IBM System i™

32 bits. vs. 64 bit
Why is that Interesting ?

Erik Rex
Cert. Consult. IT Specialist
System i
rex@dk.ibm.com

i want stress-free IT.
i want control.
i want an i.

IBM System i™

32 bits. vs. 64 bit
Why !!! -is that Interesting ?

Erik Rex
Cert. Consult. IT Specialist
System i
rex@dk.ibm.com

i want stress-free IT.
i want control.
i want an i.

Intro

- **S/38 was born 1978 with 48 bit**
- **AS/400 in 1988 inhered 48 bit**
- **AS/400 migrated 1995 HW, OS and Appl. to 64 bit**
  - CISC → RISC (ver.3.6)
- **Everybody alse was running 16 bit or 32 bit**

64 bit… What are you talking about ???

- **AS/400 was a ’Full 64 bit computer in HW and SW**
- **We were very proud of our 64 bits, and we talked a lot about it, and everybody else would either try to neglect or strive to be able to do the same**

- **But… if you ask, what are you actually talking about, very few know what it is**

Addressing possibilities

- **16 bit – 65.536**
- **32 bit – 4.294.967.296**

- **48 bit – 281.474.976.710.656**
- **64 bit – 18.446.744.073.709.551.616**
  - 18.4 Quintillion bits or 16.777.216 Terabytes

Definition of an 64 bit Computer

- **A 64 bit processor is able to address 64 bit of Virtual Address spaces.**
- **Can calculate and store data in 64 bit format.**
- **General Purpose Registers and Arithmetic logical units are 64 bit wide.**

What it adds

- **64 bit is scalability**
  - Address possibilities
  - SMP
  - Transactions
  - More capacity
  - Better design ~ more mph

64 bit Misconception

- **Not !**

  2 x faster than 32 bit system

What does it mean for the business

- **Large file support**
- **Large physical memory support**
- **Large 64 bit application Virtual address Space**
- **64 bit Integer computation**

Complementary

- **64 bit is complementary to 32 bit**
  - True for AIX Power and Windows
  - Different 64 bit HW and different 64 bit versions of OS
  - AS/400 \ iSeries\ i5 creates 64 bit applications on the fly

… unless

- No 64 bit OS or driver or application confusion

Virtual Memory

- **1. time used in 1961 in an Atlas Computer (UK)**
- **System/3 oldies perhaps remember ‘Overlay programming’**
- **Virtual Memory = Swap file (windows/Unix)**
- **Virtual memory ~ Single Level Store (i5/OS)**

Single Level Storage

- **The 5. element of the ‘Sacred Principles’**
  - Large address is a side price
  - Task Switching is primary goal
  - Address switching/sharing
  - Performance
- **Single Level Store vs. Virtual Memory**
  - One copy of the object in memory
  - Data and programs will be copied to memory
  - Instant access of other users

the ‘Sacred Principles’

Technology independence

Object-based design

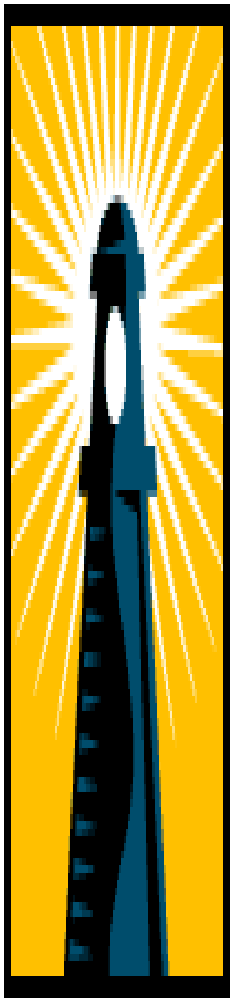
Hardware integration

Software integration

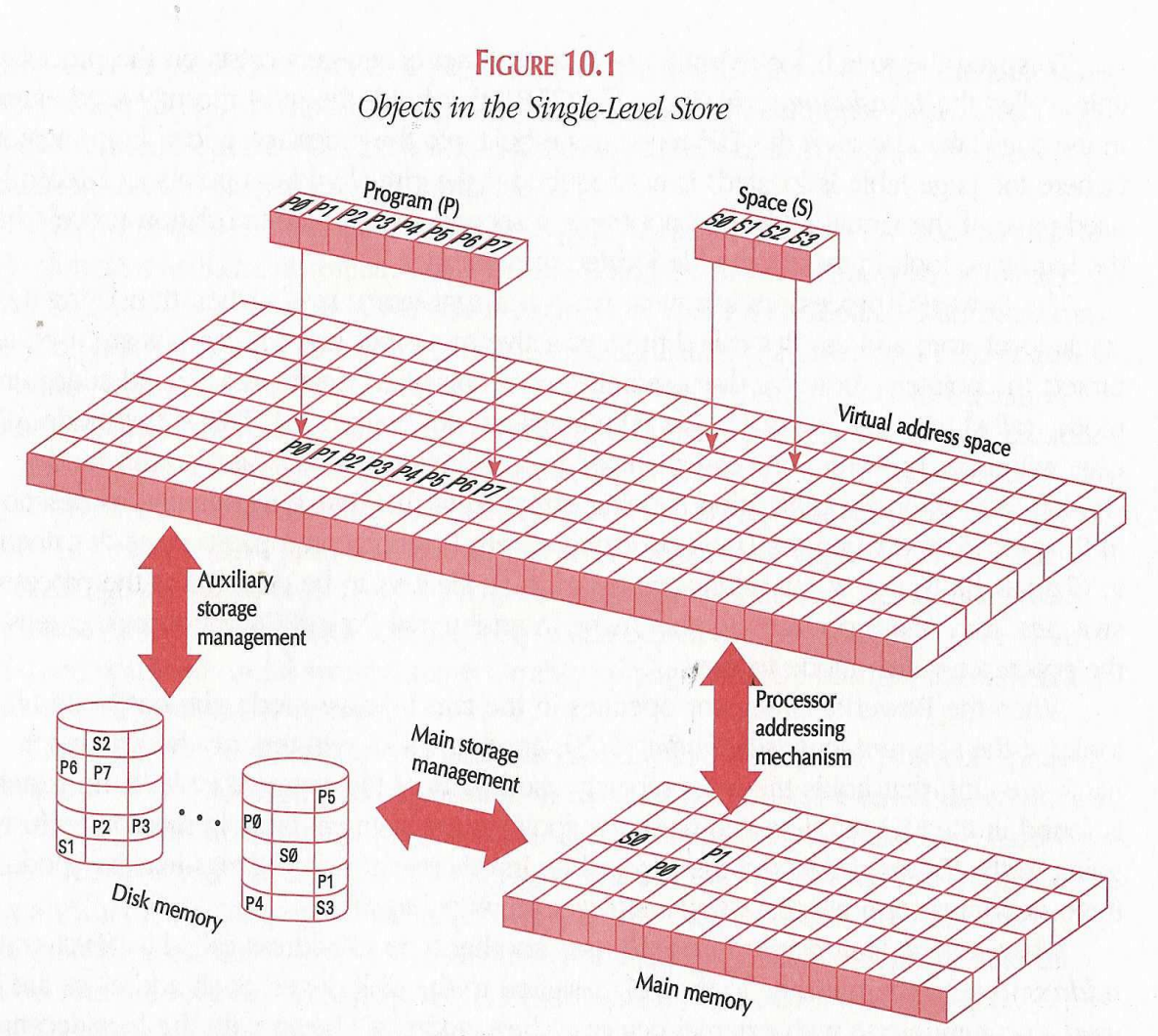
Single Level Store

Single Level Storage

- **The 5. element of the ‘Sacred Principles’**
  - Large address is a side price
  - Task Switching is primary goal
  - Address switching/sharing
  - Performance
- **Single Level Store vs. Virtual Memory**
  - One copy of the object in memory
  - Data and programs will be copied to memory
  - Instant access of other users



Single Level Store – the big picture



Single Level Store – The Business

- You don't know the technology – system does
- You don't know Program and Data location
- You only ref. objects by Name
- You don't worry about extending files
- You don't care about Disk Space
- You don't redistribute data
- You don't need a DBA
  - Try this on UNIX or Windows

32 bit ~ 64 bit ~ Single Level Store

- **Impressing that we all know… because**
  - OS does not know
  - Compilers don’t see it
  - Application programs don’t care
    - Only SLIC knows and Care

Surprise

- **I5/OS and Application programs address Objects**
  - Use 16 bytes ~ 128 bit
  - Used since 1978

- **Now could be the time to introduce the System i as an 128 bit server**

32 bit on 64 bit System i

More

Time

?

New 32 bit JVM Improves Development Options

- A new 32-bit i5/OS JVM V1.5 is expected to reduce the memory footprint for Java applications, which is especially valuable on smaller systems
  - Note: JVM 1.5 will be supported with WebSphere Application Server 6.1*
- Implements a standard IBM JVM with consistent tooling and tuning options
- Simple to implement without code changes for the majority of applications
- The classic 64-bit JVM will continue to offer best scalability and performance for large enterprises with i5/OS V5R4

Notes

i5/OS V5R4 provides an option to use a new Java Virtual Machine that is being deployed across all IBM platforms. The new and more compatible i5/OS JVM is expected to help ISVs with application portability as well as configuration and service tools compatibility as they deploy new Java-based applications on i5/OS that were written on another platform. Providing the option of a 32-bit JVM in i5/OS should also help ISVs reduce the memory footprint and, in some cases, improve the performance of their applications on smaller iSeries servers.

Note that the new 32 bit version of the JVM will be supported with IBM WebSphere Application Server 6.1 that is planned for delivery later in 2006.

It is expected that larger customers will continue to use the scalable, classic 64-bit JVM (recently enhanced to JDK 1.5).

IBM Technology for Java Virtual Machine

- **I5/OS V5R4 delivers two JVM and SDK environments:**
  - Java ™ 2 Software Development Kit, Standard Edition
  - "Classic", running in 64-bit mode
  - "32-bit" IBM Technology for JVM runs in 32-bit mode
- **"32-bit" implements a standard IBM JVM 1.5 runtime environment and SDK:**
  - Specifically focused for applications on smaller systems, with less impact on memory utilization
  - Simple implementation, mostly without code changes for the majority of the applications
  - Affects management of storage references
- **The 64-bit JVM continues to offer the best scalability and performance for large enterprises with i5/OS V5R4**

Notes: IBM Technology for Java Virtual Machine

IBM System i

Comparing the 32-bit JVM with the 64-bit one must include the non functional requirements such as scalabilityand predictability needs for the application. It is clear that a 32-bit JVM needs less space to address objects,but it is also evident that the addressability is smaller. So, as on any platform, one needs to closely examine the application architecture, its functionality and the non functional requirements if asked to select one of the implementation ways.

Apart from the teraspace addresses needed by native methods when migrating to the 32-bit JVM, there are some other remarks regarding performance: ILE native method calls will typically be slower, but are currently a focus item within development. However, since i5/OS PASE native methods use the same address space as the 32-bit JVM, they will execute faster. As always, the overall performance results will highly depend on the application being used.

Remember also that only JIT or interpreted mode are accepted, which makes the use of optimization levels on the CL commands associated with the JVM no longer relevant.

To set the compiler to use JIT: □ From a command line prompt on your iSeries™ server, add the environment variable by using the Add Environment Variable (ADDENVVAR) command. Then, run your Java program using the Run Java (RUNJVA) command or JAVA command. For example, use: ADDENVVAR ENVVAR (JAVA_COMPILER) VALUE(jitc) JAVA CLASS(Test) □ Set the java.compiler system property on the iSeries command line. For example, enter JAVA CLASS(Test) PROP((java.compiler jitc)) □ Set the java.compiler system property on the Qshell Interpreter command line. For example, enter java - Djava.compiler=jitc Test By default, WebSphere Application Server runs in JIT mode.

WAS V6 and 64-bit platforms: Key Advantages

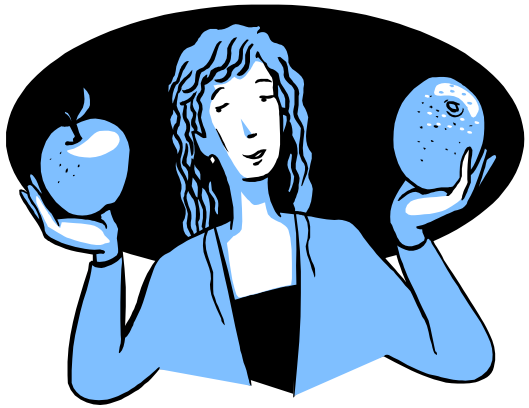
- 64-bit enabled WAS products leverage 64-bit performance features
  - Capability of Java™ heaps much larger than the ~2Gb limits of the 32-bit platforms
  - Transparent IBM Just-in-Time (JIT) compiler enhancements that generate optimized machine code to directly leverage 64-bit platform performance features
    - Usage of extra 64 bit registers
    - Extended machine instructions and precision computations

WAS and 64-bit performance : Downside

- **The reality of WAS 64-bit performance is a mixed bag**
  - Applications capable of taking advantage of WAS 64-bit features can see significant performance gains
  - The downside for WAS 64-bit applications is a significantly greater memory footprint that can lead to performance loss
    - All address references are 64-bit wide, twice the size of address references in 32-bit deployments
    - This results in an increased memory footprint reducing hardware cache efficiency and thus performance

Cross-Platform Comparisons

- **Be careful with cross-platform comparisons**
- **Differences exist across platforms**
- **Hardware examples are**
  - CPU clock speed
  - Number of processors
  - Bus speed
  - L2 & L3 cache
  - Disk seek access time
  - I/O wait times
  - Tuning
- **Software examples are**
  - OS paging
  - Process priorities
  - Tuning
- **Be careful drawing conclusions about one platform versus another**



Is 64-bit for me ? – The Checklist

- ✓ **Does your application on 32 bit systems need a Java heap much greater than ~2GB for higher performance? If Yes → WAS 64-bit**

  (on existing 64 bit systems 4-5 GB)

- ✓ **Does your application use computationally intensive algorithms for statistics, security, encryption, etc which can benefit from high precision computation support? If Yes → WAS 64-bit**

- ✓ **Does your application need a Java heap just a little more than ~2GB i.e. 2~3GB ? If Yes → WAS 32-bit on 64-bit platforms**

- ➢ **If you answered "No" to all above questions, then WAS 32-bit on 64-bit platforms is better suited for you**

Thanks