

您的潜力. 我们的动力

Microsoft[®]
微软(中国)有限公司

64位开发的今天与明天

Lin Hua (蔺华)
Developer Evangelist
DPE
Microsoft China

Who's Lin Hua?

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司



This is Lin Hua!

Webcasts

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

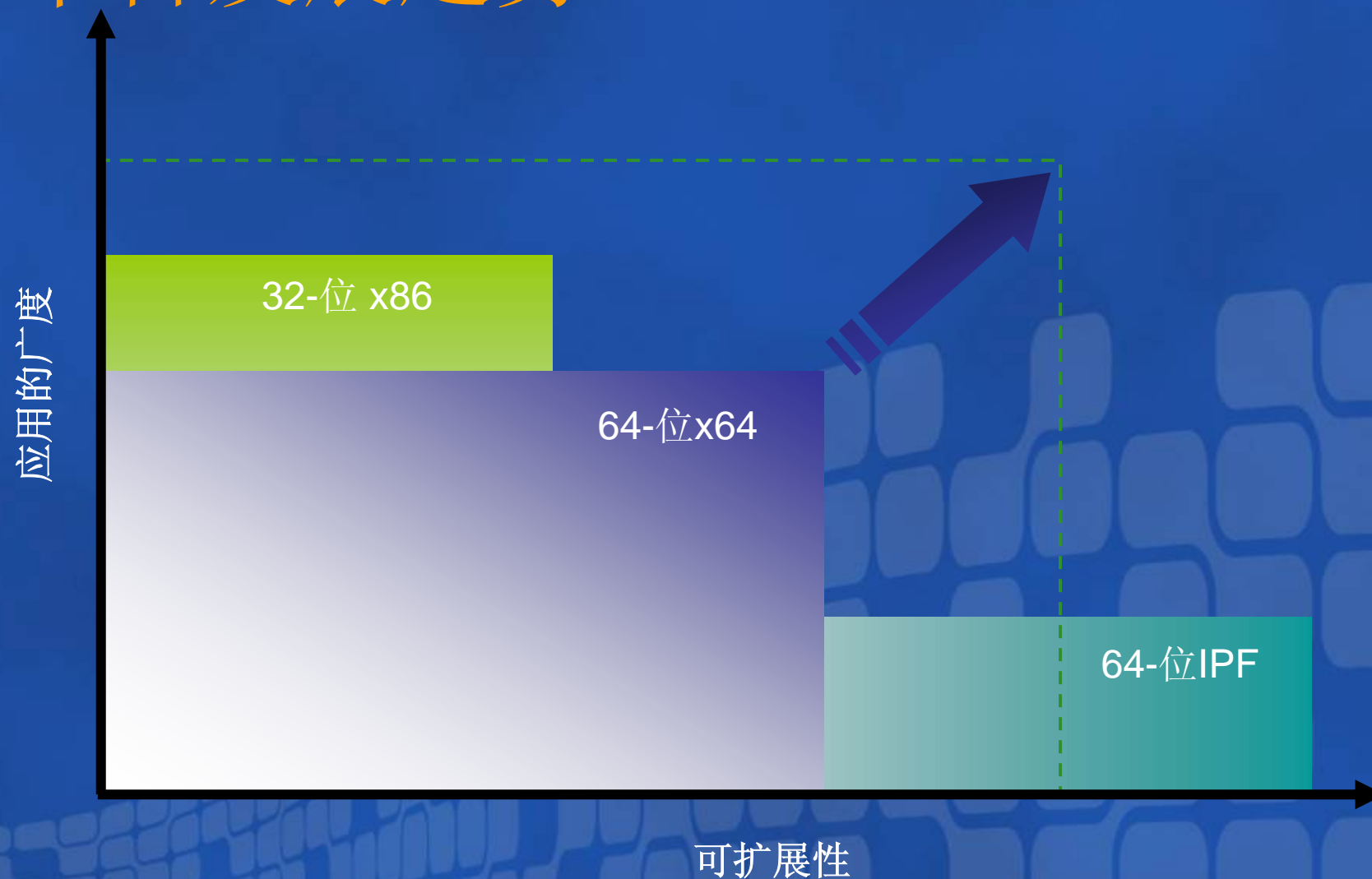
讨论议程

- 为什么要迁移到64位
- 今天: 64-bit Native C++
- 明天: 64-bit .NET Framework

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

平台发展趋势



微软内部的64位应用

- ◆MSN Search 已经把整个搜索引擎都部署在数千台Windows x64上
- ◆Microsoft.com已经把整个网站都迁移到x64上,该网站每天有1千3百万用户访问
- ◆MSN Messenger的服务器在迁移到x64之后,获得了10倍的性能提升
- ◆微软内部的财务系统已经全面的移植到了x64, 查询效率提升67%
- ◆超过5000微软员工已经在Windows XP Professional x64 Edition上进行日常工作

支持的体系架构

x64

(64-bit 扩展)

- 基于x86指令集的64位扩展
- 使得客户们在充分利用已在32位Windows应用程序上的投入的同时,也能够获得最新的64位技术所带来的高性能



IPF

(Itanium Processor Family)

- 基于显式并行指令集运算 (Explicitly Parallel Instruction Computing: EPIC) 架构
- 运行在Windows的平台上, 带给客户们最高级别的性能和可扩展性



您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

Windows 64-bit Memory

Architectural Component	Windows 32-bit	Windows 64-bit
Virtual memory	4 gigabytes (GB)	16 terabytes
Paging file size	64 GB	512 terabytes
Paged pool	470 megabytes (MB)	128 GB
Non-paged pool	256 MB	128 GB
System cache	1 GB	1 terabytes
System Page Table Entry (PTE)	1.2GB	128GB

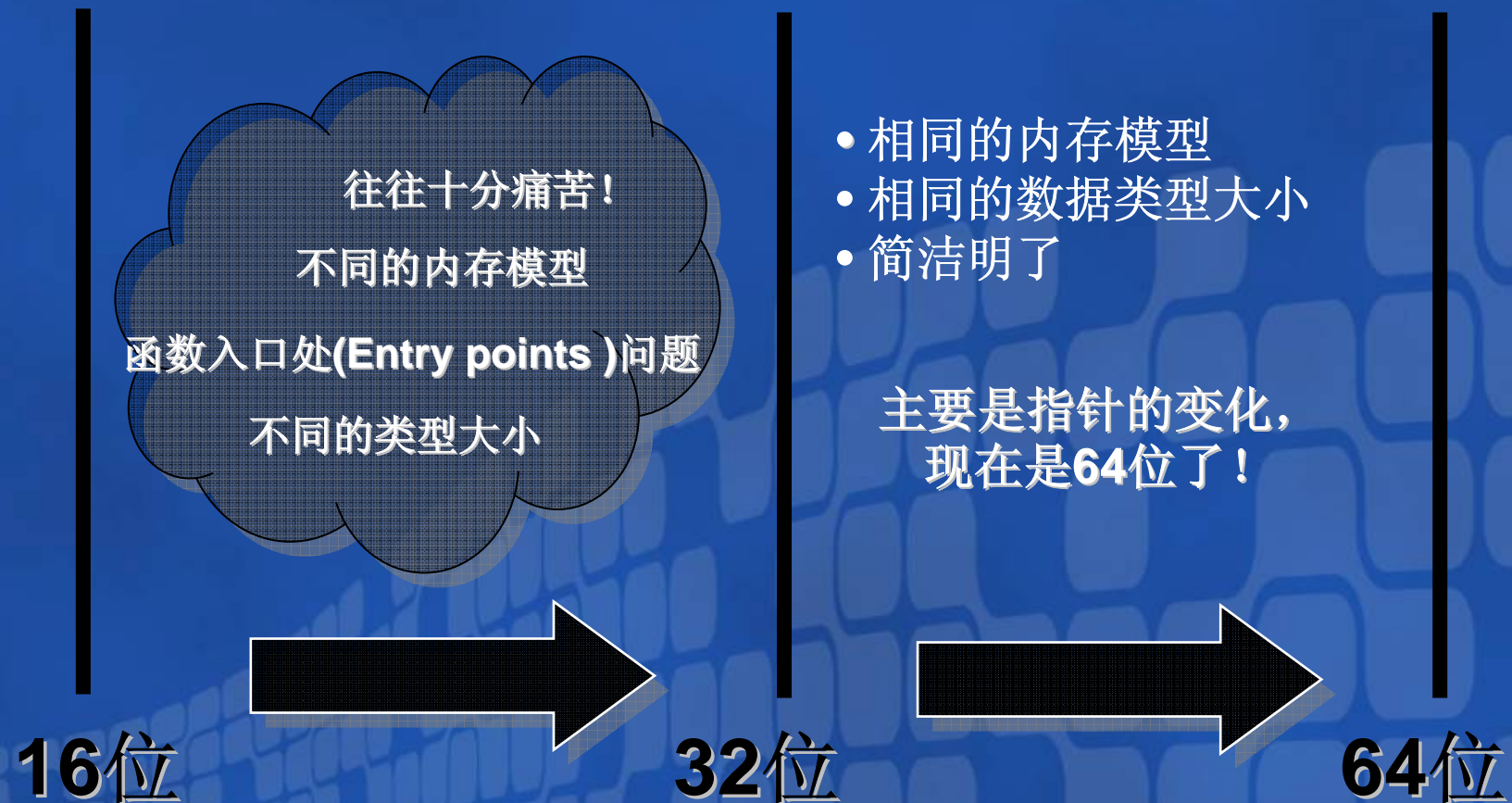
$2^{64} \gg 2^{32}$

18,446,744,073,709,551,616 \gg 4,294,967,296

16位-->32位-->64位

16位到32位转换过程

32位到64位转换过程



您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

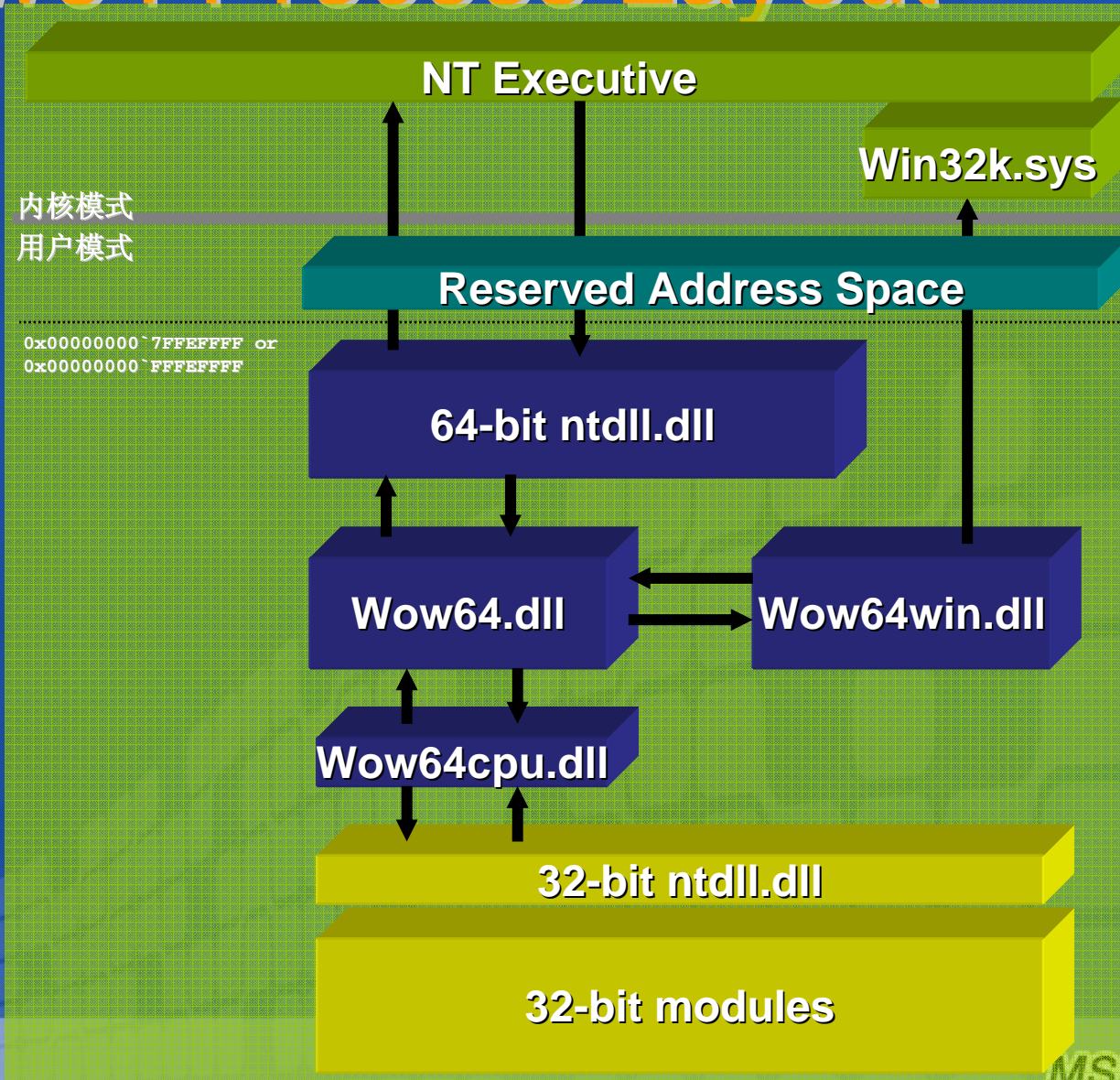
讨论议程

- 为什么要迁移到64位
- 今天: 64-bit Native C++
- 明天: 64-bit .NET Framework

WOW

- Windows on Windows 64 (WOW64)
- 隔离运行32位应用
 - 特别的文件和注册表隔离
 - Console
 - GUI
 - Services
- 单独的DLL
- 该子系统是运行在用户模式的

Wow64 Process Layout



当前的开发模型

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

基于x86的计算机 (或者使用 Wow)



- 在这里开发和测试**32**位代码
- 跨平台编译成**64**位二进制文件
- 把可执行文件发送到**64**位机器

64位计算机



- 在这里执行代码
- 继续调试遗留的**bug**, 如果有.....

把你的应用迁移到64位

- 使用编译器开关 (-Wp64)
- 在64位中的一些不同 – LLP64
 - `T *`, `size_t`: 64 位
 - `int`, `long`: 内建的数值都是32位的
 - `long long`: 永远是64位
- 同一个进程中不能混合64位和32位的代码
- 使用几乎与Win32相同的API
 - 除了少数几个API不同之外 (如`GetClassLong/SetClassLong(_ptr)`)

API	Data Model	int	long	pointer
Win32	ILP32	32	32	32
Win64	LLP64	32	32	64
UNIXes	LP64	32	64	64

调整相应的代码

- 指针的大小是64位的 (int, long还是32位)
- 使用size_t和新的多态数据类型
 - INT_PTR, LONG_PTR, DWORD_PTR...
- 新的系统API的修改所引起的改动
- 常量, 特别是16进制和2进制表示的值很容易出错
- 嵌入的汇编代码需要移除

C++代码例子

32-bit:

```
int x, y, z;  
x = (int) &y;  
z = cArray.GetSize();
```

64-bit:

```
INT_PTR x, y, z;  
x = (INT_PTR) &y;  
z = cArray.GetSize();
```

C++代码例子 (续)

32-bit:

```
char charArray[3], *char_ptr;  
char_ptr = (char *) ((int)charArray + 1);  
*char_ptr = 'a';
```

64-bit:

```
char charArray[3], *char_ptr;  
char_ptr = (char *) ((size_t)charArray+1);  
*char_ptr = 'a';
```


C++代码例子 (续)

32-bit:

GetClassLong / SetClassLong

GetWindowLong / SetWindowLong

64-bit:

GetClassLong_ptr / SetClassLong_ptr

GetWindowLong_ptr / SetWindowLong_ptr

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

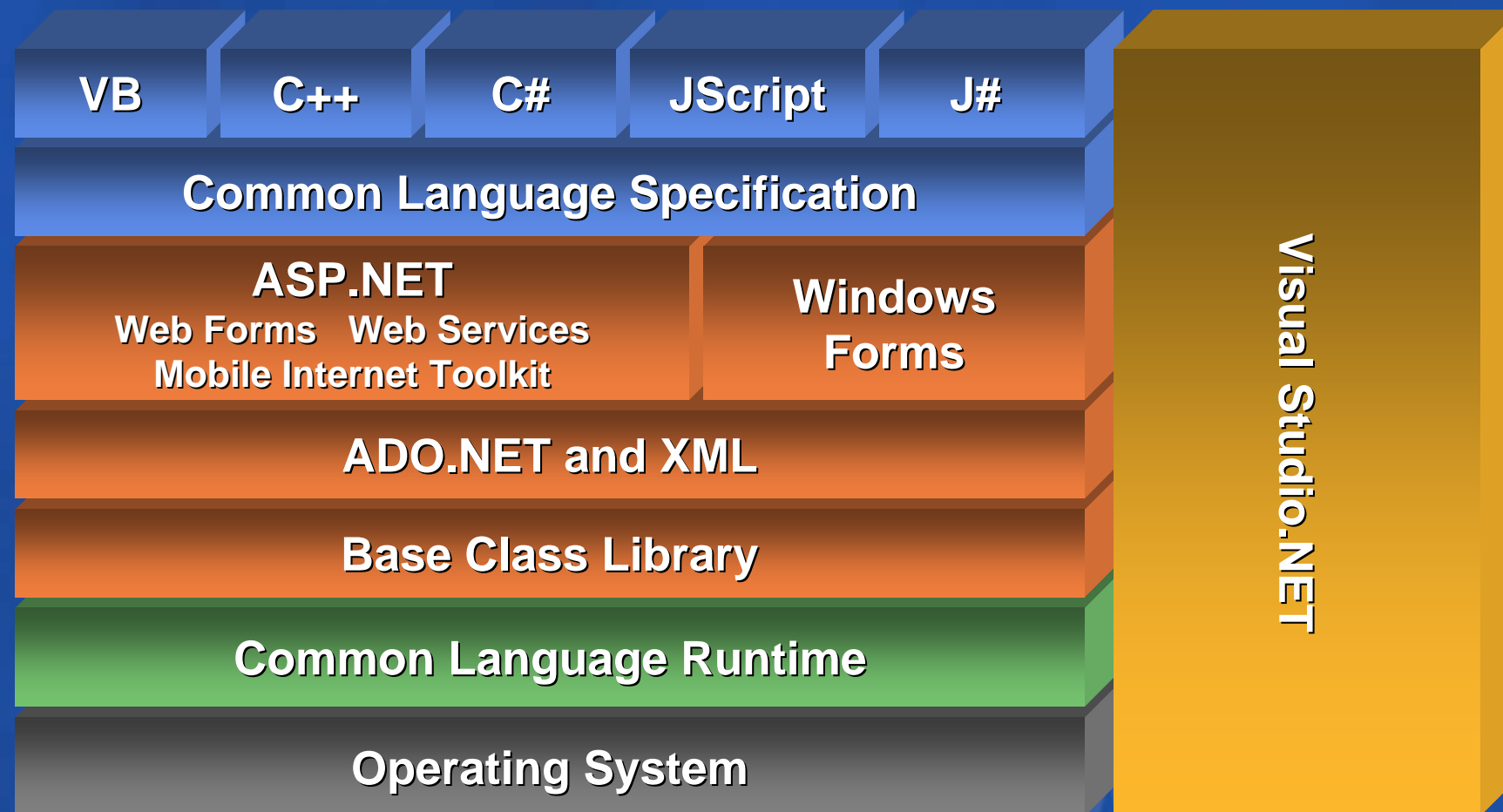
讨论议程

- 为什么要迁移到64位
- 今天: 64-bit Native C++
- 明天: 64-bit .NET Framework

.NET 概括

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司



开发工具路线图

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

Visual Studio .NET 2003

- “Everett Release”
- Windows Server 2003 integration
- Support for .NET Compact Framework and device development
- Improved performance

Visual Studio 2005

- “Whidbey” release
- SQL Server 2005 integration
- 64 bit Framework
- Improved IDE productivity and community support
- Extended support for XML Web services
- On...

Visual Studio “Orcas”

- Windows “Longhorn” integration
- New UI tools and designers
- Extensive managed interfaces (WinFX)

64位上的.NET

- 开发工具
 - Visual Studio 2005
 - 今年年底正式发布
 - 现在已经发布beta2版
- 处理器支持
 - 包括 x64 和 IPF
 - 当然也包括32位的处理器
- 操作系统
 - Windows Server 2003 SP1 (all 64-bit editions)
 - Windows XP 64-bit Edition

VS.NET 2003中的编译器

- VB, C#, J#
 - **PE32, ILONLY**,可验证性
 - C# 也能编写unsafe code
 - 在64位上, Dll Import 签名可能不正确
- C/C++
 - 混合模式的映像/应用运行在WOW64
 - 也可以用C++托管扩展创建一个托管应用
 - 指针算法是不可验证的

VS .NET 2005中的编译器

- VB, C/C++, C# 可以编译所有架构的应用
 - C# 中新的开关: **/platform**
 - **anycpu, x86, Itanium, x64**
 - C/C++中新的开关: **/clr**
 - **safe, pure**
 - 所有编译器都能产生 **PE32** 或者 **PE32+**
- J# 生成 **PE32, ILONLY** 和 **32BITREQUIRED**

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

demo

Visual Studio .NET 2005



MSDN Webcasts

64位上的托管代码

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 可验证性 代码可以运行
 - 不用重新编译
 - 32-bit .NET 代码可以运行在 WOW64上
 - 64-bit .NET 代码可以本地运行
- 不可验证代码可能需要一些调整:
 - **COM Interop, P/Invoke**
 - Assemblies 含有本地代码 (e.g. Managed C++)

编写可移植性的托管代码

- Verifiable
 - 不要生产本地代码 (IL only)
 - 不要有指针算法
- 遵循互操作规则
- 了解两种架构的不同之处
- 谨慎处理浮点运算

遵循互操作规则

- 签名
 - DllImport, Declare, #include
 - System.IntPtr
 - 在相应的平台上有相应的COM组件及其API
- Marshaling
 - StructLayoutAttribute
 - Marshal.SizeOf

其他问题

- 浮点类型的直接比较
- API改变
- Serialization
 - XML serialization没有任何的问题
 - Binary serialization可能是有问题的
 - 在64位的CLR中, **System.IntPtr** 的大小改变了
 - Alignment不需要考虑

您的潜力. 我们的动力

Microsoft[®]
微软(中国)有限公司

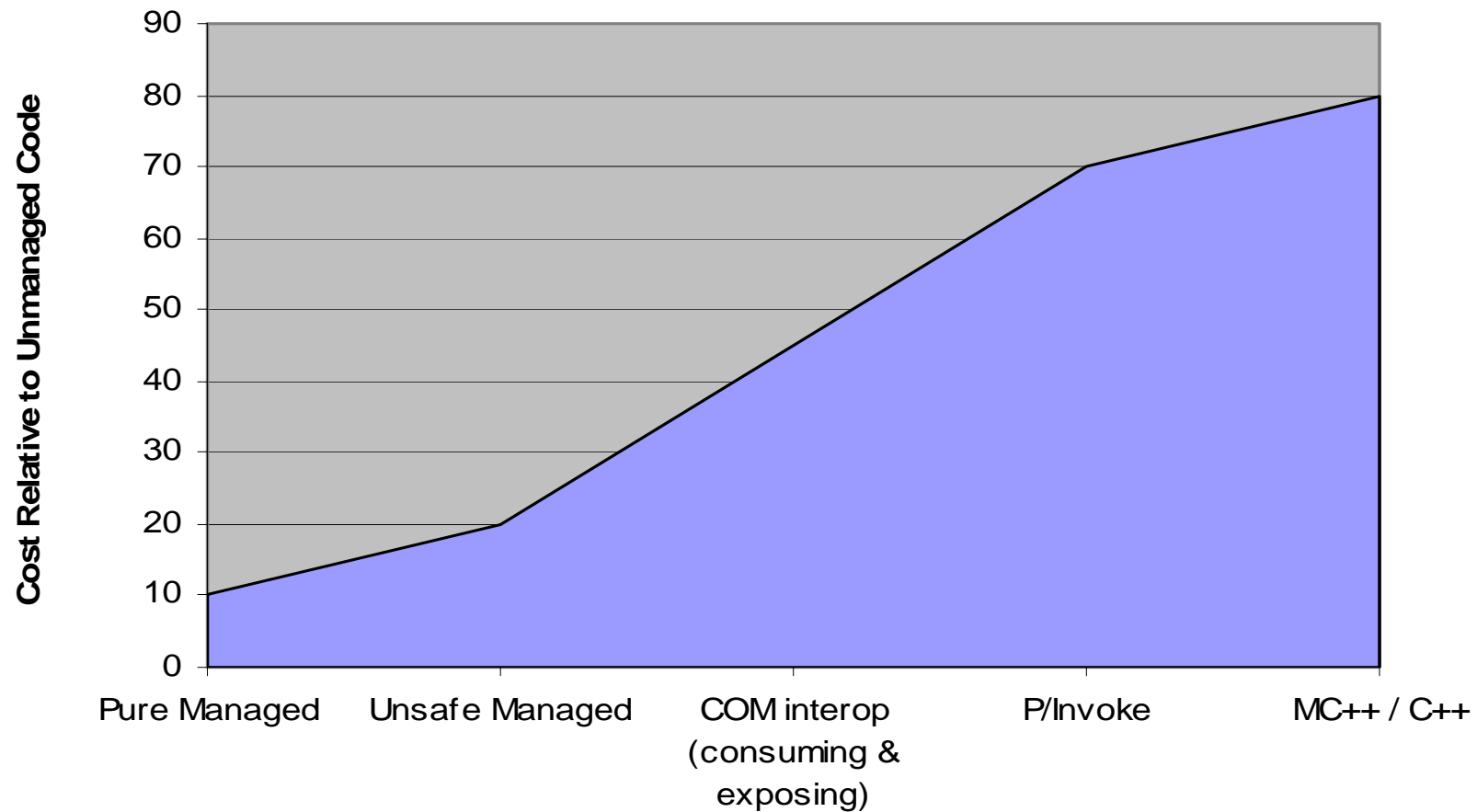
demo

COM Interop

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

程序移植成本



总结


- 编写x64程序和32位应用程序几乎是一样的
- 注意数据类型和API的改变
- 不是所有的代码都需要移植
- 代码的移植的主要精力放在对64位指针上
- .NET程序移植最为简单直接
- 应用迁移中的互操作是需要仔细注意的
- 这个全新的平台将让我们可以做更多的事情

Q&A


您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问 

提问(A)

删除(D)

问题管理器(Q)

您的潜力. 我们的动力

Microsoft®
微软(中国)有限公司

Microsoft®

Your potential. Our passion.™

© 2004 Microsoft Corporation. All rights reserved.

This presentation is for informational purposes only. Microsoft makes no warranties, express or implied, in this summary.



MSDN Webcasts