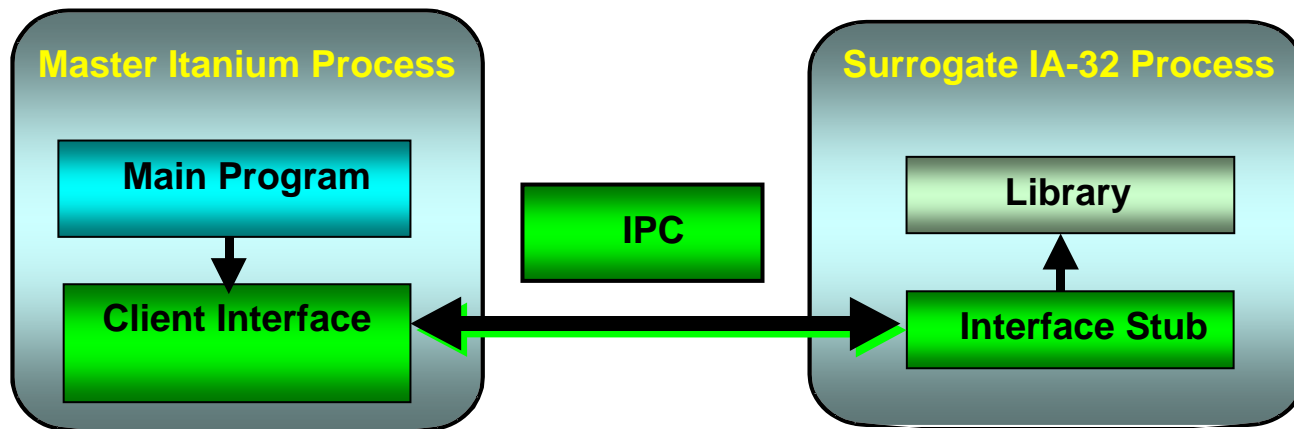
● IA-32 library

- The library can only be used by IA-32 apps.
- OS does not allow mixing of IA-32 and Itanium™ instructions.



Combining 32Bit and 64Bit Code

- **32-bit library access to 64-bit process through IPC/RPC/Shared memory objects**
 - **Surrogate binaries can be used to manage the IPC translation with no changes to existing code**



* Third party names and brands are the property of their respective owners

Byte Order Issues

- Endianism
 - Several UNIX legacy platforms are big-endian
 - Most Linux platforms are little-endian
 - Itanium HW can support both however
 - Use named bit fields and unions/structs
 - Check all networking code
 - Use host-to-network family of routines, i.e. htons()

int32_t i = 1025

Little Endian

char*[0] = 0x01

char*[1] = 0x04

char*[2] = 0x00

char*[3] = 0x00

Big Endian

char*[0] = 0x00

char*[1] = 0x00

char*[2] = 0x04

char*[3] = 0x01

Identification of Dependencies

- Make a detailed list of what you really need
 - OS
 - Tools
 - DLLs
 - Drivers
- Intel works/ed with hundreds of ISVs to get application migrated to Itanium® Architecture
 - Check with ISV on availability
 - Intel contact may be able to help
 - Intel has programs to enable ISVs in all GEOs
- Intel and OEMs might be able to loan development HW

Middleware for Itanium[®] Architecture

- Java
 - Not different from IA32 (no data type size dependence)
 - But look at JNI calls
 - Recommended JVM: BEA JRockit
 - Latest version: 5.0 R26 available for Itanium
 - In-depth tuned for Itanium[®] architecture
- MONO / .NET alternative
 - Available for Itanium[®] Linux too

Broad Ecosystem Support

Application Choice



- >5500 native applications
- 32-bit application support with IA-32 Execution Layer

Operating System Choice



- Windows*, Linux*, Unix*, & VMS support

System Vendor Choice



- Broad selection from top global & regional OEMs
- 2-way to 512-way systems
- >15 large SMP systems

*Other names and brands may be claimed as the property of others.

Industry Leaders Spearhead New Itanium® Solutions Alliance



Services to Help Migrate

- “Standardized” Linux – the Linux Standard Base (LSB) www.linuxbase.org
- Developer Resources
 - HP: www.hp.com/go/linuxdev
 - Intel: www.intel.com/software
 - Red Hat: www.redhat.com/developer
 - Novell: developer.novell.com/linux

Resources

- “Solaris to Linux Porting Guide” – www.hp.com/go/LinuxDev
- Solaris to Linux Porting Kit – www.hp.com/go/STK
- devresource.hp.com/linux – developer discussion forums
- “Itanium Architecture for Software Developers”, Philip Ezolt
 - www.intel.com/intelpress: ISBN 0-97-028464-0
- “Optimizing Linux Performance”, Walter Triebel
 - www.hp.com/hpbooks: ISBN 0-13-148682-9

Summary

- Plan and prepare your migration to Itanium® Architecture
- The “Code Clean” part can be most time consuming
- For your source code, 64Bit Itanium® is not really different from 64bit x86-64 architecture
- Tuning, documentation and availability of developer tools might make porting to Itanium even much easier than for other 64bit architectures
- Intel, OEMs and OSVs offer a lot of resources and support to make migration a smooth process

Questions?

