2009 System z Expo

October 5 – 9, 2009 – Orlando, FL

Session Title:
End to End Performance of WebSphere Environments on Linux for IBM System z

Session ID: zLP03

Speaker Name: Dr. Juergen Doelle

Authorized IBM Training

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®
DB2*, DB2 Connect, DB2 Universal Database, Informix®, Tivoli®, ECKD, Enterprise Storage Server®, FICON Express, HiperSocket, OSA, OSA Express

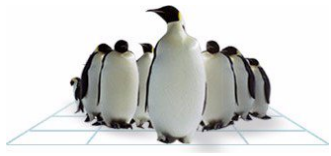
The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

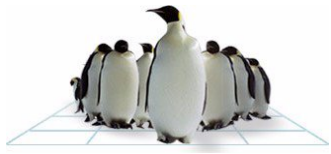
Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Agenda



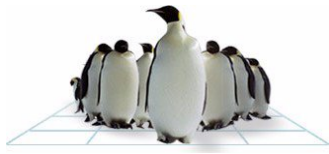
- **Objectives**
- **Performance Areas**
  - Network
  - Java
  - IBM system z10 vs z9
  - 64 bit
- **WebSphere Application Server Cluster**
- **Virtualization - z/VM vs Xen**

Objectives



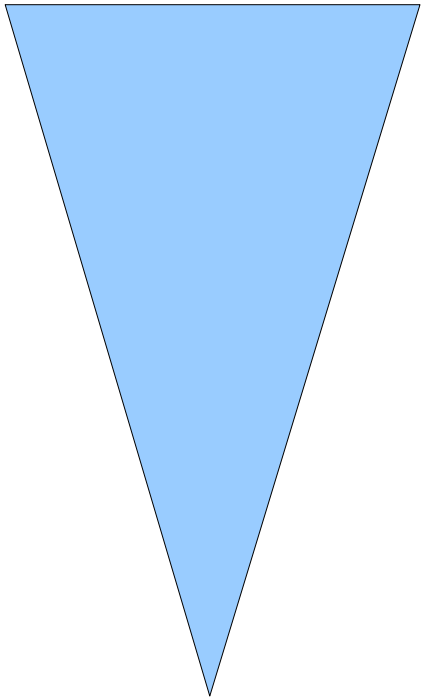
- **Demonstrate how WebSphere Application Server performance can benefit from the advantages provided by IBM System z**
  - What are important system settings in regard to performance
  - Which performance relevant areas have been identified
  - What needs to be done to get the best performance
  - How WebSphere Application Server environments on Linux on System z scale
- **We did no high end benchmarking!**
  - Customer-like environments are used.

Performance tuning at all layers

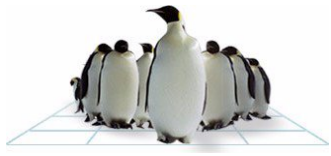


- **"Optimize your stack from the top to the bottom"**
  - Application design
  - Application setup
  - **Application server**
  - Database
  - **Operating system**
  - **Virtualization system**
  - **Hardware**

Covered in this presentation

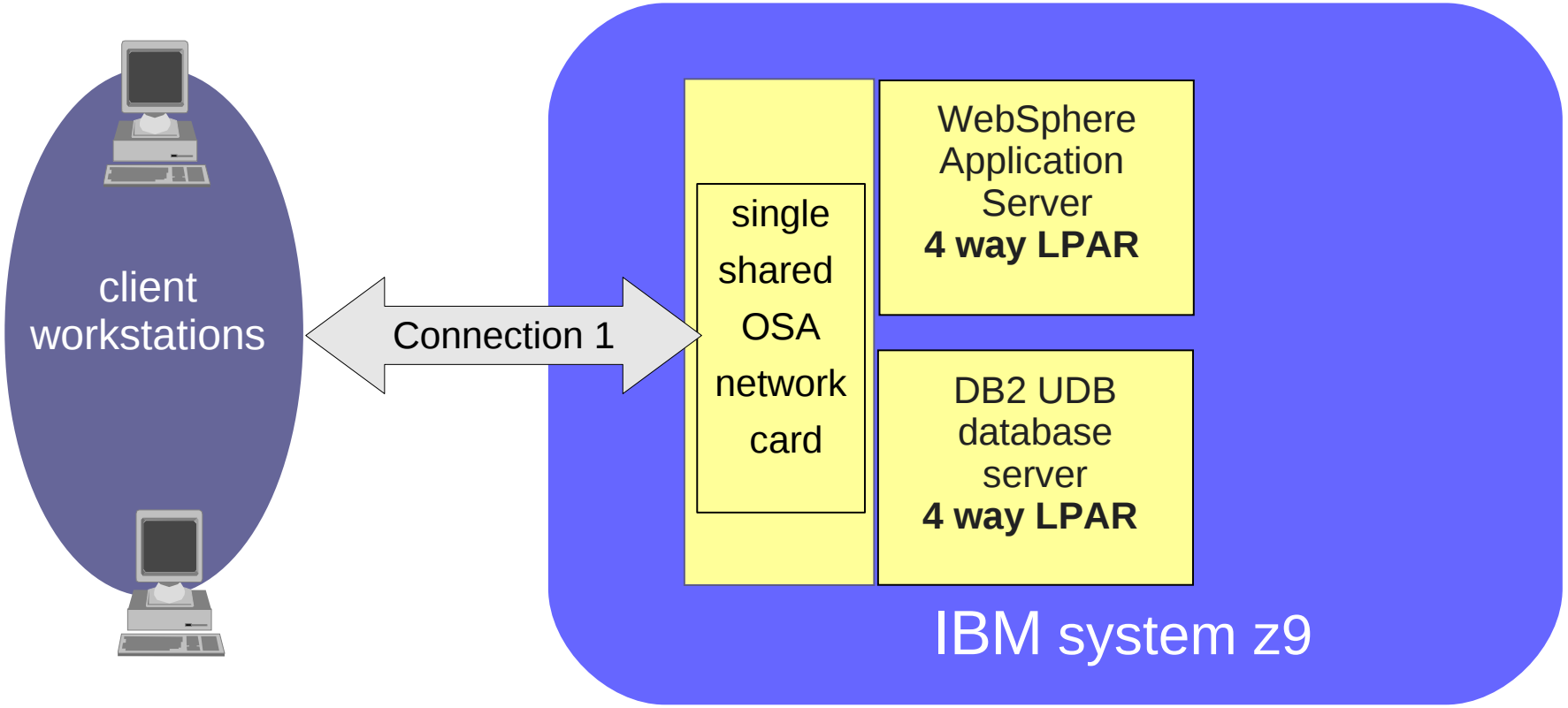


Agenda



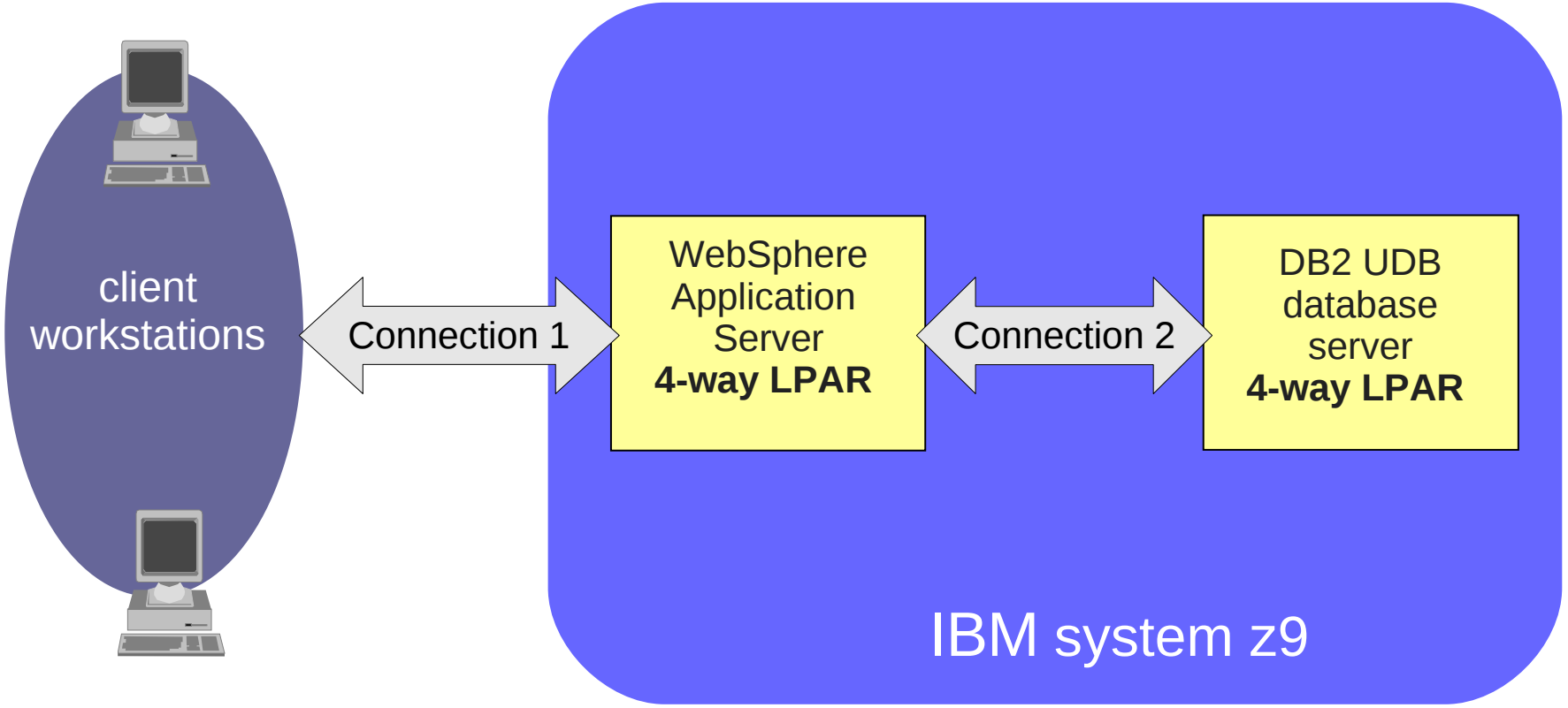
- Objectives
- **Performance Areas**
  - **Network**
  - Java
  - IBM system z10 vs z9
  - CPU Scaling
  - 64 bit
- WebSphere Application Server Cluster
- Virtualization - z/VM vs Xen

WebSphere Base Environment (LPAR)



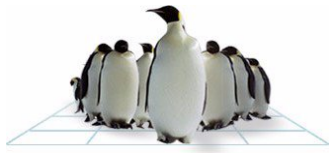
- **let's start with a simple setup**
- **when increasing the load, the first bottle neck was the single shared network connection**

Network constraints – Base Environment



- **first tuning step:** **separate the connection to the database (2nd OSA card)**
  ==> improvement +10%
- **second step:** **use Hipersockets for connection 2**
  ==> improvement +33%

Network constraints – setup changes



- **Choose your MTU size carefully!**
  - Avoid fragmentation, lots of small packages can drive up CPU utilization
  - Use the largest MTU size supported in the path, and verify it
  - How-To:

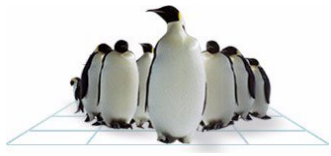
```
ping -M do system15.ibm.com  -s 8000 -c3
PING system15.ibm.com 8000(8028) bytes of data.
From dyn-9-152-198-41.ibm.com icmp_seq=0 Frag needed and DF set (mtu = 1500)
```

- **For really busy network devices consider to**
  - Increase the number of inbound buffers in the qeth driver (default 16)
    - How-To:
      Device has to be offline

```
echo <number> > /sys/bus/ccwgroup/drivers/qeth/<device_bus_id>/buffer_count
```

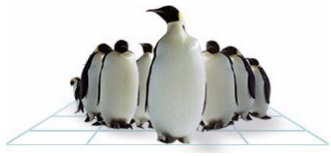
    - Consumes memory!
      - 64KB per buffer, maximum 128 buffer = 8 MB per device
      - for tuning purpose, start with a large value, monitor the impact and then iteratively reduce the number of buffers until throughput drops down
  - Use channel bonding
  - Use OSA express 3 cards

Networking – Connection types



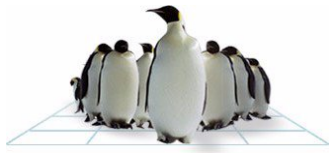
- **z/VM guest to guest communication**
  - VSWITCH without an OSA card
  - Guest LAN (no layer 2 support)
- **LPAR to LPAR communication on the same System z box**
  - use Hipersockets
    Hipersockets are completely driven by CPU
- **External connectivity:**
  - Use 10 GbE cards with MTU 8992
  - Use the new OSA express 3 card
  - VSWITCH with an OSA card
  - Attach OSA directly to the Linux guest

Agenda



- Objectives
- **Performance Areas**
  - Network
  - **Java**
  - IBM system z10 vs z9
  - CPU Scaling
  - 64 bit
- WebSphere Application Server Cluster
- Virtualization - z/VM vs Xen

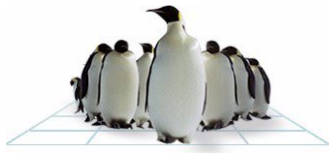
Java on servers: Heap size



- **Heap size needs to be sized adequately**
  - maximum heap size <= available memory
    - avoids paging in Linux and z/VM
  - Heap too small: frequent garbage collection and OutOfMemoryErrors
  - Heap too big: "waste" of memory
  - 31-bit Java kits: larger heap sizes up to 1.6 GB (modify memory layout)
    - also true for 31-bit Java kits in a 64-bit Linux environment

- **Useful Java interpreter parameters for fine tuning – workload dependent**
  - setting a fixed heap size: **-Xms** (initial), **-Xmx** (maximum), when initial==maximum
  - monitor garbage collection (GC): **-verbose:gc**
  - control GC behavior: **-Xgcpolicy:[optthruput, optavgpause, gencon]**
  - 64-bit: smaller size of heap objects: **-Xcompressedrefs**

Java on servers: larger heaps for 31-bit Java kits (1)

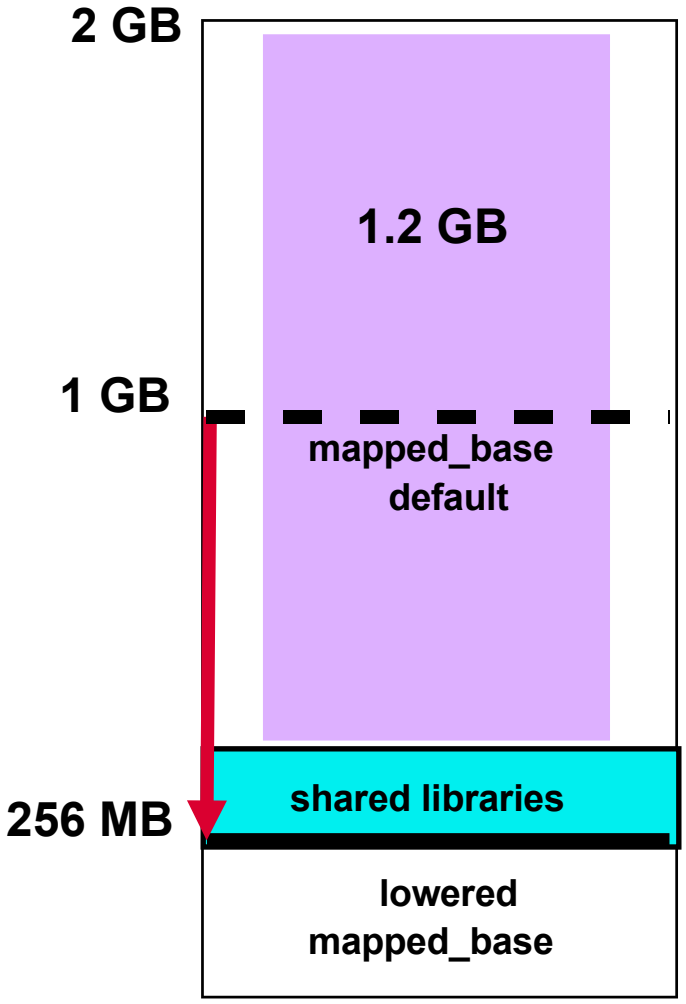


- **Modify Linux memory layout**
  - reorder mapped base for shared libraries
  - relevant for 31-bit emulation mode on Novell SLES 9,10

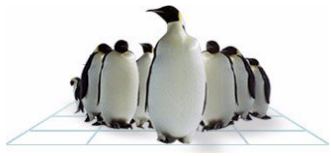
How-To:

- **PID is the process ID of the process you want to change the layout of**
  - usually the bash shell which starts the Application Server
  - $$ gives the current shell **PID**, /proc/self/... works as well
- **Display memory map of any PID by**
  - cat /proc/<PID>/maps
- **Check the mapped base value by**
  - cat /proc/<PID>/mapped_base
- **Lower the value to e.g. 256 MB by**
  - echo 268435456 >/proc/<PID>/mapped_base

==> retry to allocate a larger heap size



Java on servers: larger heaps for 31-bit Java kits (2)



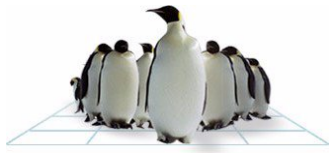
- **Modify Linux memory layout**
  - RHEL includes flex-mmap patch; turn off Linux prelinking
  - Applies RHEL 4,5 distributions (31-bit emulation mode)

How-To:

- **Show state of flex-mmap patch**
  - cat /proc/sys/vm/legacy_va_layout
  - 0 means flex-mmap is enabled; 1 means old memory layout
- **Enable flex-mmap if disabled**
  - echo 0 > /proc/sys/vm/legacy_va_layout
- **Disable Linux prelinking**
  - in /etc/sysconfig/prelink set PRELINKING=no
- **Apply setting by running the daily cron prelink job immediately**
  - **# /etc/cron.daily/prelink <ENTER>**

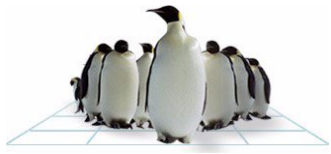
==> retry to allocate a larger heap size

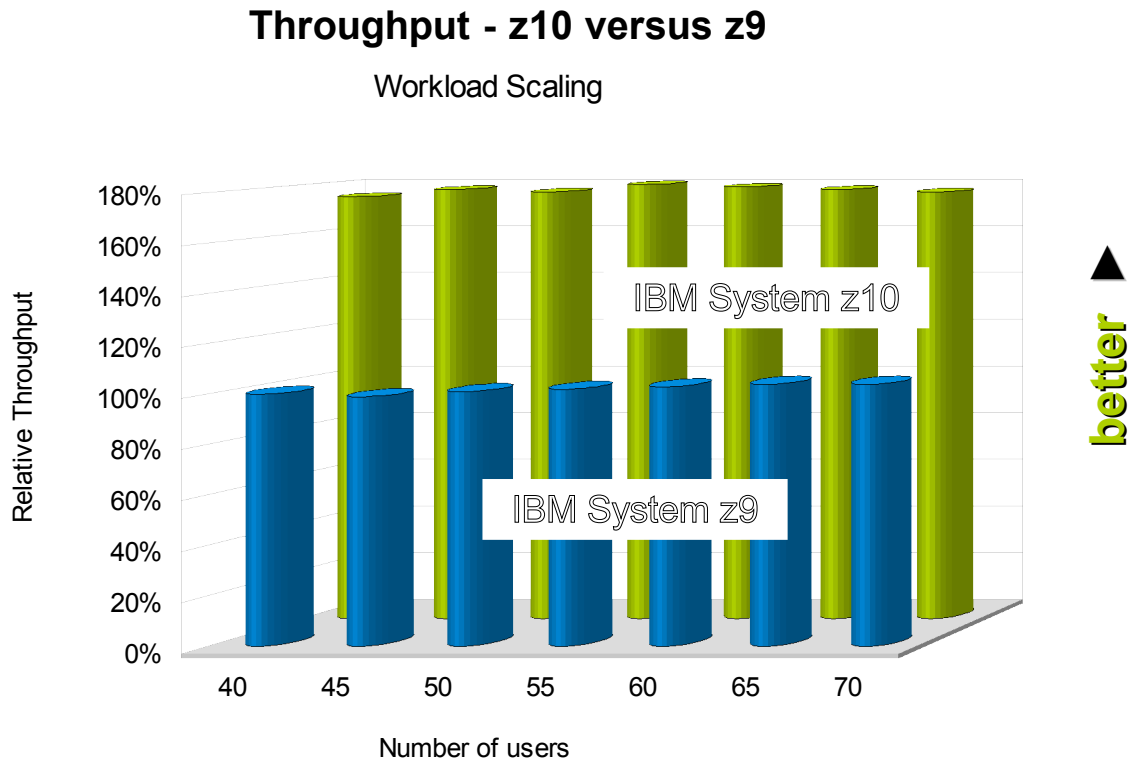
Agenda



- Objectives
- **Performance Areas**
  - Network
  - Java
  - **IBM system z10 vs z9**
  - CPU Scaling
  - 64 bit
- WebSphere Application Server Cluster
- Virtualization - z/VM vs Xen

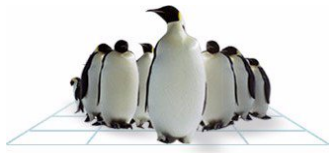
WebSphere Application Server Cluster - Comparison IBM System z9 versus z10 (1)





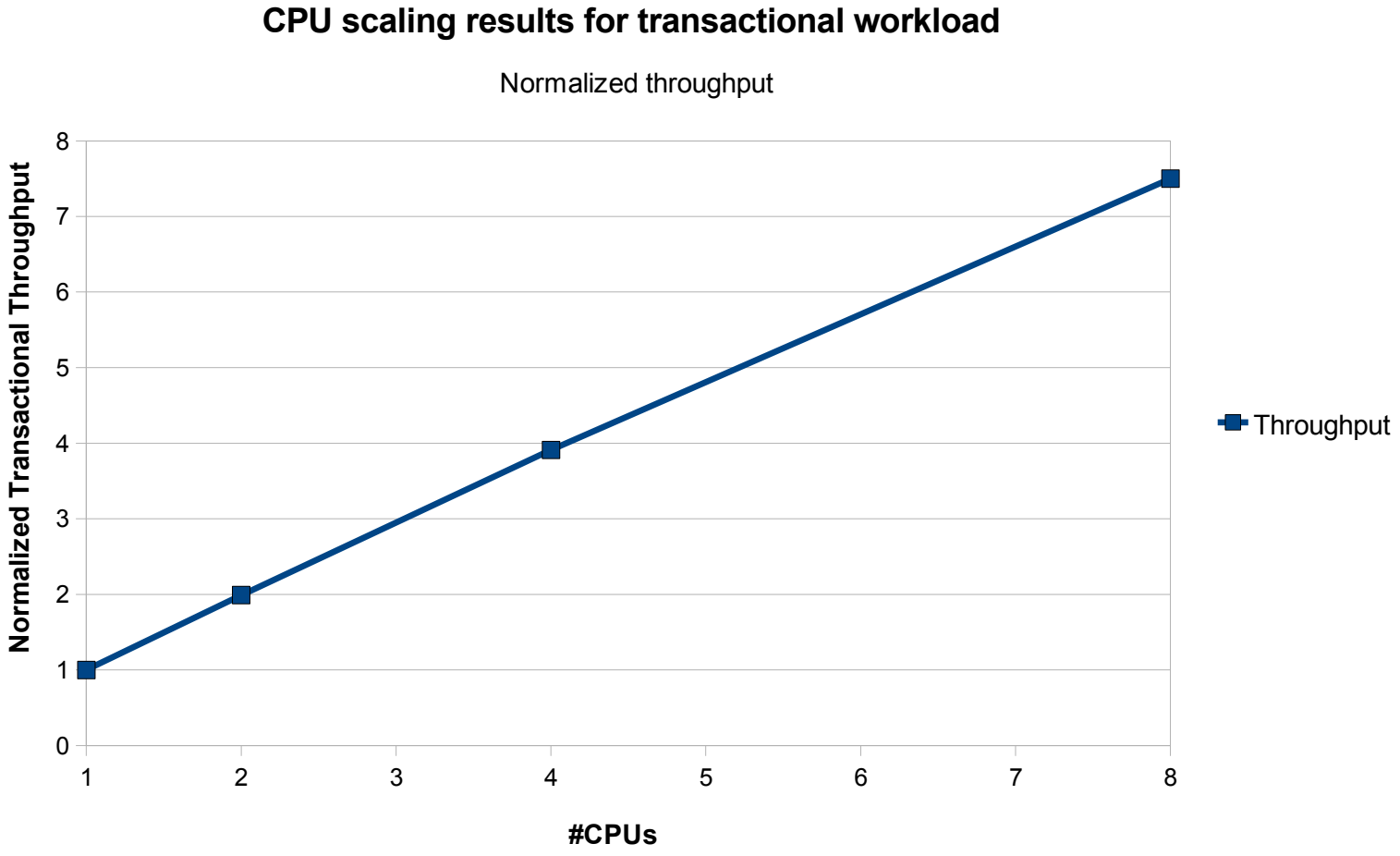
- **IBM System z10 provides constantly about 80% higher throughput!**
  - In case of an computing intensive workload 80% - 100% performance improvement have been shown on IBM System z10
  - In case of an disk I/O intensive workload improve the disk I/O bandwidth by using e.g. an IBM DS8000 and ensure an optimized setup
    - More information at: ibm.com/developerworks/linux/linux390/perf/tuning_rec_dasd.html

Agenda



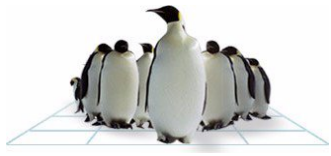
- Objectives
- **Performance Areas**
  - Network
  - Java
  - IBM system z10 vs z9
  - **CPU Scaling**
  - 64 bit
- WebSphere Application Server Cluster
- Virtualization - z/VM vs Xen

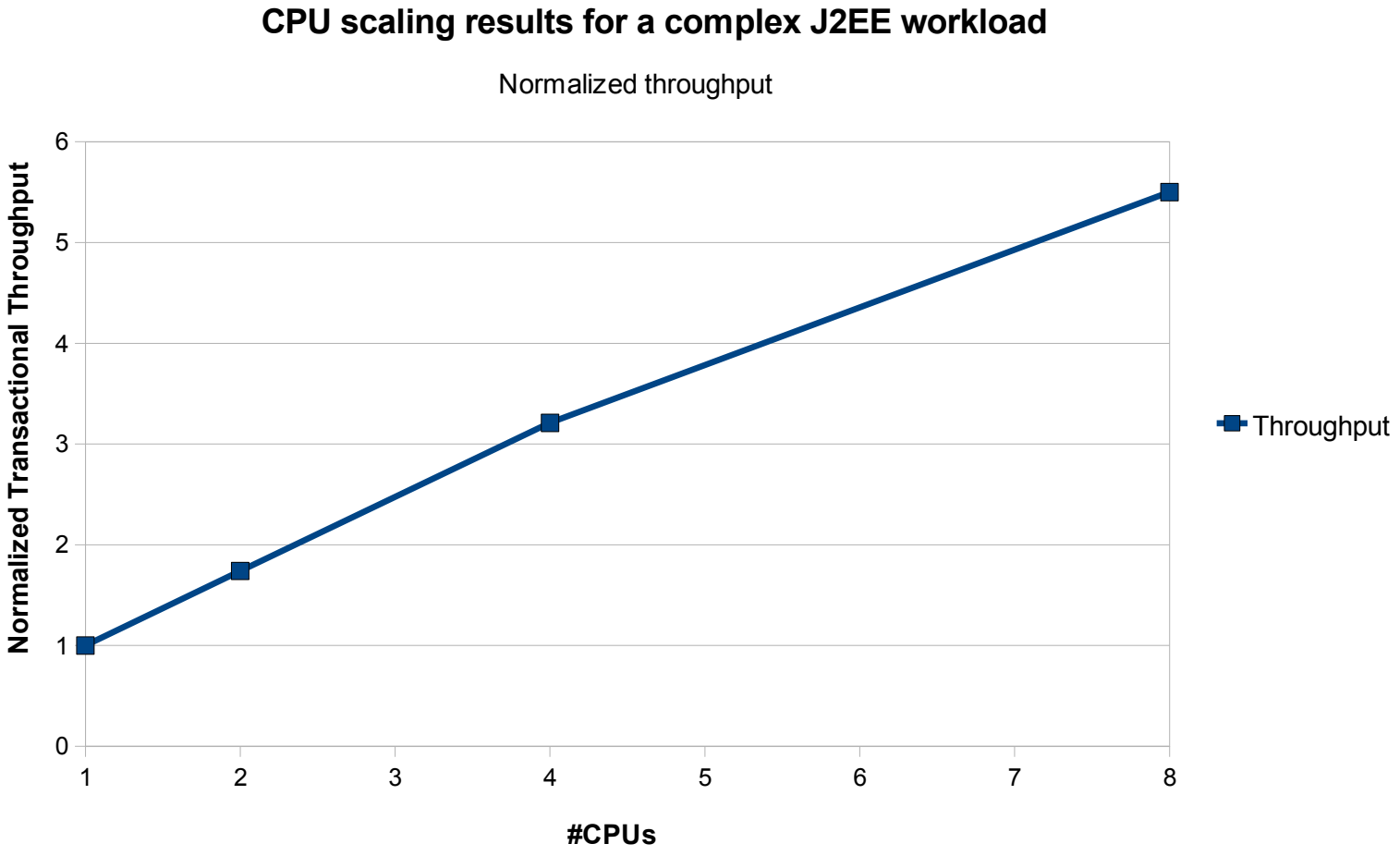
Websphere 6.1 CPU scaling – transactional workload



- **Linear CPU scaling!**
  - makes planning the resources needed for scaling this workload easy
- **Hardware: IBM System z9**

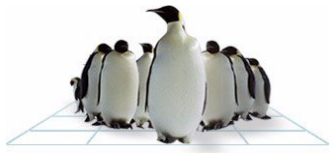
Websphere 6.1 CPU scaling – J2EE workload





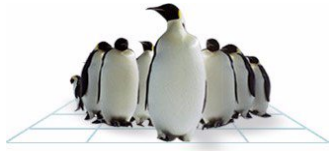
- **Very linear CPU scaling**
- **Hardware: IBM System z10**
  - The higher CPU power results in much higher requirements on the capacity of the environment!

Agenda



- Objectives
- **Performance Areas**
  - Network
  - Java
  - IBM system z10 vs z9
  - CPU Scaling
  - **64 bit**
- WebSphere Application Server Cluster
- Virtualization - z/VM vs Xen

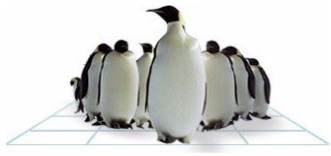
31-bit versus 64-bit

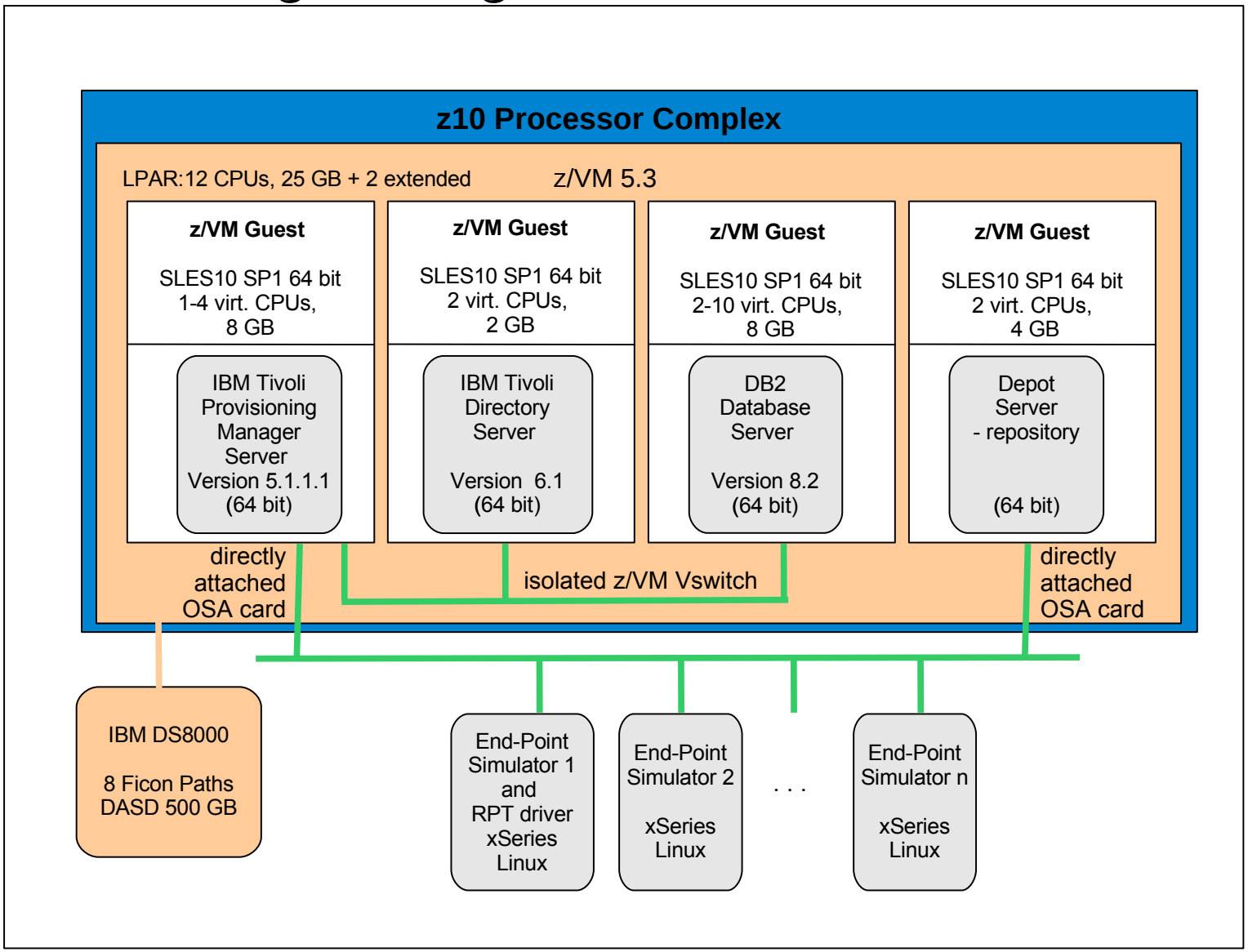


- **64-bit WebSphere Application Server**
  - You can run 31-bit WebSphere in the 31-bit emulation layer of 64-bit distributions (RHEL5, SLES10) – there is no dependency on the distributions!
  - Pro: Provides the possibility of very large Java heaps
  - Contra: needs additional CPU cycles and memory resources because of larger addresses

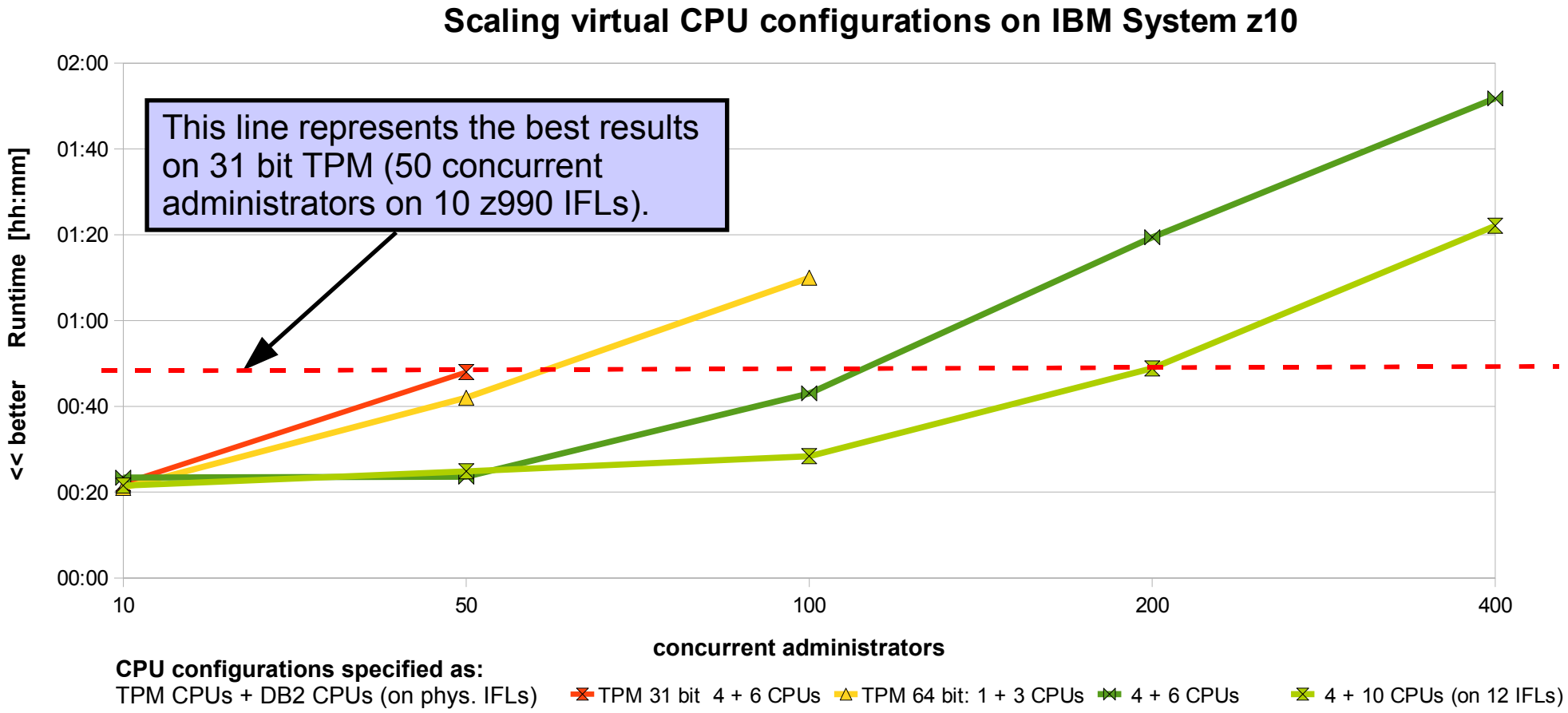
- **If the application does not need the additional memory size and heap then the use of 31-bit is recommended**
  - if the application does not use long living large data objects, the garbage collection does an excellent job to reduce the memory requirements
  - There may be constraints like supported configuration, local 64-bit database connection

Sample for WebSphere 64-bit application - Tivoli Provisioning Manager



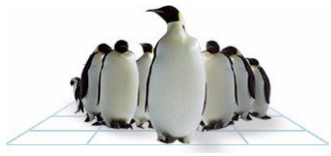


Tivoli Provisioning Manager (TPM) 5.1.1.1 - 64 bit



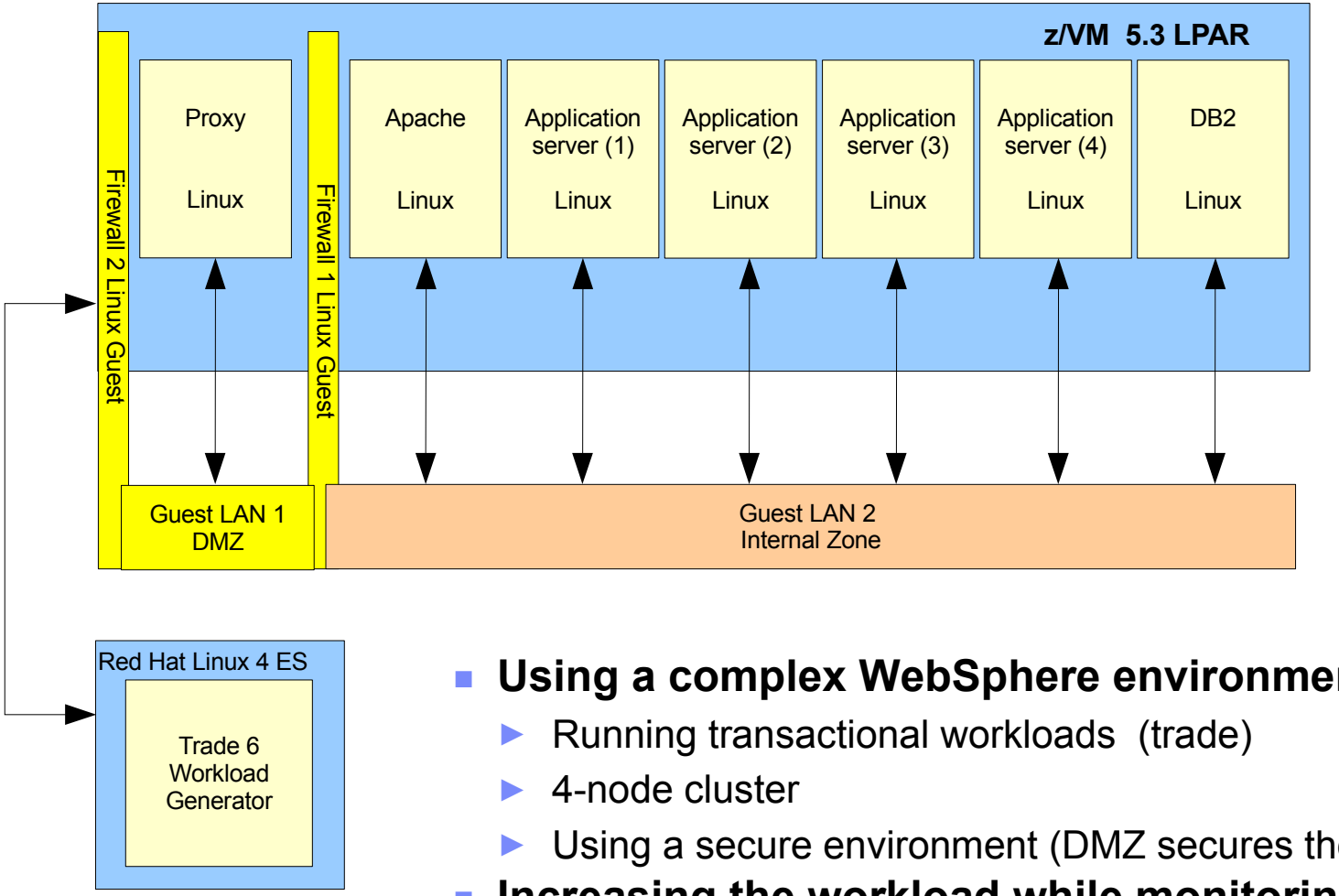
- **The 31 bit Tivoli Provisioning Manager was limited to 50 concurrent administrators!**
  - All configurations below the red dashed line exceed the performance of 31 bit (many at reduced cost or higher scale)!
  - This administrator limitation was blown away with 64 bit TPM
    Now we are able to drive 200 administrators at the same runtime on System z10

Agenda



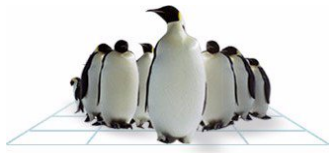
- **Objectives**
- **Performance Areas**
  - Network
  - Java
  - IBM system z10 vs z9
  - CPU Scaling
  - 64 bit
- **WebSphere Application Server Cluster**
- **Virtualization - z/VM vs Xen**

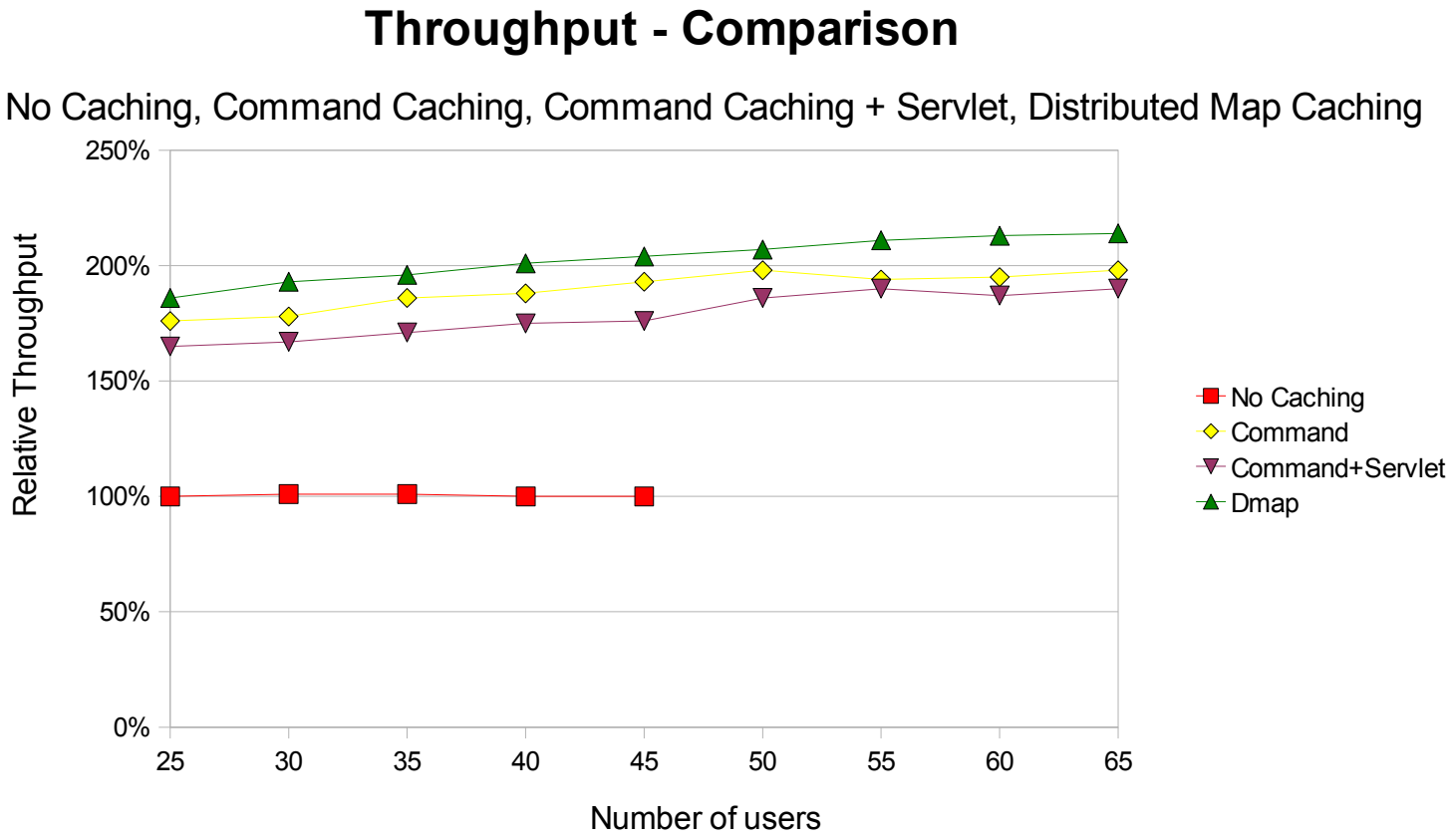
WebSphere Application Server Cluster - Environment



- **Using a complex WebSphere environment**
  - Running transactional workloads (trade)
  - 4-node cluster
  - Using a secure environment (DMZ secures the internal zone)
- **Increasing the workload while monitoring throughput**

WebSphere Application Server Cluster - Varying caching modes

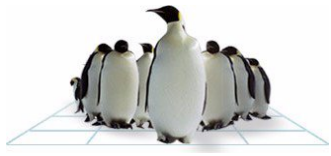


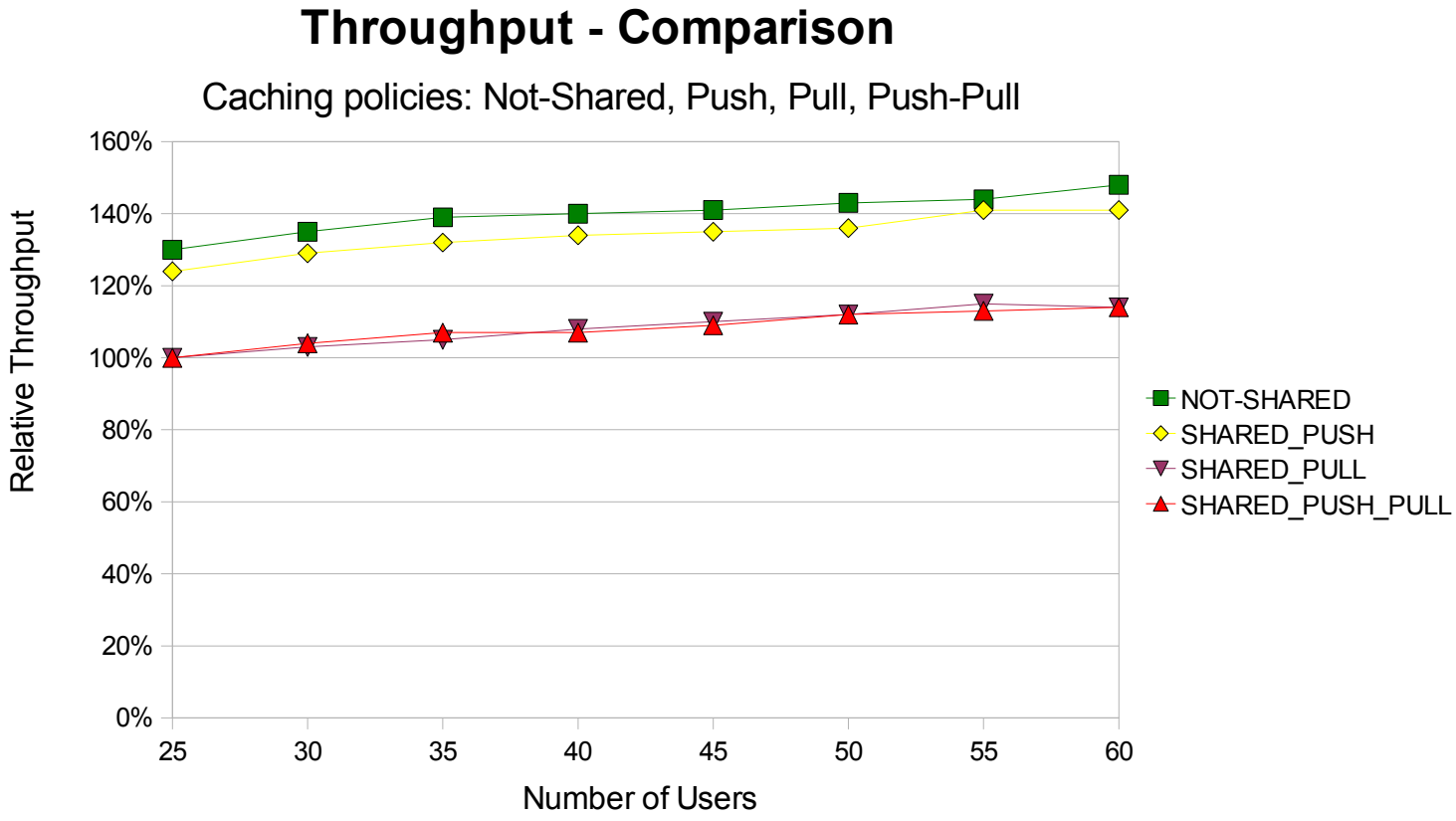


- **Distributed map mode provides the best throughput**
- **Caching and the cache mode needs support from the application!**

➢ **Enable caching is recommended!**

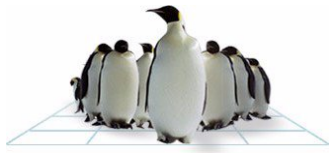
WebSphere Application Server Cluster - Vary caching policies

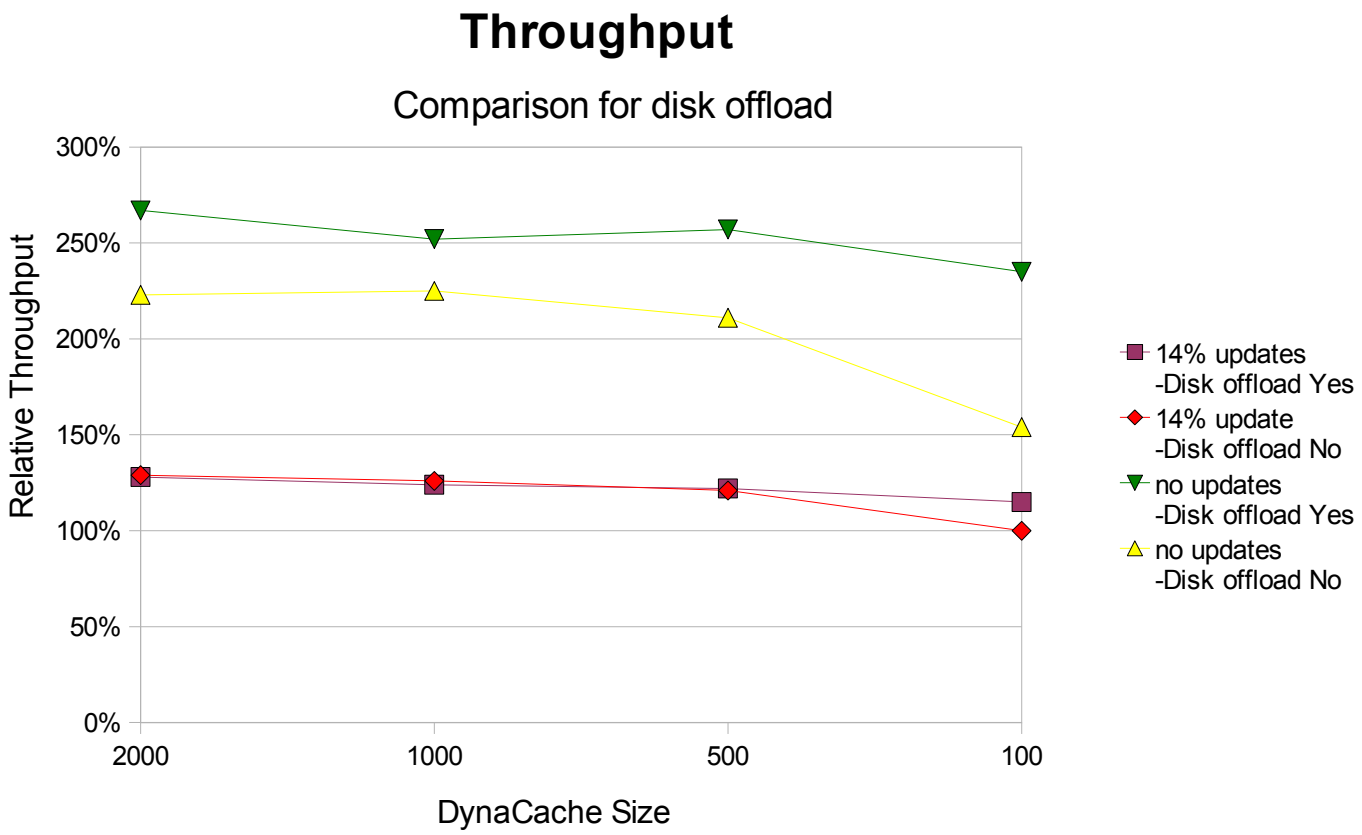




- **Policies**
  - push: Cache entries for an object are automatically distributed between application servers
  - pull: Cache entries for an object are requested from other application servers on demand
- **Keeping shared caches in the cluster consistent is related with overhead**
- **Impact is highly workload dependent!**

WebSphere Application Server Cluster - Cache disk offload

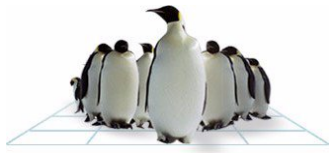




- **Allows smaller cache sizes**
- **Is not related with additional CPU cost!**

➢ **Enabling cache offload to disk proved to be very effective**

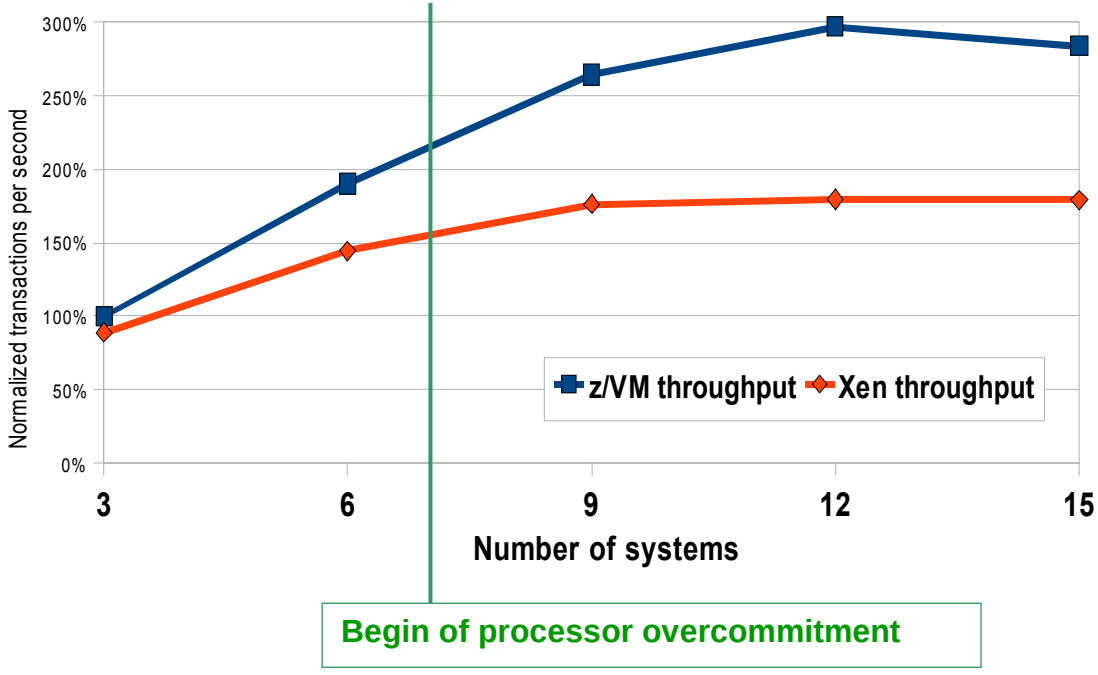
Agenda



- **Objectives**
- **Performance Areas**
  - Network
  - Java
  - IBM system z10 vs z9
  - CPU Scaling
  - 64 bit
- **WebSphere Application Server Cluster**
- **Virtualization - z/VM vs Xen**

z/VM and Xen Virtualization - Processor Overcommitment

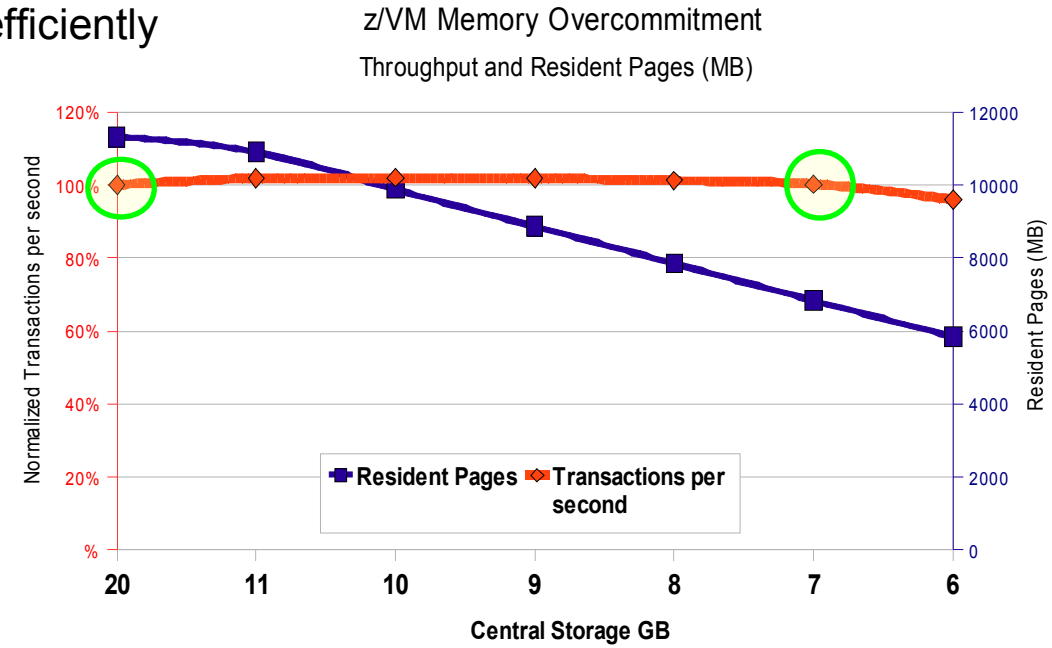
- **Throughput rate with z/VM is significantly higher - measured on IBM System z9**
  - z/VM scales very well, even when the processors are overcommitted
    - Xen flattens when reaching the processor overcommitment
  - z/VM scales until the system is fully utilized
  - Xen scales until processor overcommitment is reached



- **z/VM handles processor overcommitment very efficiently**
  - This will show even better results when running on a IBM System z10

z/VM and Xen Virtualization - Memory Overcommitment

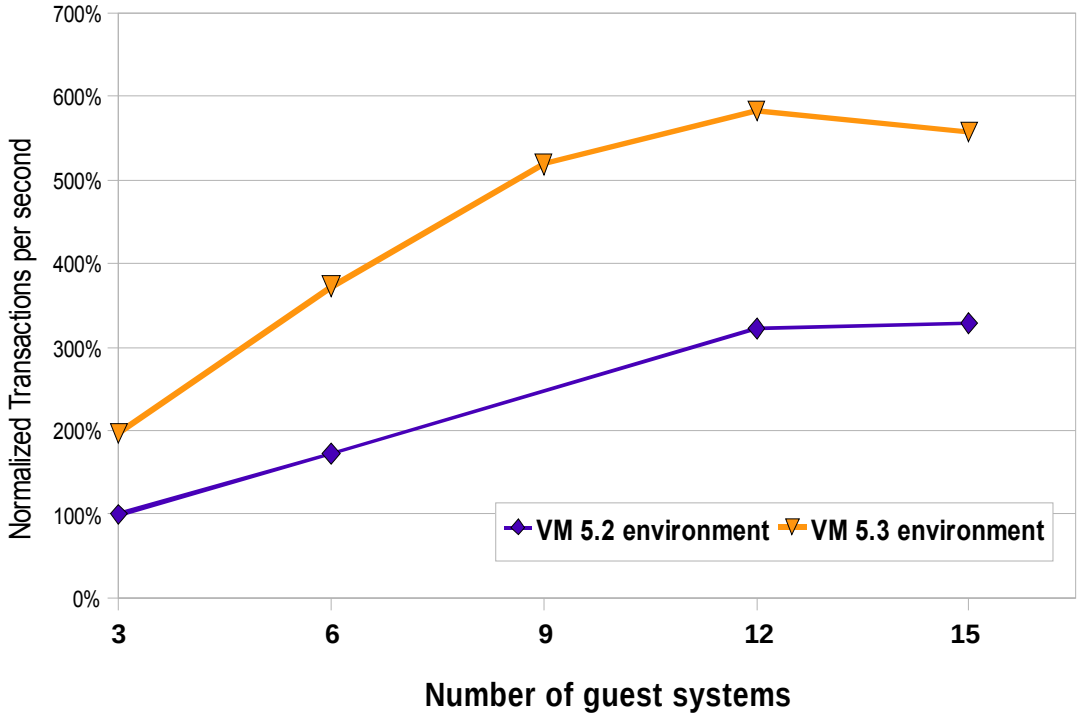
- **z/VM's memory overcommitment is outstanding**
  - z/VM handles memory resources very efficiently
    - Storage allocation is optimized to allocate what is needed only
  - Throughput did not degrade
    - Same storage throughput with 20 GB and 7 GB



- **z/VM automatic memory management provides**
  - Optimized memory utilization
  - Very flexible guest management
  - High flexibility for a Dynamic Infrastructure®

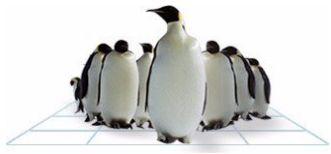
z/VM V5.2 versus z/VM V5.3
Virtualization throughput comparison

- **Current software levels provides a significant improvement in throughput**
- **Sample:**
  - Software versions used for these measurements:
    - z/VM 5.2 → 5.3
    - Java 1.4 → 1.5
    - WebSphere Application server 6.0.2 → 6.1.0.11
    - DB2 8.2 → 9.1



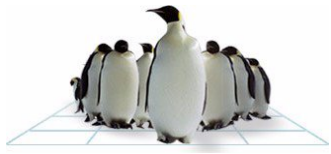
- **Benefits from impressing performance improvement**
  - Keep your software up to date

Summary



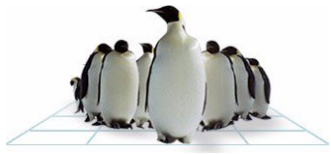
- **Important performance areas for WebSphere Application Server environments are**
  - Network
    - the bandwidth from user to the application server
    - the bandwidth of the interconnect to the database
    - suitable connectivity type
  - the appropriate Java heap size
  - IBM System z10 showed an improvement of 80% in throughput with WebSphere workloads
  - Our WebSphere workloads scaled very linearly with the amount of CPUs, which makes it easy to plan the resource usage for growing workloads
  - If a workload needs large data structures as Tivoli Provisioning Manager, the 64 bit WebSphere Application Server provides very large heaps with the impressive performance advantage
  - Usage of WebSphere DynaCache can be highly recommended!
    - The sharing policies NOT-SHARED and SHARED-PUSH provided the best performance for caching in the WebSphere cluster
    - Configuring Cache disk-offload works very effectively without additional overhead
    - Caching requires application support
  - z/VM is a very good virtualization platform for WebSphere environments
    It provides a high level on resource overcommitments for CPU and memory
  - Keeping the software levels in your WebSphere environment up to date can provide impressive performance improvements.

White papers:



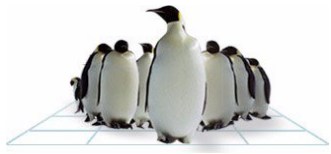
- **WebSphere Application Server Base Performance**
  - http://www.ibm.com/developerworks/linux/linux390/perf/tuning_pap_websphere.html#wasbp
- **WebSphere Application Server 6.1 Base Performance**
  - http://www.ibm.com/developerworks/linux/linux390/perf/tuning_pap_websphere.html#wasbp61
- **End-to-End Performance of a WebSphere Environment Including Edge Components**
  - http://www.ibm.com/developerworks/linux/linux390/perf/tuning_pap_websphere.html#weec
- **Tuning WebSphere Application Server Cluster with Caching**
  - http://www.ibm.com/developerworks/linux/linux390/perf/tuning_pap_websphere.html#wascc
- **z/VM virtualization performance**
  - http://www.ibm.com/developerworks/linux/linux390/perf/tuning_pap_VM.html#cppu
- **z/VM and Xen virtualization performance**
  - http://www.ibm.com/developerworks/linux/linux390/perf/tuning_pap_VM.html#xen

Related Topics at 2009 System z Expo



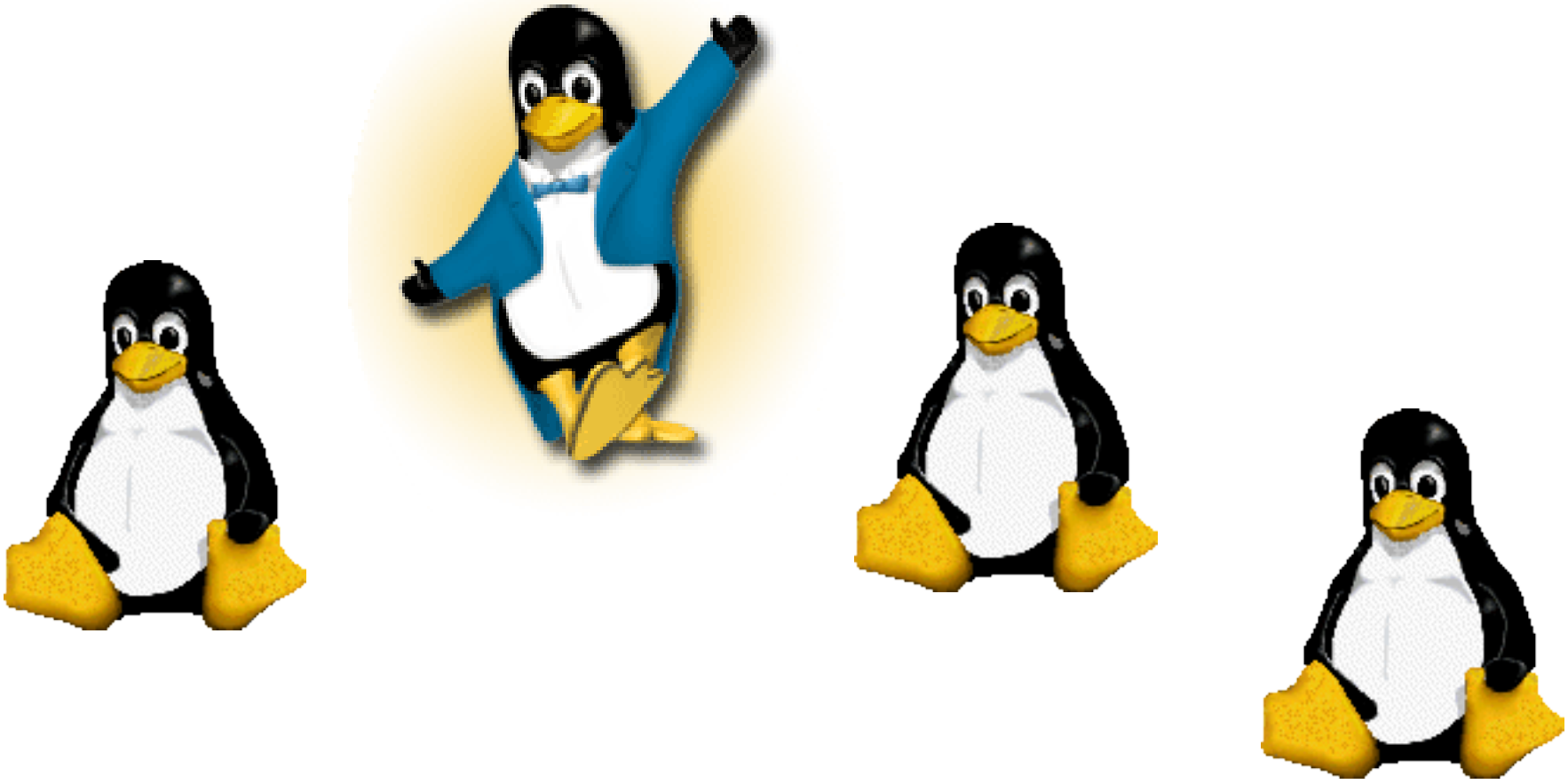
- **Getting Started with WebSphere and Virtualization for System z Linux**
  zLA01 Tuesday 10:35 AM
- **Sizing Memory for WebSphere Applications on System z**
  zLA02 Tuesday 2:35 PM
- **Performance Tuning and Monitoring: DB2 for Linux, Unix and Windows (LUW) for Linux**
  zLA08 Wednesday 4:10 PM
- **Performance Experience with Databases on Linux for IBM System z**
  zLP02 Wednesday 1:00 PM

Visit us !

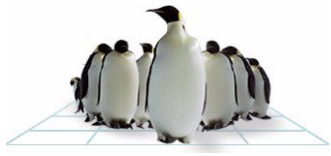


- **Linux on System z: Tuning Hints & Tips**
  - http://www.ibm.com/developerworks/linux/linux390/perf/
- **Linux-VM Performance Website:**
  - http://www.vm.ibm.com/perf/tips/linuxper.html
- **IBM Redbooks**
  - http://www.redbooks.ibm.com/
- **IBM Techdocs**
  - http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs

Questions



WebSphere Application Server Cluster - Cache sharing policies



- **Not-shared:**
  - Cache entries for this object are not shared among different application servers. These entries can contain non-serializable data.
- **Shared-push:**
  - Cache entries for this object are automatically distributed to the DynaCaches in other application servers or cooperating Java virtual machines (JVMs). Each cache has a copy of the entry at the time it is created. These entries cannot store non-serializable data.
- **Shared-pull (Deprecated)**
  - Cache entries for this object are shared between application servers on demand. If an application server gets a cache miss for this object, it queries the cooperating application servers to see if they have the object. If no application server has a cached copy of the object, the original application server executes the request and generates the object. These entries cannot store non-serializable data.
  - This mode of sharing is not recommended.
- **Shared-push-pull:**
  - Cache entries for this object are shared between application servers on demand. When an application server generates a cache entry, it broadcasts the cache ID of the created entry to all cooperating application servers.
  - Each server then knows whether an entry exists for any given cache ID. On a given request for that entry, the application server knows whether to generate the entry or pull it from somewhere else.
  - These entries cannot store non-serializable data.