WebSphere Application Server

IBM WebSphere Application Server and 64-bit platforms:
64-bit Performance Demystified

J. Stan Cox (stancox@us.ibm.com)
Piyush Agarwal (agarwalp@us.ibm.com)

Our Speakers Today

Piyush Agarwal
WAS Performance Analysis
Research Triangle Park, NC
agarwalp@us.ibm.com

J. Stan Cox
WAS Performance Analysis
Research Triangle Park, NC
stancox@us.ibm.com

Roadmap of today's presentation

- WAS and the 64-bit platforms supported
- Key Advantages of WAS on 64-bit platforms
- Downside of the 64-bit platforms
- Scalability and other useful information about 64-bit platforms
- How to decide if 64-bit is for you?
- Closing Notes
- Resources

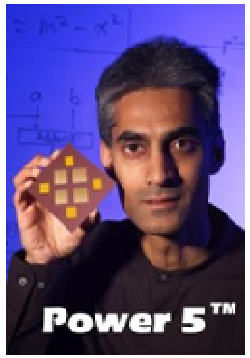
Roadmap

- ✓ **WAS and the 64-bit platforms supported**
  - V6.0.1, V6.0.2 and V6.1 (and V7)
  - 64-bit platforms: POWER™, x86-64, SPARC™
- Key Advantages of WAS on 64-bit platforms
- Downside of the 64-bit platforms
- Scalability and other useful information about 64-bit platforms
- How to decide if 64-bit is for you?
- Closing Notes
- Resources

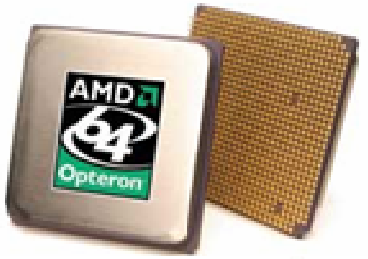
WAS and the 64-bit platforms supported

- IBM® WebSphere® Application Server (WAS) 6.0.x introduces support for 64-bit platforms
  - **V6.0.x**
    - Linux® on x86-64 platform, on POWER™ platform (LoP) and on IBM System z™
    - Windows® Server 2003 Enterprise x64 Edition on x86-64 platform
    - AIX® on POWER platform
    - HP-UX on Itanium 2
  - **V6.1**
    - All the V6.0.x platforms
    - Solaris 10 on SPARC™ and x86-64
    - All IBM platforms now run IBM J9 (5.0) JVM for both 32-bit and 64-bit

WAS V6 and 64-bit platforms

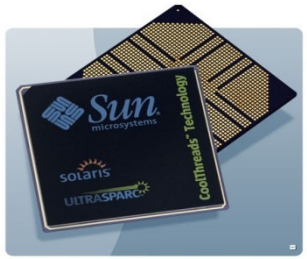


- POWER™
  - **P**erformance **O**ptimization **W**ith **E**nhanced **R**ISC
  - Supports both 32-bit and 64-bit computing at full native performance (not an emulation layer)
  - Simultaneous multithreading (SMT), large number of registers (120 GPRs, 120 FPRs)
  - Large caches, very high chip-memory bandwidth
  - Virtualization, micro-partitioning



- x86-64
  - In 2003, AMD introduced the x86-64 architecture as an enhancement to the Intel® x86-32 in their Athlon64™ and Opteron™ processors
  - Adopted by Intel under the name (EM64T) in newer Pentium™ and XEON™ processors.
  - Full native performance and compatibility for 32-bit x86 applications
  - Additional hardware registers available in 64-bit mode
  - 64-bit wide computing for faster high precision computations via double precision integer support and Instruction set extensions

- SUN SPARC™
  - **S**calable **P**rocessor **Arc**hitecture
  - Newer processors support high level of multithreading (8 cores/chip x 8 threads/core)
  - All Cores connected to memory and I/O subsystems through high bandwidth interconnect
  - Smaller caches but with high associativity

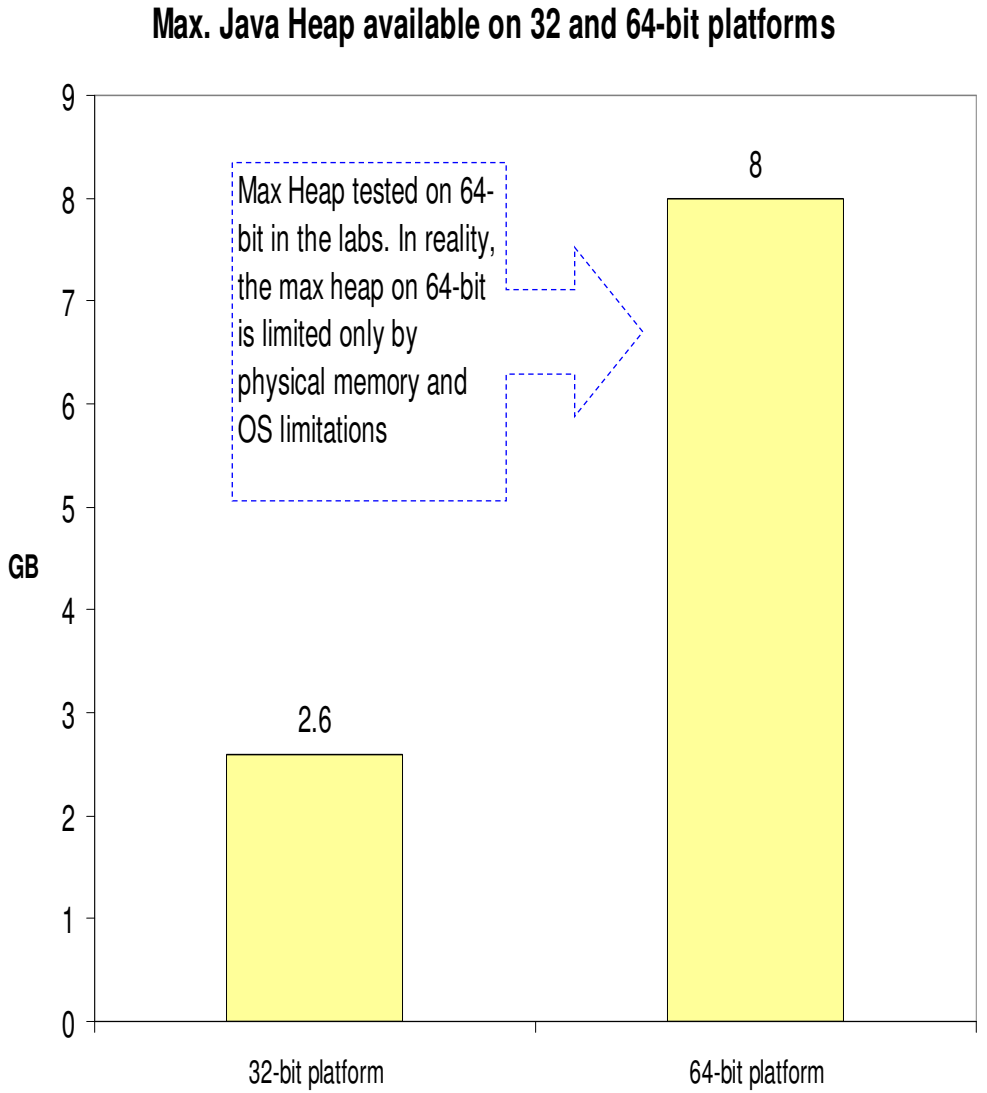


Roadmap

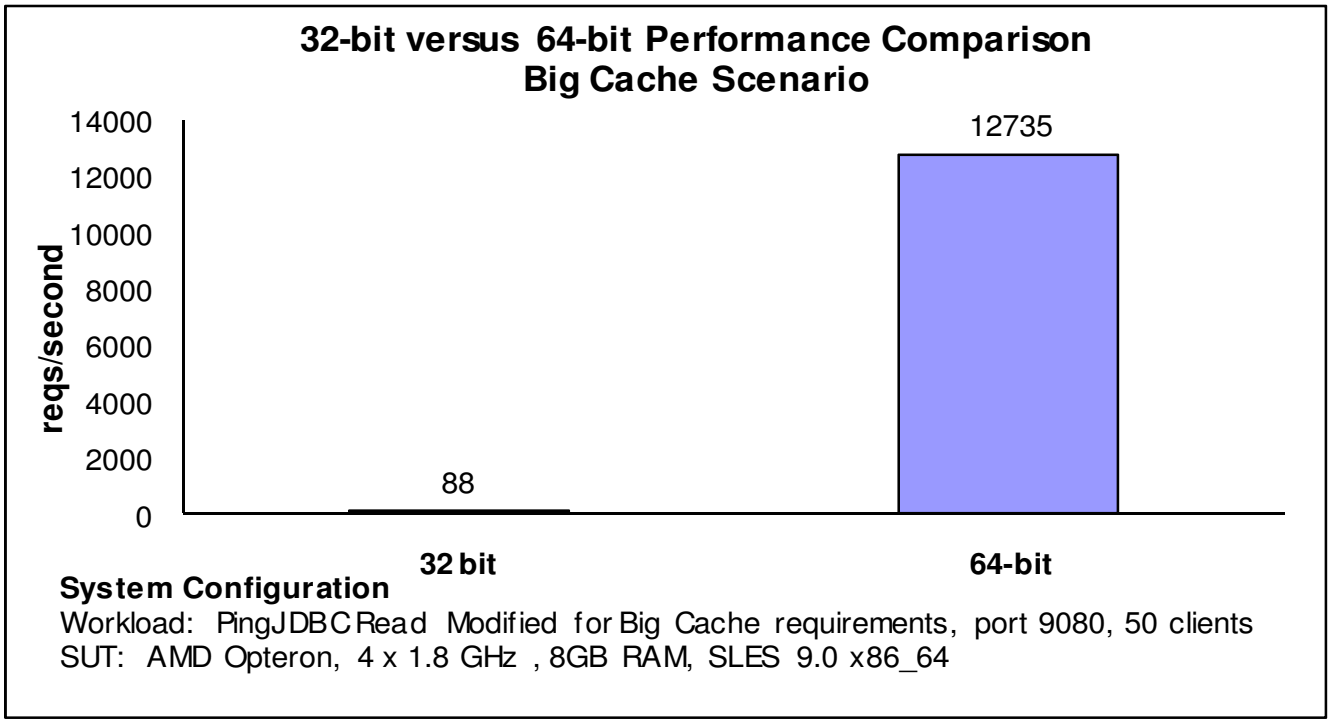
- ✓ WAS V6 and the 64-bit platforms supported
- ✓ **Key Advantages of WAS on 64-bit platforms**
  - Leveraging large heaps
  - Leveraging high precision computations and extra registers
- Downside of the 64-bit platforms
- Scalability and other useful information about 64-bit platforms
- How to decide if 64-bit is for you?
- Closing Notes
- Resources

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 registers (x86-64)
    - Extended machine instructions and precision computations



Large heaps can yield significant performance



- WAS 64-bit is able to allocate a heap large enough to cache the entire dataset
- WAS 32-bit cannot, causing repeated access to the database to fetch large objects over the network
- This example shows a “best-case” scenario for large heap usage by a WebSphere application.
  - Applications that require cache entry invalidations will typically not see performance gains that are this dramatic
  - Dynamic data is not generally 100% cache-able

IBM JVM 64-bit code generation optimizations

- **The IBM JVM and Just-in-Time (JIT) compiler automatically take advantage of 64-bit capabilities for all supported platforms.**
  - JVM supports large heap allocation (no artificial limit below OS/Platform capabilities)
  - IBM JIT generates machine code to take advantage of 64-bit instruction extensions and high precision computational features
  - IBM JIT leverages extra registers available in 64-bit mode on x86-64, reducing register spills, memory loads and stores, improving the bus utilization and overall memory performance

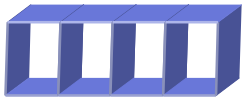
Roadmap

- ✓ WAS V6 and the 64-bit platforms supported
- ✓ Key Advantages of WAS on 64-bit platforms
- ✓ **Downside of the 64-bit platforms**
  - ➢ Implications of memory growth on the 64-bit platforms
- Scalability and other useful information about 64-bit platforms
- How to decide if 64-bit is for you?
- Closing Notes
- Resources

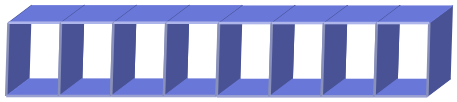
WAS and 64-bit performance : Downside

- 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

4-byte ptr address ref (in 32-bit)



8-byte ptr address ref (in 64-bit)



Downside of 64-bit platforms

Primitive fields are NOT references (4 bytes)

```
Public Class Foo
{
   private int A;
   private char[] B;
   private HashMap C;

}
```

Downside of 64-bit platforms

Array fields are references (8 bytes), compared to 4 bytes on 32-bit.

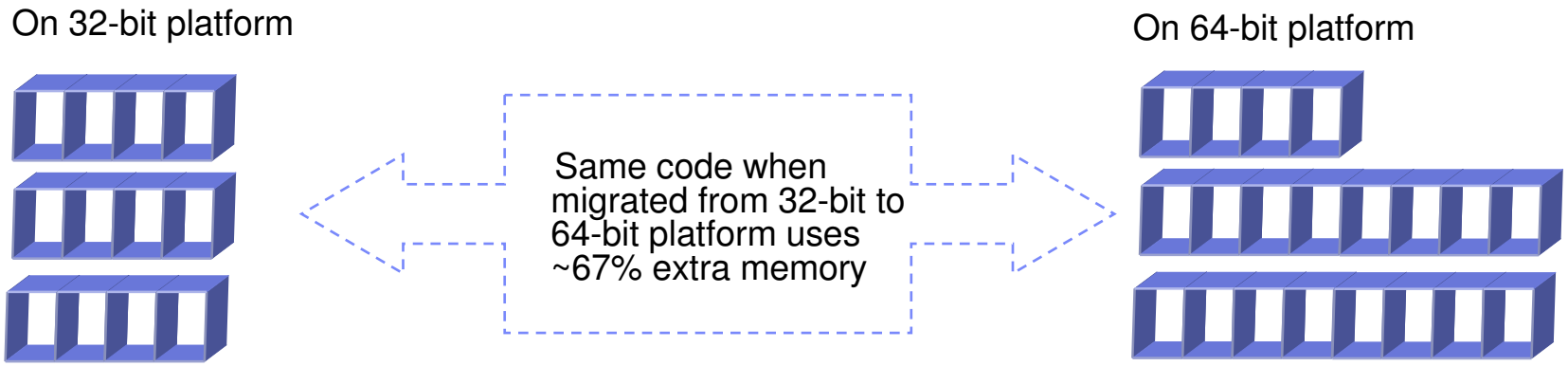
```
Public Class Foo
{
    private int A;
    private char[] B;
    private HashMap C;
}
```

Downside of 64-bit platforms

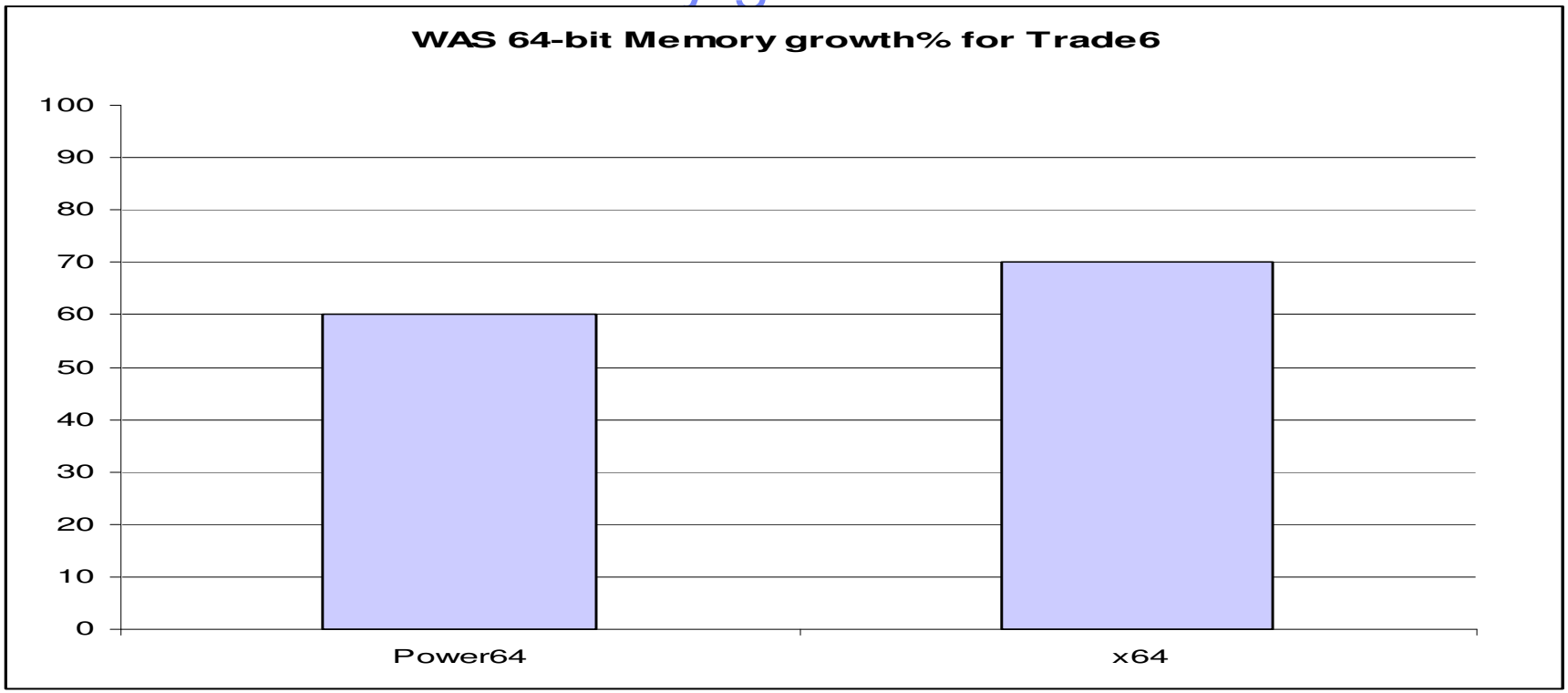
Object fields are references (8 bytes), compared to 4 bytes on 32-bit.

```
Public Class Foo
{
    private int A;
    private char[] B;
    private HashMap C;

}
```

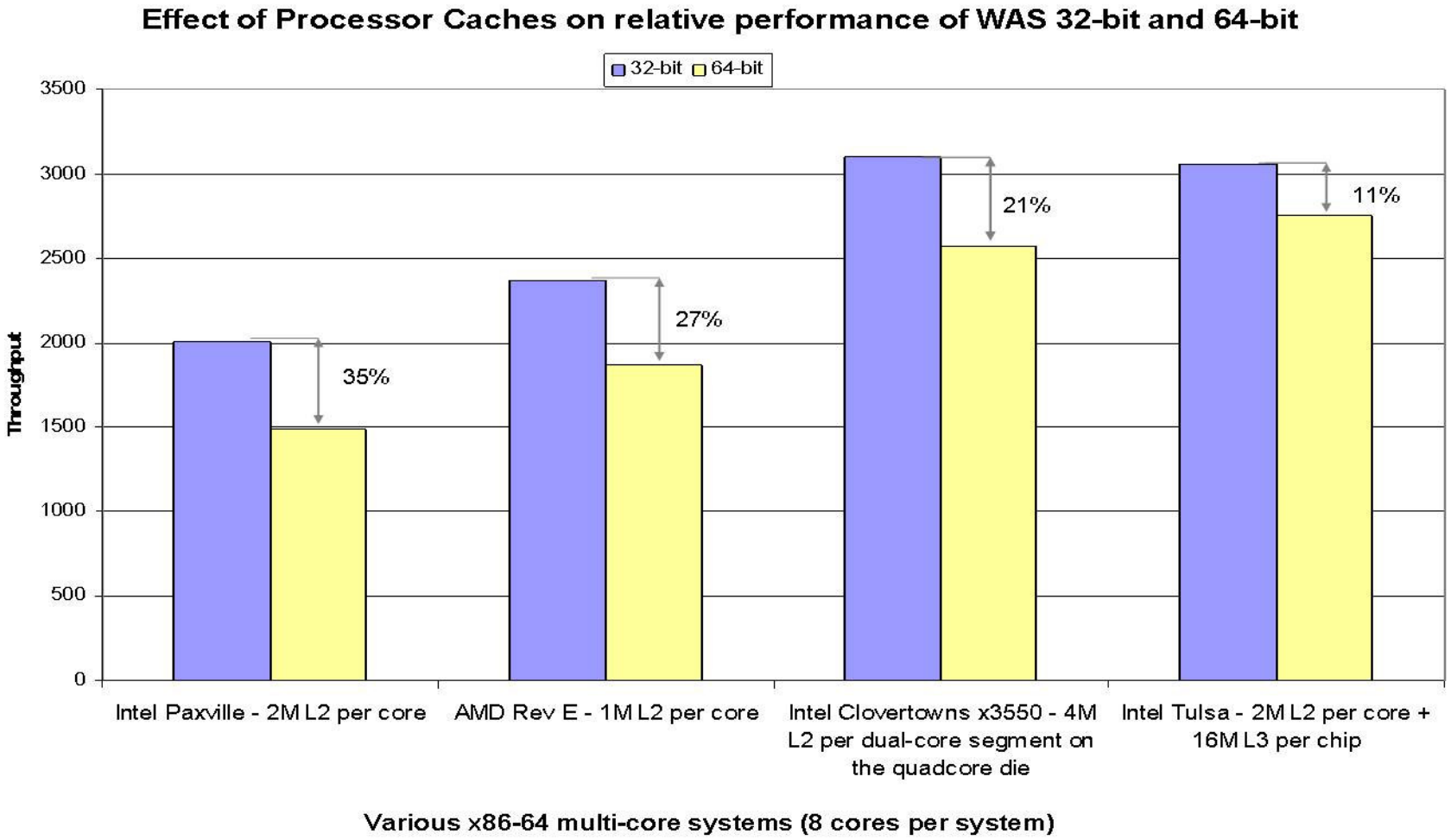


WAS and 64-bit memory growth



- In IBM Java for 64-bit all address references are 64-bit wide, twice the size of address references in 32-bit deployments
  - This results in a significantly increased memory footprint
    - POWER yields 60% memory growth for Trade6 on WAS 6.1
    - x86-64 yields 70% for Trade6 on WAS 6.1
  - Can reduce hardware cache efficiency, increasing CPI and reducing application performance

Processor cache size impact on 64-bit performance



- General performance expectation for applications that do not specifically leverage 64-bit features is a degradation of 10-35% when run on 64-bit compared to 32-bit platforms on WAS 6.1
- Processor cache size influences relative performance of 32-bit and 64-bit applications.
- Generally speaking bigger processor caches help 64-bit performance
- From a price/performance perspective, bigger hardware cache can buy you better performance

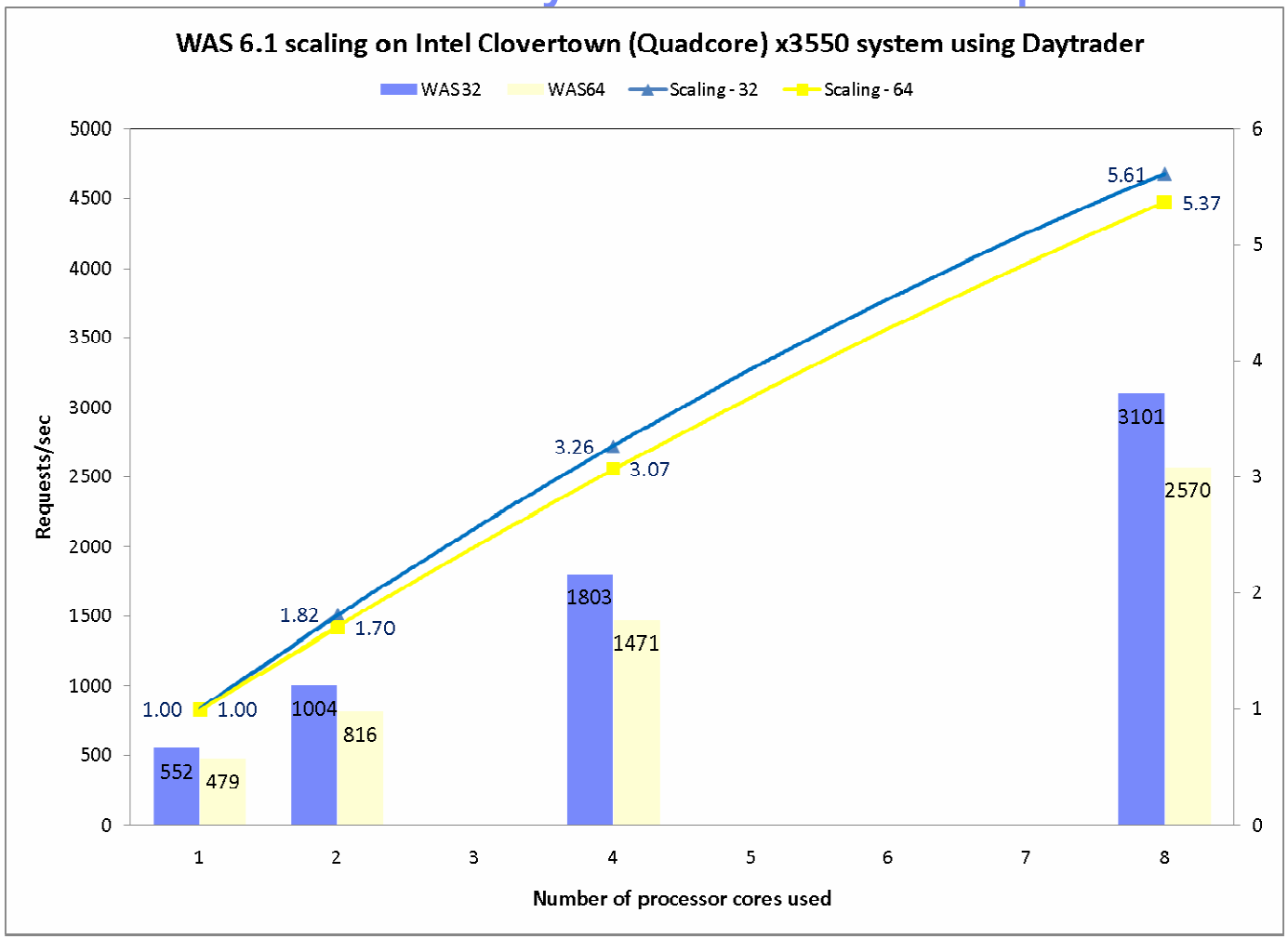
You can re-engineer your applications for 64-bit

- Many existing applications may not improve by moving to 64-bit, but you might be able to re-engineer the applications to use the additional memory and significantly improve performance.
  - Example: Use co-located memory caches (Dynacache or ObjectGrid) for database type of workloads
- Applications can be moved from WAS 32-bit to WAS 64-bit without much effort.
  - Native code (JNI) or any 3rd party libraries (which uses native code/JNI) would require porting/recompilation for 64-bit OS.

Roadmap

- ✓ WAS V6 and the 64-bit platforms supported
- ✓ Key Advantages of WAS on 64-bit platforms
- ✓ Downside of the 64-bit platforms
- ✓ **Scalability and other useful information about 64-bit platforms**
  - Scalability of WAS 64-bit on multi-core platforms
  - Running WAS 32-bit on 64-bit platforms
  - Impact of large heaps on GC Pause times
- How to decide if 64-bit is for you?
- Closing Notes
- Resources

WAS 64-bit Scalability on multi-core platforms



- Chart above shows scaling on an Intel 2P/8C (quad-core) system running WAS V6.1 32-bit and 64-bit.
- The overall scaling of 64-bit is slightly lower than that of 32-bit on this hardware. Potential reasons include higher impact on caches and bus utilization in the 64-bit mode.

WAS 32-bit on 64-bit platform support

HW Platform	64-bit OS platforms	WAS 6.02		WAS 6.1.	
		32-bit	64-bit	32-bit	64-bit
POWER	AIX 5L 5.2				
POWER	AIX 5L 5.3				
POWER	Red Hat AS 4.0				
POWER	SLES 9				
POWER	SLES 10				
SPARC	Sun Solaris 8				
SPARC	Sun Solaris 9				
SPARC	Sun Solaris 10				
x86-64	Sun Solaris 10				
x86-64	Red Hat AS 4.0			V6.1.0.11	
x86-64	SLES 9			V6.1.0.11	
x86-64	SLES 10			V6.1.0.11	
x86-64	Windows Server 2003			V6.1.0.11	
zSeries	zLinux RH AS 4.0**				
zSeries	zLinux SLES 9**				
zSeries	zLinux SLES 10**				
zSeries	z/OS 1.6**				
PA-RISC	HP-UX 11iv2				
Itanium 2	HP-UX 11iv2				

Supported

Not Supported

** WAS is 31-bit or 64-bit on zLinux and z/OS

- Many customers want to standardize on 64-bit OS in their infrastructure. They should run WAS 32-bit if their apps don't leverage WAS 64-bit features.
- Support for WAS 32-bit on 64-bit platforms is shown in the chart.
  - Running WAS 32bit and WAS 64-bit on the same 64-bit platform is supported
  - These instances can be part of same *cell* as well (*mixed mode* support) with either a 32/64 bit DMgr
- Very few 64-bit platforms don't support running WAS 32-bit

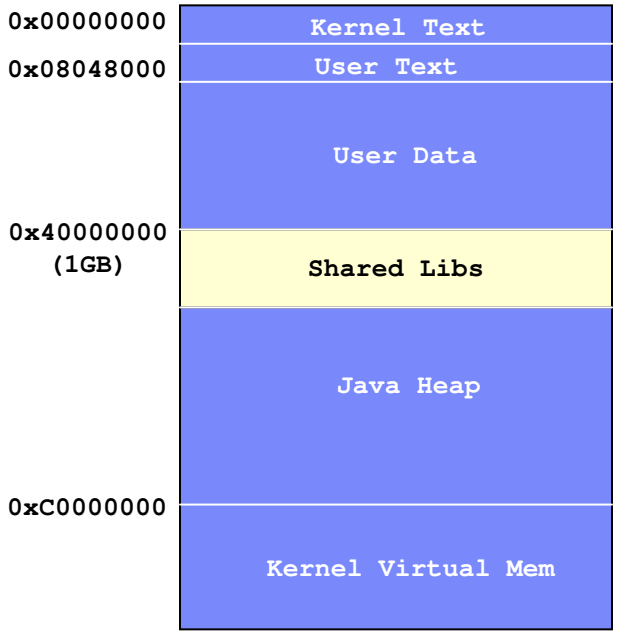
Running WAS 32-bit on 64-bit platforms

- WAS is available in both 32 and 64-bit editions for many supported 64-bit platforms
  - 32 and 64-bit WAS installs can co-exist on these 64-bit platforms
    - Applications can be deployed on both 32 and 64 bit to compare performance
    - Provides a simplified migration path for 32-bit applications
    - 32-bit applications run natively (full performance is contingent on OS implementation)
  - Customers can choose the preferred edition for optimal performance
  - WAS 32-bit on 64-bit operating systems can provide incrementally greater heap size
    - ~3Gb heaps are possible on Linux

On Linux the Address where shared libs are loaded can be moved leaving more room for the Java heap

echo "0x10000000" > /proc/self/mapped_base

The above command moves it to 256M



Linux Process Virtual Address Space

Impact of large heaps on GC Pause times

- Java does not set an artificial limit on maximum heap size.
  - But what is the effect on GC with very large heaps?
- IBM Concurrent GC technology (optavgpause, gencon) – options in large heap environments.
  - Gencon (Generational GC) can outperform flat-heap collection while providing smaller pause times.
  - But this requires tuning effort, out-of-box defaults will not give best results in most cases.

Example application in Lab, tuned to run with either 1024MB or 6144MB heap. What is the effect of GC frequency and pause time?

1024MB
Average GC Pause Time: **239 ms**
Average GC Mark Time: **224 ms**
Frequency: **2000ms**

6144MB
Average GC Pause Time: **418 ms**
Average GC Mark Time: **368 ms**
Frequency: **19000ms**

6144MB with IBM Concurrent GC (optavgpause)
Average GC Pause Time: **30 ms**
Frequency: **22000ms**

Roadmap

- ✓ WAS V6 and the 64-bit platforms supported
- ✓ Key Advantages of WAS on 64-bit platforms
- ✓ Downside of the 64-bit platforms
- ✓ Scalability and other useful information about 64-bit platforms
- ✓ **How to decide if 64-bit is for you?**
  - The Checklist
- Closing Notes
- Resources

Is 64-bit for me ? – The Checklist

- ✓ Does your application need a Java heap much greater than ~2GB for higher performance? If Yes → WAS 64-bit
- ✓ 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
- ✓ Do you need to support 64-bit OS although your application doesn't leverage 64-bit features? If Yes → WAS 32-bit on 64-bit platforms
- ➢ If you answered "No" to all above questions, then WAS 32-bit on 32-bit platforms is better suited for you

Roadmap

- ✓ WAS V6 and the 64-bit platforms supported
- ✓ Key Advantages of WAS on 64-bit platforms
- ✓ Downside of the 64-bit platforms
- ✓ Scalability, Cross platform comparisons and other useful information about 64-bit platforms
- ✓ How to decide if 64-bit is for you?
- ✓ **Closing Notes**
  - Resources

Closing Notes

- Version 6.02 onwards WAS extends support to various 64-bit platforms
  - WAS 64-bit versions leverage 64-bit platform features providing the capability to have Java heaps much larger than the ~2Gb limitations of 32-bit platforms
  - The IBM JVM automatically generates optimized code leveraging 64-bit CPU performance features
  - Applications that leverage these features can see significant performance gains when deployed on a WAS 64-bit install
  - Applications can see a significantly increased memory footprint requirement on 64-bit versus 32-bit. This can result in reduced hardware cache efficiency and performance

Roadmap

- ✓ WAS V6 and the 64-bit platforms supported
- ✓ Key Advantages of WAS on 64-bit platforms
- ✓ Downside of the 64-bit platforms
- ✓ Scalability, Cross platform comparisons and other useful information about 64-bit platforms
- ✓ How to decide if 64-bit is for you?
- ✓ Summary
- ✓ **Resources**

Resources

- WebSphere Application Server Performance website
  - http://www.ibm.com/software/webservers/appserv/performance.html
  - Provides performance articles, best practices, Trade Benchmark etc.
- WebSphere Application Server Infocenter
  - http://www.ibm.com/software/webservers/appserv/was/library/
  - Provides install/admin guides, software/hardware pre-requisites, OS related tuning info
- Redbooks
  - http://www.redbooks.ibm.com
  - Best Practices for High-Volume Web Sites
  - DB2 UDB/WebSphere Performance Tuning Guide
  - IBM WebSphere V6.1 Performance, Scalability, and High Availability WebSphere Handbook
  - WebSphere Version 6 Application Development Handbook

Legal Notices