Performance in a virtualized environment

zLG11

Siegfried Langer

Abstract

zLG11 Performance in a virtualized environment

Siegfried Langer

Lecture — Intermediate

Performance tuning is an art. Typically there are no fixed rules to its optimization as many factors are influencing system throughput and resource consumption, as well as service level requirements.

In a virtualized environment this becomes even more complex as virtualized systems may compete with the hypervisor for resources.

The presentation will cover general performance considerations in a virtualized environment with focus on Linux and z/VM.

Agenda

- **Performance tuning**
  - What is different in a virtualized environment
- **Memory/storage management**
  - z/VM storage hierarchy
  - Memory over-commitment
- **Performance guidelines in a z/VM – Linux on System z environment**
  - Processor
  - Disk storage
  - Application considerations
- **More information**
  - Performance monitoring
  - Linux Health Checker
  - Information sources

Definition of Performance

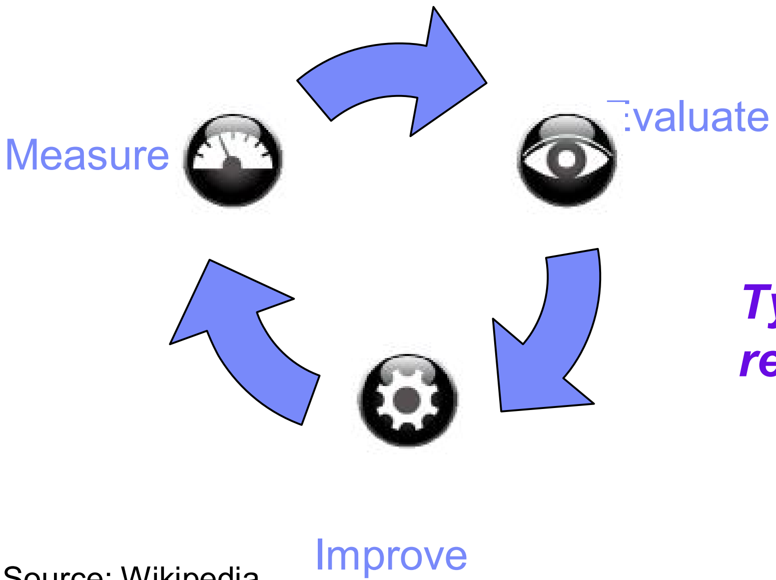
Performance tuning is the improvement of system performance.

- Response time
- Batch elapsed time
- Throughput
- Utilization
- Users supported
- Internal Throughput Rate (ITR)
- External Throughput Rate (ETR)
- Resource consumed per unit of work done
-
-

Performance tuning

Systematic tuning follows these steps:

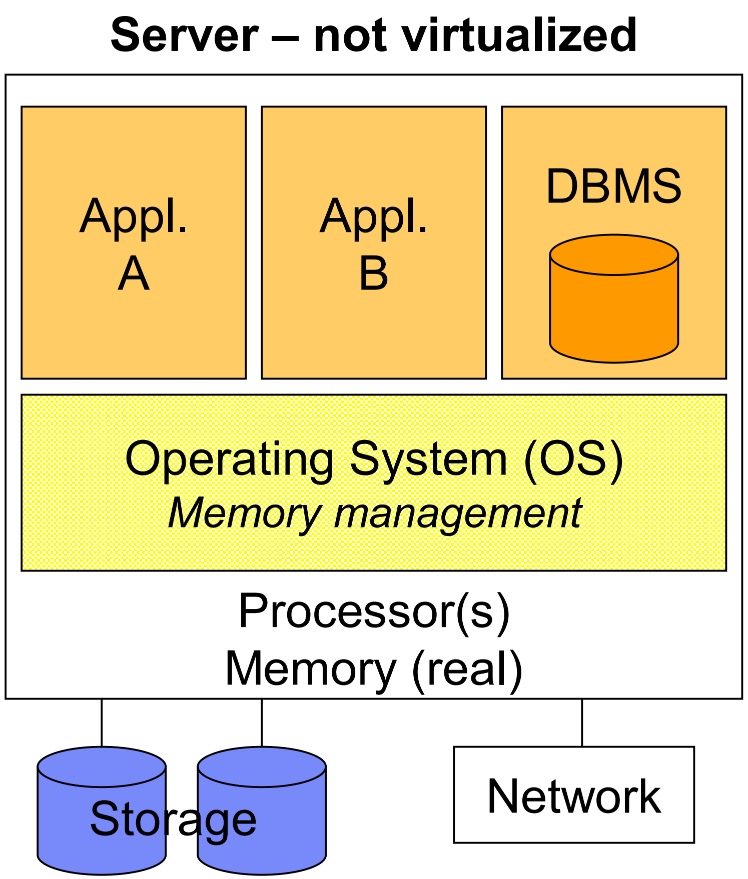
- Assess the problem and establish numeric values that categorize acceptable behavior.
- Measure the performance of the system before modification.
- Identify the part of the system that is critical for improving the performance. This is called the bottleneck.
- Modify the part of the system to remove the bottleneck.
- Measure the performance of the system after modification.



Typically, removing a bottleneck will reveal a new bottleneck in another area!

Source: Wikipedia

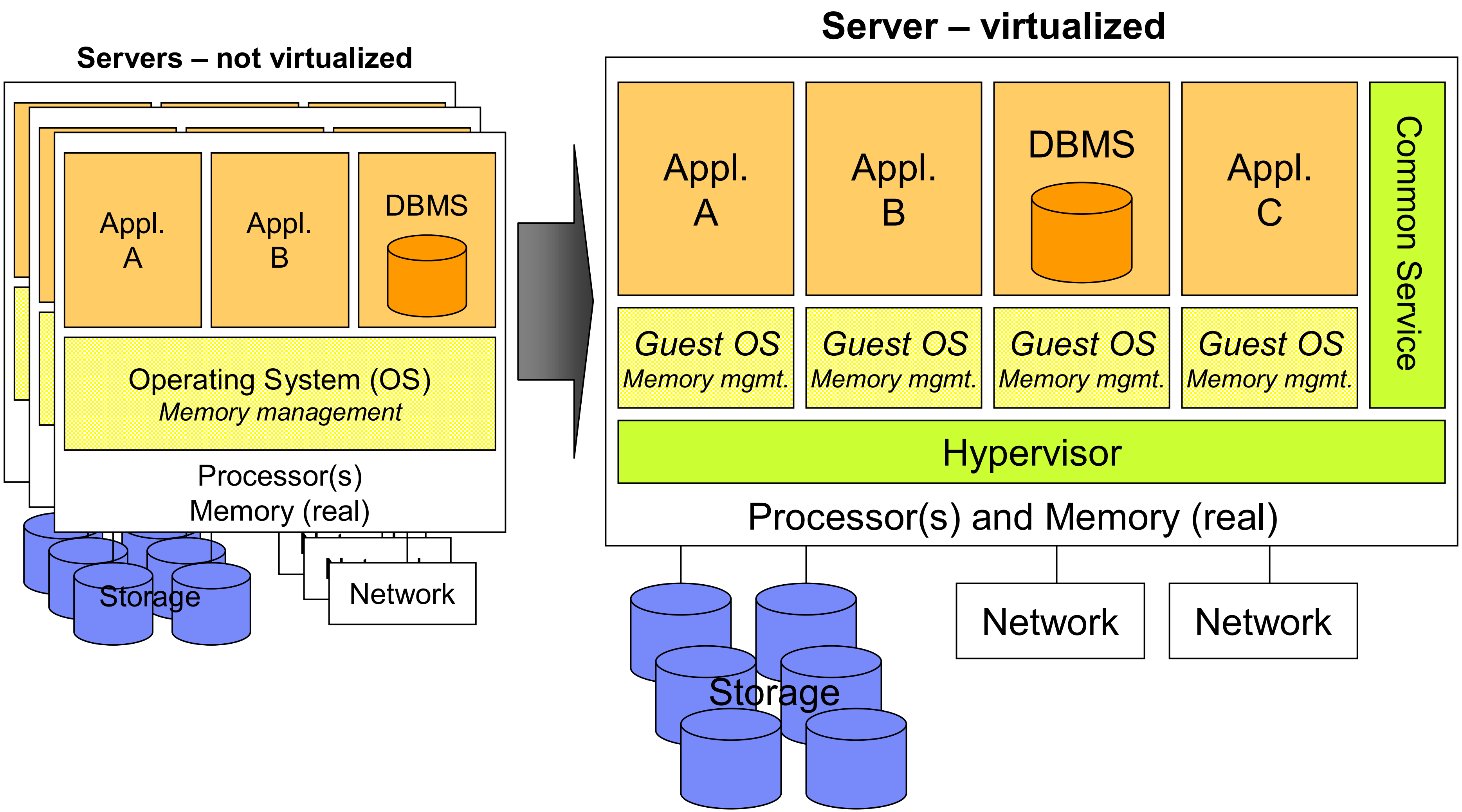
Tuning consideration



- Storage layout
  - Striping
- Memory management
  - Memory layout (heap, etc.)
  - Data in memory
  - Virtual memory
- Priority settings
- Buffers
- Application tuning/optimization
- Database Management System (DBMS)
  - Database physical design
  - DB logical design
  - Buffers/cache size
- Network settings
  - MTU size
  - Buffers
-

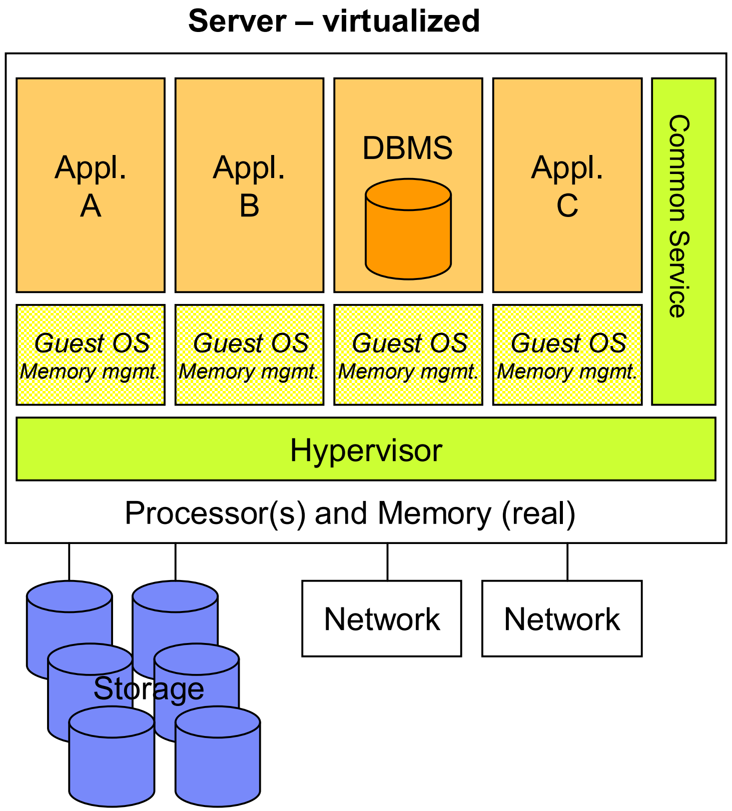
Tune/optimize for most critical application(s)!

Performance consideration



Tuning consideration

- Storage layout
  - Striping
- Memory management
  - Memory layout (heap, etc.)
  - Data in memory
  - Virtual memory
- Priority settings
- Buffers
- Application tuning/optimization
- Database Management System (DBMS)
  - Database physical design
  - DB logical design
  - Buffers/cache size
- Network settings
  - MTU size
  - Buffers
-



Plus:

- Resource allocation (processors, memory, I/O)
- Multi-level memory management
- Internal network
- Virtual I/O
- Common services (e.g., security services)
- more users
-

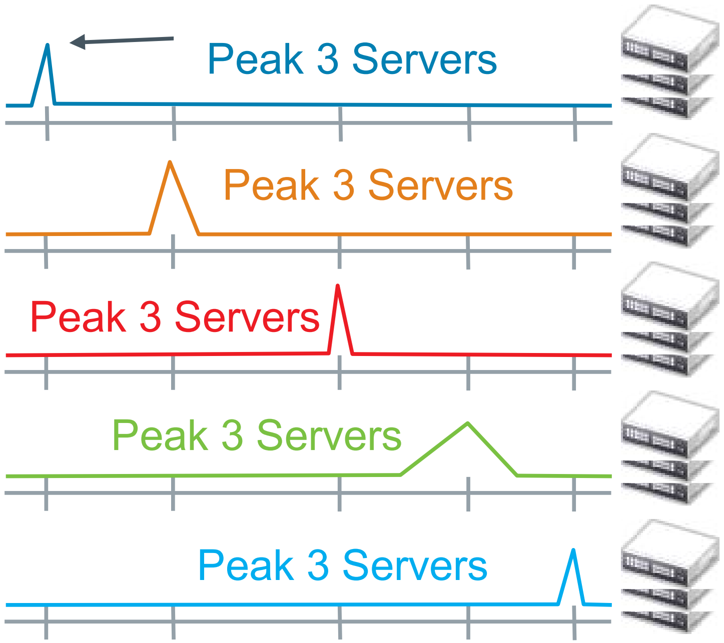
Tune/optimize for a balanced system!

Infrastructure choices for z/VM and Linux on System z

- Linux in z/VM or in LPAR, or both
- z/VM Architecture:
  - How many z/VM
  - How many guests per z/VM
  - How to spread the applications?
  - Which CPU and memory resources for each z/VM?
  - z/VM memory and CPU limitation
- DASD technology:
  - ECKD, SCSI or both (DS8000, XIV, V7000, SVC)
  - Performance SSD vs HDD
  - Striping
  - PAV, HyperPAV
- Network topology:
  - VSWITCH, VLAN
  - Guest LAN
  - HiperSockets
- HA/DR architecture
- z/VM facilities like RACF, Dirmaint, Performance Toolkit, …

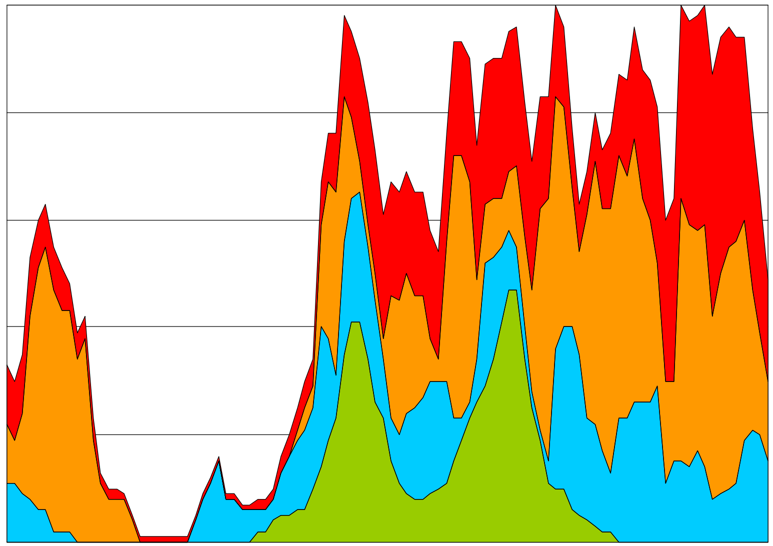
Remember – why consolidation

Utilization on x86 systems



According to a study by Gartner, data centers that do not use virtualization have an average server CPU utilization rate of only 15%.

Mixed Utilization on IBM High End Servers



IBM High End Server: Up to 100% utilization

- Highly virtualized and shared resources
- Fewer servers, less power, cooling & admin
- Optimized use of SW assets

Agenda

- **Performance tuning**
  - What is different in a virtualized environment
- **Memory/storage management**
  - z/VM storage hierarchy
  - Memory over-commitment
- **Performance guidelines in a z/VM – Linux on System z environment**
  - Processor
  - Disk storage
  - Application considerations
- **More information**
  - Performance monitoring
  - Linux Health Checker
  - Information sources

Memory (storage) management

Usually, the term ***storage*** is used by z/VM, while ***memory*** is used by Linux.

- **z/VM**
  - Demand paging between storage areas
  - Block paging with DASD (disk)
  - Steal from central based on LRU with reference bits
  - Steal from expanded based on LRU with timestamps
  - Paging activity is traditionally considered normal
  - Transparent to the guest
  - Virtual storage can be greater than real
- **LPAR**
  - Dedicated storage, no paging
- **Linux**
  - Paging on page basis
  - Swapping activity is traditionally considered bad

z/VM storage hierarchy

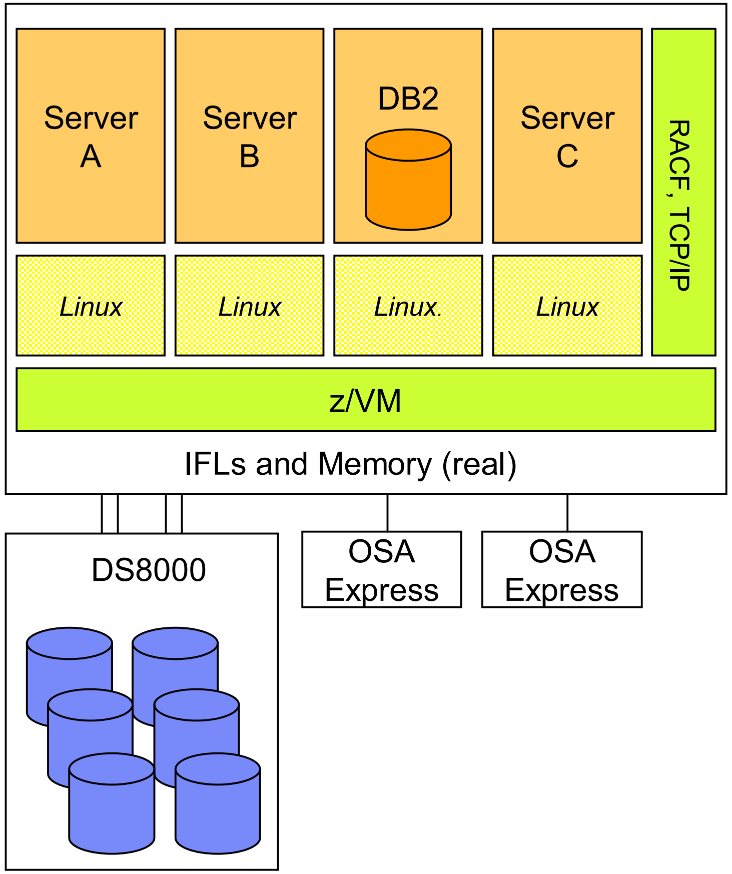
- **Main storage (central storage)**
  - Directly addressable and fast-accessible by user programs
  - Maximum size of main storage is restricted by the amount of physical storage.
- **Expanded storage**
  - Expanded storage also exists in physical storage, but is addressable only as entire pages.
  - Physical storage allocated as expanded storage reduces the amount for main storage.
  - Expanded storage is optional, and its size is configurable.
    Recommendation: configure at least 2 GB Expanded
  - Expanded storage acts as a fast paging device used by z/VM.
- **Paging space**
  - Paging space resides on DASD.
  - When paging demands exceed the capacity of expanded storage, z/VM uses paging space.

Performance guidelines - Paging

- Paging
  - To determine the smallest memory footprint required, decrease the size of the Linux virtual machine to the point where swapping begins to occur under normal load conditions.
  - At that point, slightly increase the virtual machine size to account for some additional load.

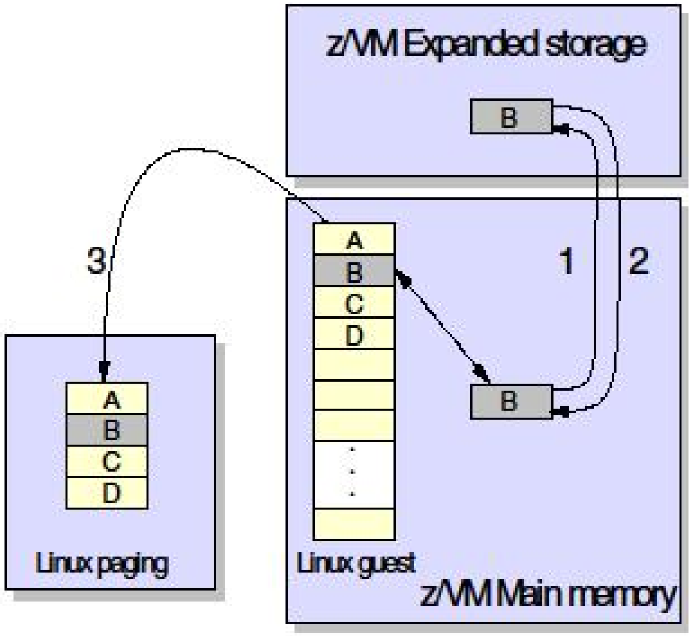
- The general rule does not apply to some sort of servers that have special memory needs.
  - Database servers
    - Database servers maintain buffer pools to prevent excessive I/O to disk. These buffer pools should not be downsized. Otherwise, performance suffers.
  - Servers running Java workload (that is, WebSphere Application Server)
    - An amount of memory is needed to host the Java heap.
    - Too small heap size degrades the performance even if no swapping occurs.

z/VM and Linux on System z



Double paging effect

1. z/VM pages out inactive page
2. Page-out attempt from Linux guest moves page into main memory again
3. Linux completes its page-out attempt and moves page B to swap device



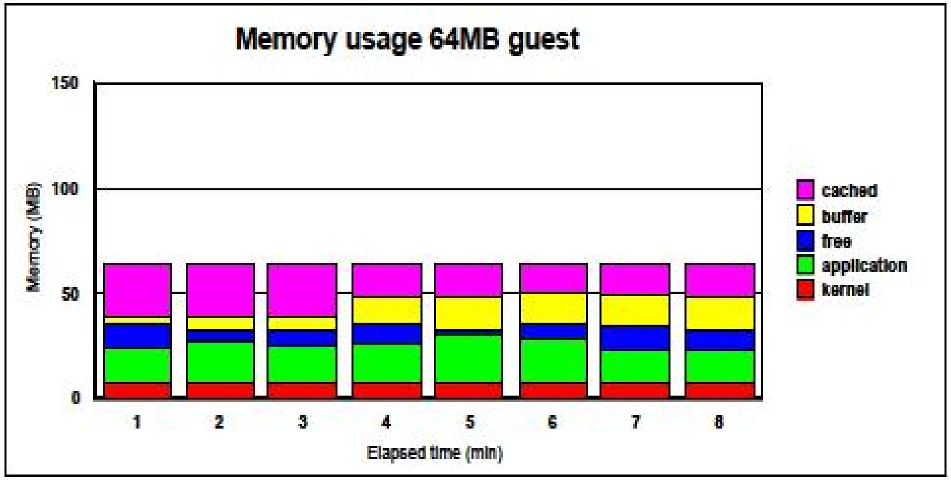
Solution:

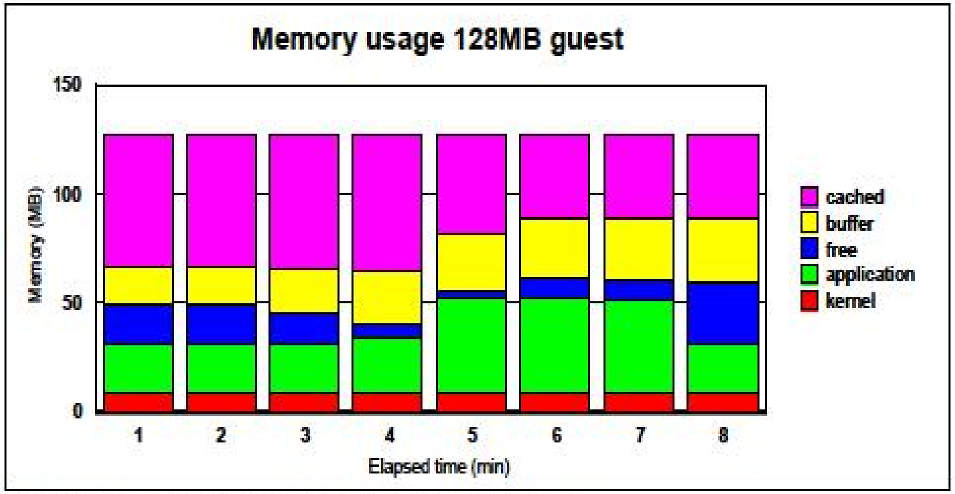
- ***Ensure that one party does not attempt to page!***
- Make the Linux guest virtual machine size small enough for z/VM to keep in main storage.
- Make the virtual machine size large enough that Linux does not attempt to swap.
- Cooperative Memory Management (CMM).
  - Storage usage information is passed from Linux to z/VM.

Aggressive caching within Linux

Linux manages its memory without regard to the fact that it is running in a virtual machine.

- The Linux kernel attempts to load as much information (applications, kernel, cache) into its perceived memory resources as possible and caches it there.
- Comparing the two Linux guests, we see a similar memory usage pattern: In both cases, additional application memory is obtained at the expense of buffer and cache memory.
- Reducing the virtual machine size by 50% reduced average caching by 60%.





Note: *Although the 64 MB guest required half the amount of memory, no appreciable effect on server response time was noted.*

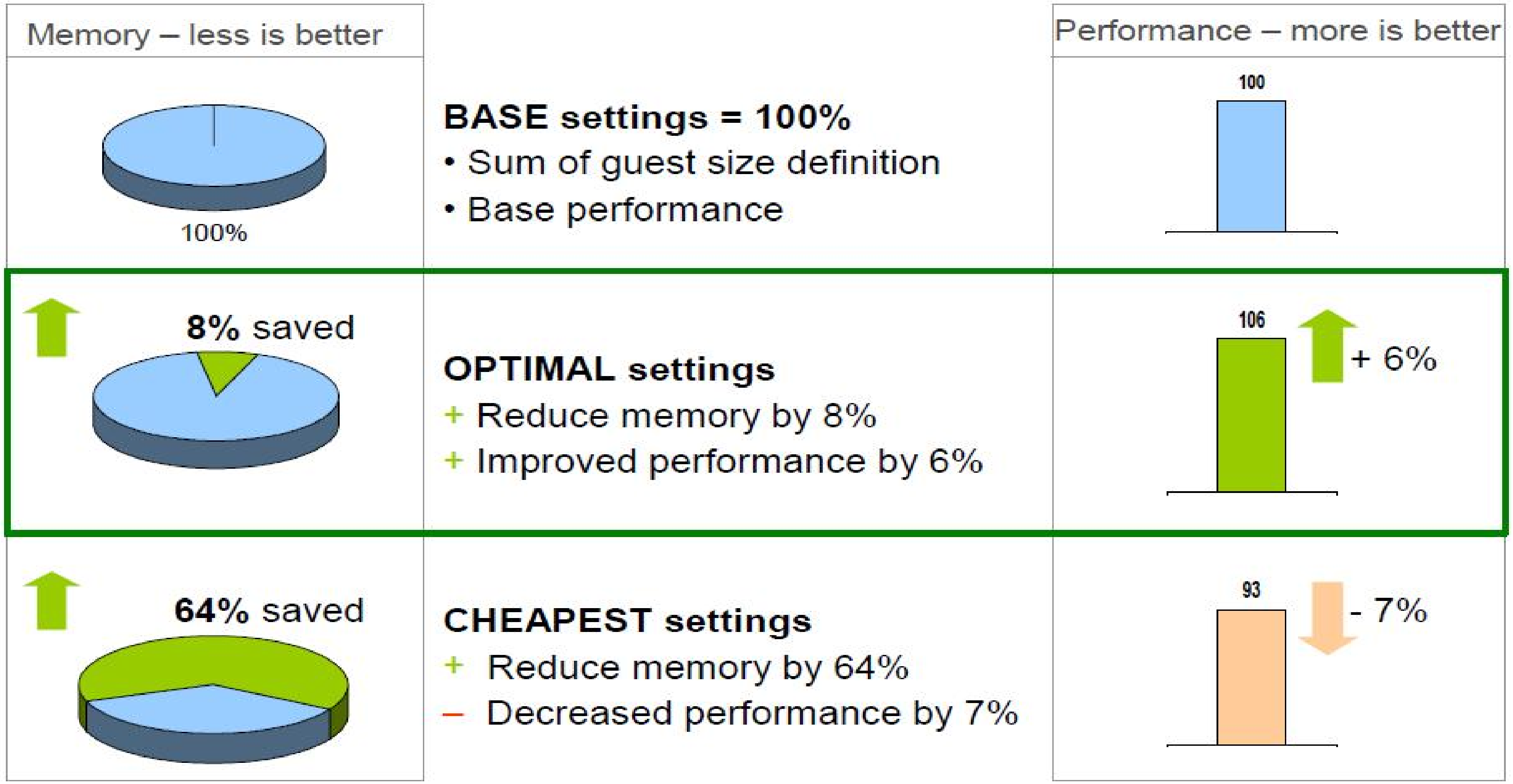
Performance guidelines – Memory & Paging

- Virtual:Real ratio should be ≤ 3:1 or make sure you have a robust paging system
  - To avoid any performance impact for production workloads, you may need to keep ratio to 1:1
  - 1.5:1 might be a good starting point/compromise for many loads
- Use SET RESERVE instead of LOCK to keep users pages in memory
- Define some processor storage as *expanded storage* to provide paging hierarchy
- Exploit shared memory where appropriate
- Size guests appropriately

- Multiple volumes and multiple paths to paging DASD
- Paging volumes should be of the same geometry and performance characteristics
- Paging to FCP SCSI may offer higher paging bandwidth with higher processor requirements
- In a RAID environment, enable cache to mitigate write penalty

Test results – Memory over-commitment

- Running a mix of server types as Linux guests on z/VM
  - LPAR with 28 GB central storage + 2 GB expanded storage
  - Guest workloads: WAS (13.5 GB), DB2 (12.0 GB), Tivoli Directory Server (1.5 GB), idling guest (1.0 GB)
- Leave guest size fixed – decrease LPAR size in predefined steps to scale level of memory over-commitment



Results & recommendations (1 of 2)

- Virtual memory = Physical memory
  - Does not provide the best performance (at least not for large LPARs, e.g. 28GB).

- Optimal memory setting: No z/VM paging !
  - See PerfKit Panel FXC113 User Paging Activity and Storage Utilization and
  - Panel FCX254 Available List Management

- Recommendations (minimum memory requirements):
  - WebSphere Application Server: Sum of all active Java heaps
  - DB2: Sum of MAX_PARTITION_MEM as reported from: "SELECT * FROM TABLE (SYSPROC.ADMIN_GET_DBP_MEM_USAGE()) AS T". Value of PEAK_PARTITION_MEM might be used, when highest allocation is captured.
  - Oracle: Sum of SGA, PGA plus memory for Linux and ASM (if used)

- Linux Page Cache:
  - Sum of read/write throughput,
  - e.g. 20 MB/sec read and 10 MB/sec write throughput require 30 MB/sec pages
    → are ignored in our case in regard to the sizes contributed from WebSphere and DB2

- Idling guests (no kind of server started!): Can be ignored

Results & recommendations (2 of 2)

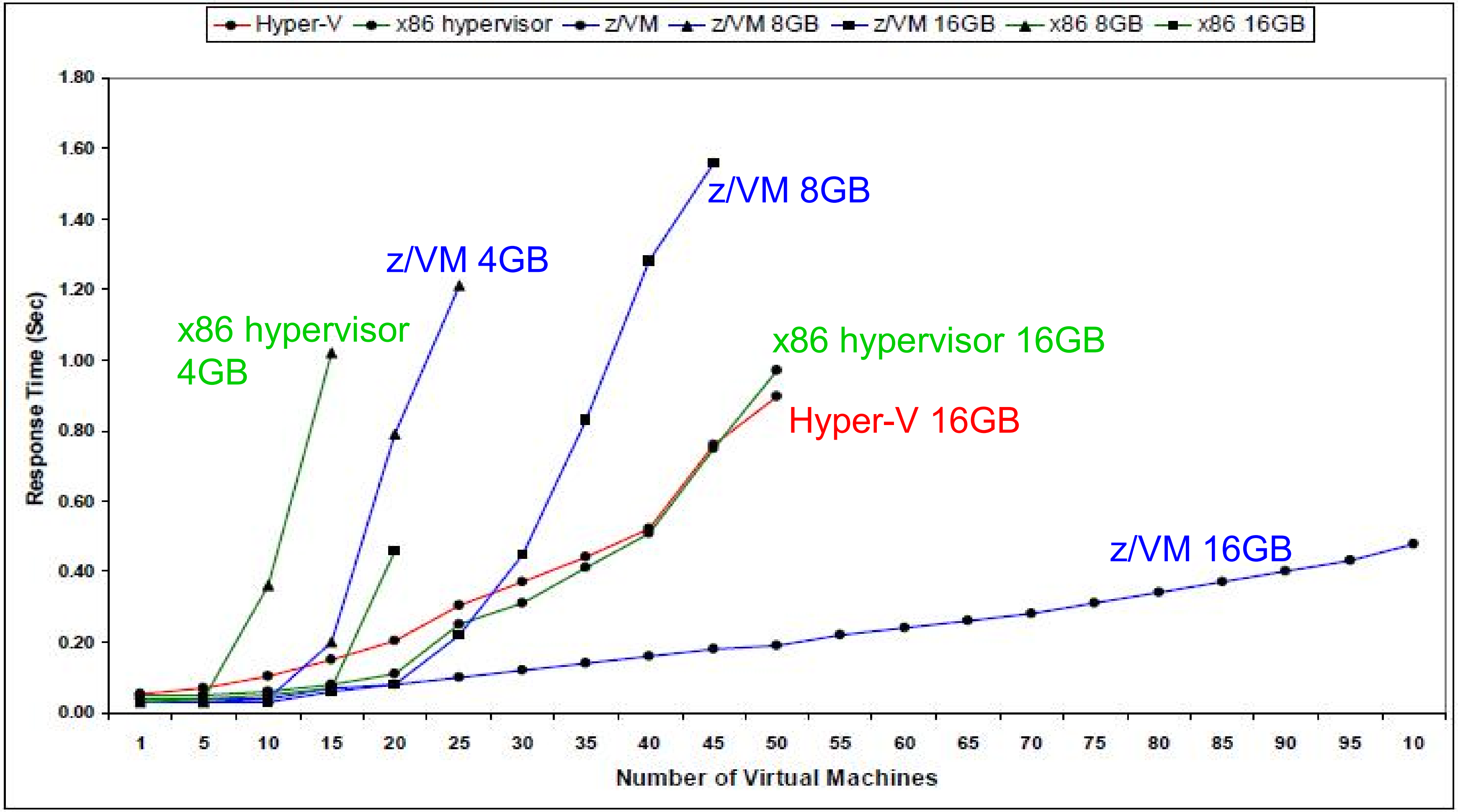
- The minimum memory size defines the lower limit, do not cross!

- Be aware of the dynamic of a virtualized environment
  – New guests are easily created,
  – Java heaps and database pools might be increased without notifying the System z administrator
  – **Monitor paging activity of your system!**

- Other workload types might follow similar considerations

- For more information see
  – Chapter 9. Memory overcommitment in
    Tivoli Provisioning Manager Version 7.1.1.1: Sizing and Capacity Planning
    http://public.dhe.ibm.com/software/dw/linux390/perf/ZSW03168USEN.PDF

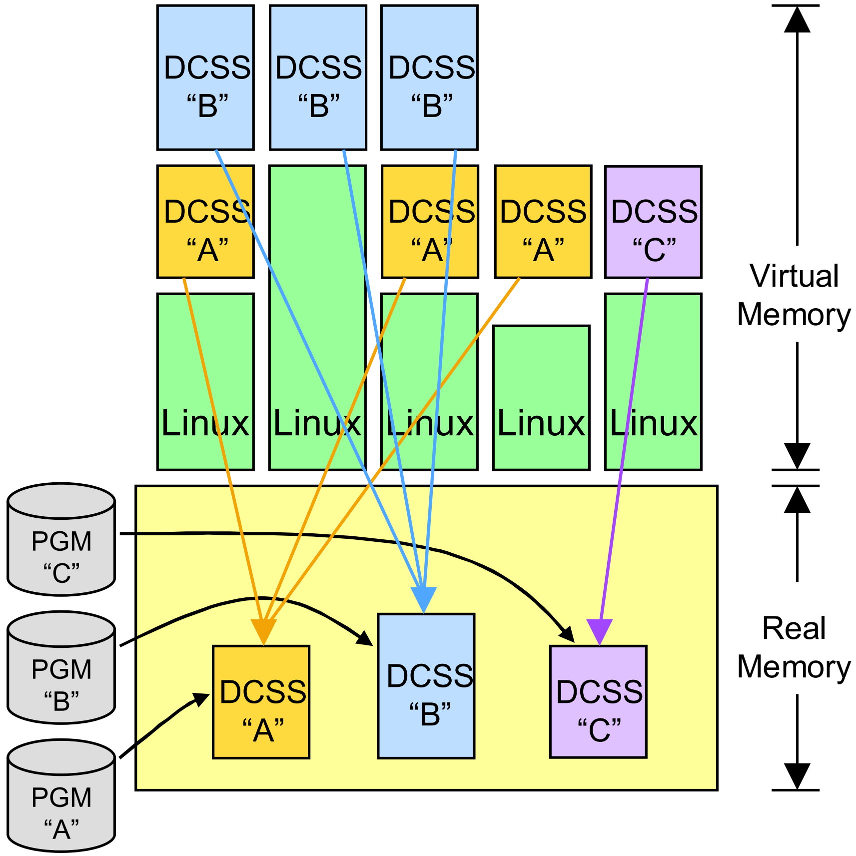
The effect of memory constraints to response time



Extreme Virtualization with Linux on z/VM

Linux Exploitation of z/VM Discontiguous Saved Segments (DCSS)

- **DCSS support is Data-in-Memory technology**
  - Share a single, real memory location among multiple virtual machines
  - Can reduce real memory utilization
- **Linux exploitation: shared program executables**
  - Program executables are stored in an execute-in-place file system, then loaded into a DCSS
  - Execute-in-place (xip2) file system
  - Access to file system is at memory speeds; executables are invoked directly out of the file system (no data movement required)
  - Avoids duplication of virtual memory and data stored on disks
  - Helps enhance overall system performance and scalability



Memory

- Rightsizing Linux memory requirements on z/VM are accomplished by trial and error.

- Don't over-configure Linux memory because:
  - Excess memory allocated to the Linux guest is used by Linux for I/O buffer and File system cache
  - In a virtualized environment under z/VM, oversized guests place unnecessary stress on the VM paging subsystem
    - Real memory is a shared resource, caching pages in a Linux guest reduces memory available to other Linux guests.
    - Larger virtual memory requires more kernel memory for address space management.

Agenda

- **Performance tuning**
  - What is different in a virtualized environment
- **Memory/storage management**
  - z/VM storage hierarchy
  - Memory over-commitment
- **Performance guidelines in a z/VM – Linux on System z environment**
  - Processor
  - Disk storage
  - Application considerations
- **More information**
  - Performance monitoring
  - Linux Health Checker
  - Information sources

Performance guidelines - Processor

- Dedicated processors – mostly political
  - Absolute share can be almost as effective
  - A virtual machine should have all dedicated or all shared processors
  - Gets wait state assist and 500 ms minor slice time
- Share settings
  - Use absolute if you can judge percentage of resources required
  - Use relative if difficult to judge and if slower share as system load increases is acceptable
  - Do not use LIMITHARD settings unnecessarily
- Do not define more virtual processors than are needed
- Small minor time slice keeps processor reactive

CPU management

- In a z/VM guest
  - *Scheduler*
    - Determines priorities based on *Share* setting, etc.
    - Factors in other resource usage and workload characteristics
  - *Dispatcher*
    - *R*uns a virtual processor on a real processor for (up to) a *minor time slice*
    - Virtual processor keeps competing for (up to) an *elapsed time slice*
    - Can dedicate processors to virtual processor

- Natively in a LPAR
  - Uses *Weight* setting like Share setting
  - Dispatches logical partitions on CPs/IFLs
  - Partitions can have dedicated processors

- Linux
  - *Scheduler* handles prioritization and dispatching processes for a time slice or *quantum*

Manage CPU and memory resources with Linux cpuplugd daemon

- Sizing Linux z/VM guests can be a complex task
  - Oversized guests often cause additional management effort by the hypervisor
  - Undersized guests often have performance-related issues with workload peaks

➨ Let the system manage the resources automatically, based on operating requirements of the guest

- Linux cpuplugd (or 'hotplug') daemon
  - Can control CPUs and memory available to a guest
  - Adding or removing resources according to predefined rules
  - Available with SLES 11 SP2 or RHEL 6.2

- IBM whitepaper:

  Using the Linux cpuplugd Daemon to manage CPU and memory resources from z/VM Linux guests

  Dr. Juergen Doelle, Paul V. Sutera; May 2012

  http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/topic/liaag/l0cpup00.pdf

System z on demand capacity

- On/Off Capacity on Demand (On/Off CoD)
  – Allows to temporarily enable PUs and unassigned IFLs available within the current model, or to change capacity settings for CPs to help meet peak workload requirements.

- Capacity Backup Upgrade (CBU)
  – A function that allows the use of spare capacity in a CPC to replace capacity from another CPC within an enterprise, for a limited time.
  – Typically, CBU is used when another CPC of the enterprise has failed or is unavailable because of a disaster event.

Performance guidelines – Disks

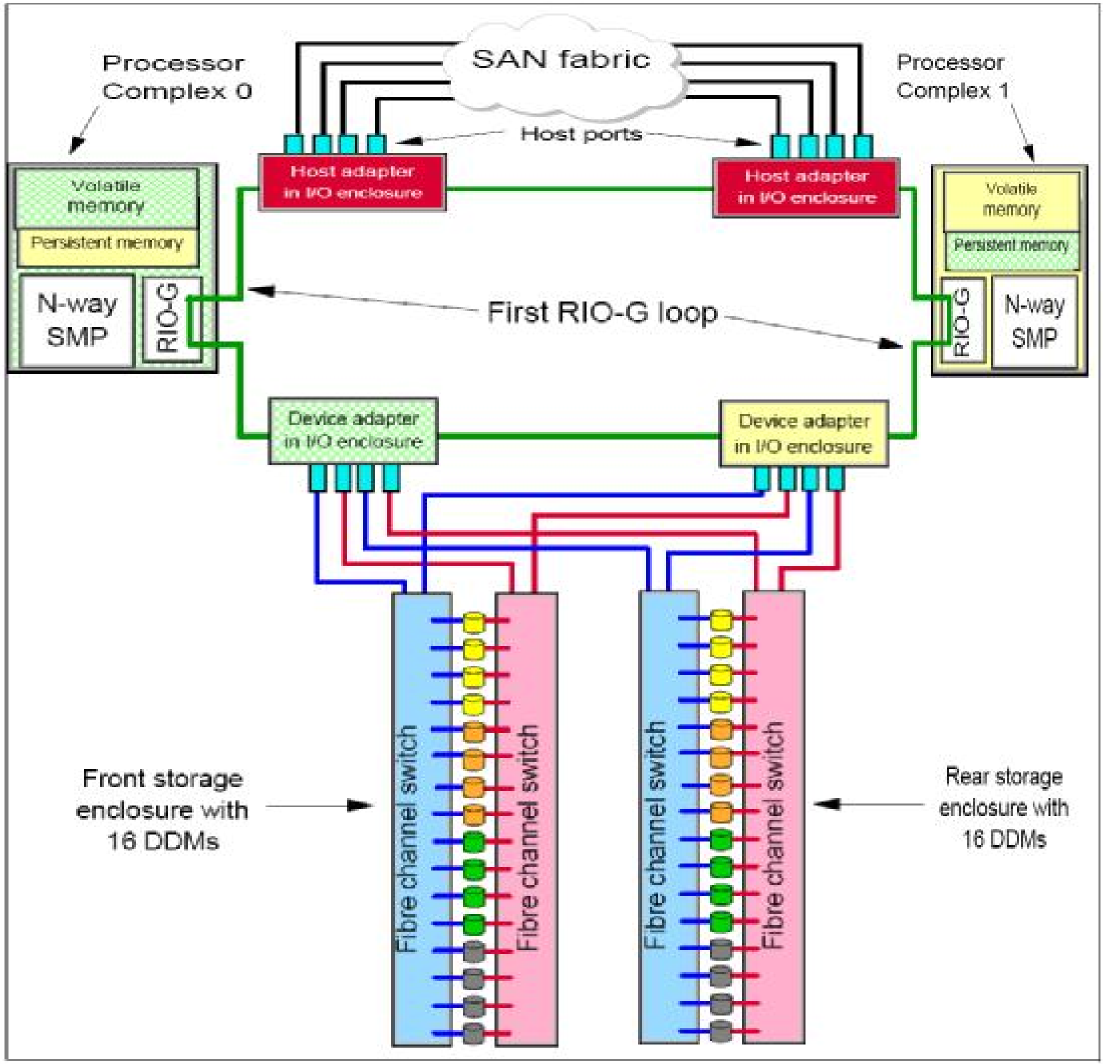
- Don't treat a storage server as a black box, understand its structure.

- Several conveniently selected disks instead of one single disk can speed up the sequential read/write performance to more than a triple. Use the logical volume manager to set up the disks.

- Avoid using subsequent disk addresses in a storage server (e.g., the addresses 5100, 5101, 5102, … in an IBM Storage Server), because
  - they use the same rank
  - they use the same device adapter.

- If you ask for 16 disks and your system administrator gives you addresses 5100-510F
  - from a performance perspective this is close to the worst case

DS8000 Architecture



- **Structure** is complex
  - disks are connected via two internal FCP switches for higher bandwidth
- DS8000 is still divided into two parts
  - Caches are organized per server
- One **device adapter pair** addresses 4 array sites
- One **array site** is build from 8 disks
  - disks are distributed over the front and rear storage enclosures
- One **RAID array** is defined using one array site
- One **rank** is built using one RAID array
- Ranks are assigned to an **extent pool**
- Extent pools are assigned to **one of the servers**
  - this assigns also the caches

ECKD versus SCSI

FICON/ECKD:

- 1:1 mapping host subchannel:dasd
- Serialization of I/Os per sub channel
- I/O request queue in Linux
- Disk blocks are 4KB
- High availability by FICON path groups
- Load balancing by FICON path groups and Parallel Access Volumes

▪ Recommendations:

- Storage pool striped disks
- HyperPAV (requires SLES 11 or RHEL 6)

FCP/SCSI

- Several I/Os can be issued against a LUN immediately
- Queuing in the FICON Express card and/or in the storage server
- Additional I/O request queue in Linux
- Disk blocks are 512 bytes
- High availability by Linux multipathing, type failover or multibus
- Load balancing by Linux multipathing, type multibus

▪ Recommendations:

- Multipathing with failover

FCP/SCSI

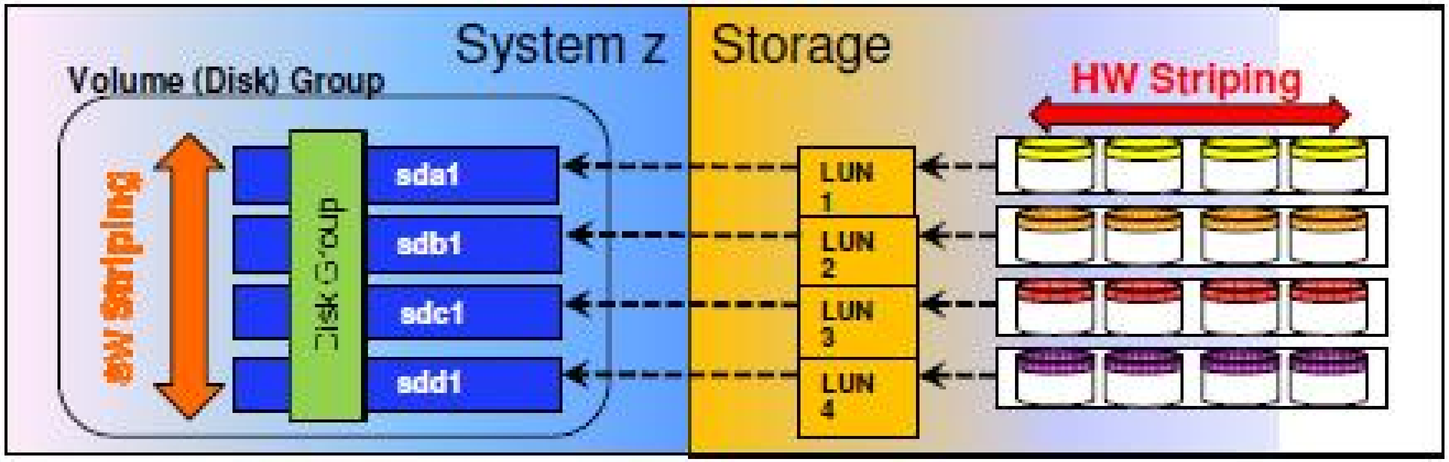
- High availability by Linux multipathing, type failover
  - Should one path fail, the operating system can route I/O through one of the remaining paths (path switching is transparent to the applications)
- Load balancing by Linux multipathing, type multibus
  - Uses all paths in a priority group in a round-robin manner, e.g. path is switched after each 1000 I/O requests
  - Connects a device over multiple independent paths to provide load balancing.

Conclusion:

- When you are using Linux on System z, and you want to establish fail-save connections to the storage server, we recommend to use multipathing.
- Multipath with failover has an performance advantage over multipath with multibus when **multiple** SCSI disks are attached via LVM.
- Multipath with multibus is the solution if you access a **single** SCSI disk and need maximum possible throughput. This increases the bandwidth.

Data Striping to Avoid I/O Hotspots

- **Stripe objects across as many physical disks as possible**
  - Minimal manual intervention
  - Evenly balanced I/O across all available physical components
  - Good average I/O response time object throughput capacity with no hotspots
  - Implementation options:
    - Can be implemented with conventional Volume Managers and file systems
    - Oracle DB: ASM can do this automatically within a given disk group or file system



Why is virtualization a problem for applications?

Physical (dedicated) servers:

- Resources (CPU, memory, network bandwidth, I/O bandwidth) dedicated to the OS.
- Wasting a little of the system resources does not matter on a dedicated server, e.g.
  - A little extra CPU burned by an idling application or background process.
  - Extra memory used. If my application doesn't use it, it will just sit there unused.
  - Network connection is dedicated, so if I'm a little chatty, who cares?
- As long as applications meet their SLAs, that's all that matters.

Virtualized systems:

- That little extra CPU – when multiplied by dozens or hundreds of idling applications lead to significant CPU burn, CPU that's not available for use by other guests.
- That extra memory allocated "just in case" by each application means less memory available to run other guests/applications or supporting paging operations.
- Extra chatter on the shared network connection means less bandwidth available for I/O or other critical application tasks (System z has an advantage here because I/O operations are offloaded from the main CPUs).

How should applications be designed differently because it will be deployed in a virtual server?

- Efficiency in applications matters more in a virtualized environment than when we had the luxury of dedicated hardware.
- For optimum performance and efficiency, applications should be designed with attention to:
  - Idle CPU burn
  - Memory efficiency
  - Main path CPU burn
  - Bandwidth of network and I/O requests

IBM whitepaper "Java Design and Coding for Virtualized Environments" by Steve Wehr, IBM ZSW03224-USEN-01, March 2012

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/5cb5ed706d254a8186256c71006d2e0a/cde7b58f7aaed0d7852579c200756eb7/$FILE/Java%20Design%20and%20Coding%20for%20Virtualized%20Environments.pdf

Agenda

- **Performance tuning**
  – What is different in a virtualized environment

- **Memory/storage management**
  – z/VM storage hierarchy
  – Memory over-commitment

- **Performance guidelines in a z/VM – Linux on System z environment**
  – Processor
  – Disk storage
  – Application considerations

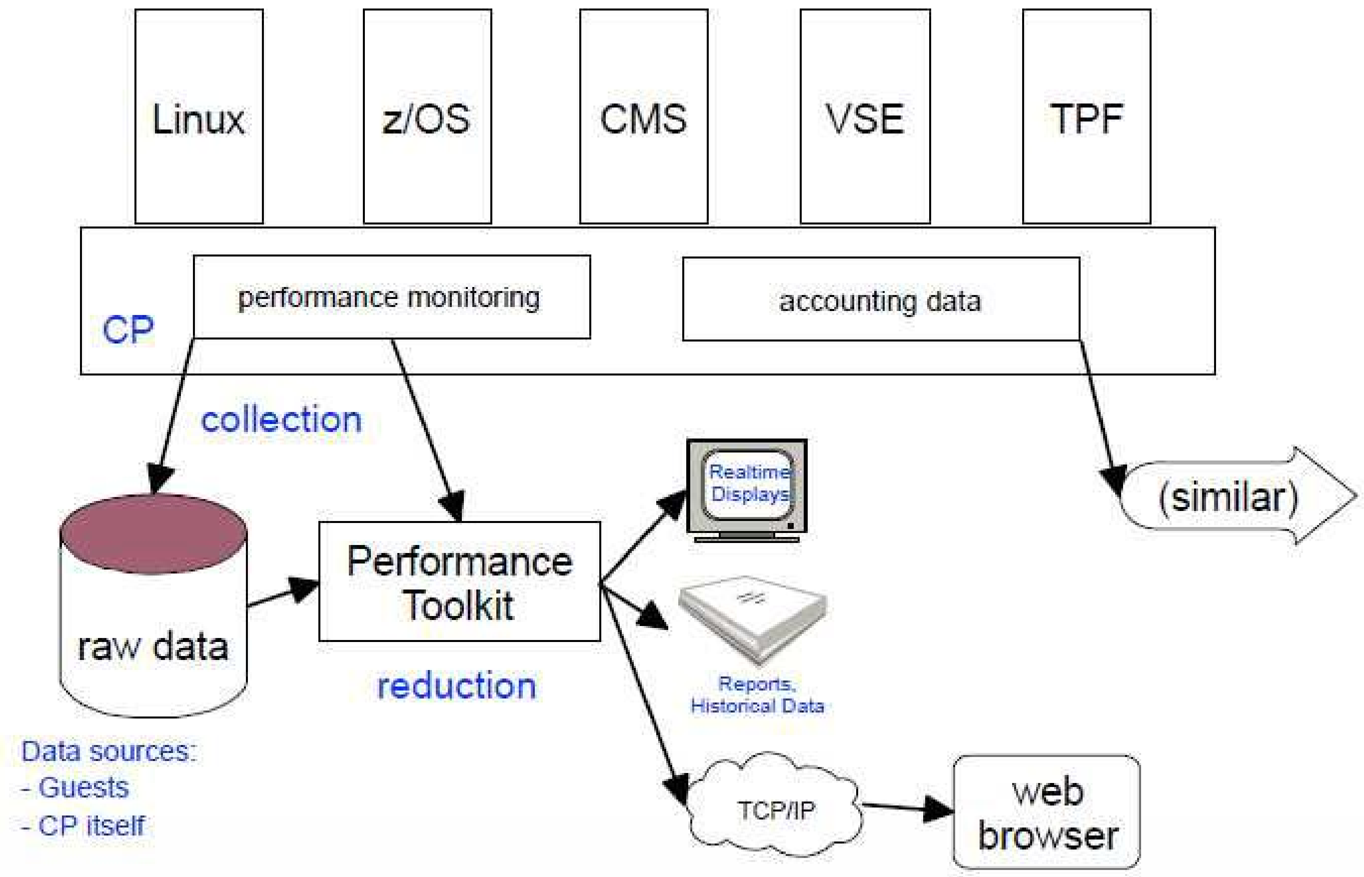
- **More information**
  – Performance monitoring
  – Linux Health Checker
  – Information sources

General Recommendations – Monitoring

Establish permanent monitoring

- **z/VM Performance Toolkit**
  – provides enhanced capabilities for a z/VM systems programmer, system operator, or performance analyst to monitor and report performance data
- **Linux sadc/sar**
  – The **sadc** command samples system data a specified number of times (*count*) at a specified interval measured in seconds (*interval*)
  – The **sar** command writes to standard output the contents of selected cumulative activity counters in the operating system
- **Tivoli OMEGAMON® XE on z/VM® and Linux**
  – Provides a wide range of information about the z/VM and Linux on System z operating systems
  – Data collection from the Performance Toolkit for VM (PTK is a prerequisite) complements data collection by the IBM Tivoli Monitoring for Linux for zSeries® agent
  – High-level views help executives understand how systems performance influences business and the bottom line
  – With granular views, IT staff can more easily track complex problems that span multiple systems and platforms and share related information
  – **Tivoli Composite Application Manager (ITCAM) for Applications – Oracle Agent**
    Provides intelligent monitoring and management of database servers.
    Out-of-the-box views show key metrics unique to each application, including buffer hits, connections used, thread activity, deadlocks and contention.

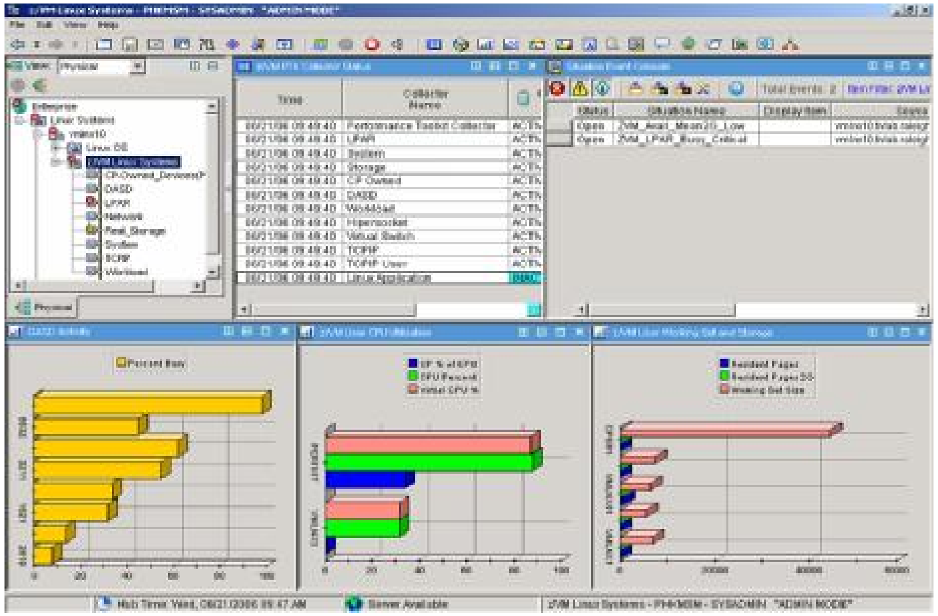
z/VM Performance Toolkit



OMEGAMON XE on z/VM and Linux

A Solution for the Needs of z/VM and Linux on System z

- Single solution for managing VM and Linux on System z
- Reflects most common implementation in marketplace
- Leverages value of z/VM Performance Toolkit



Provides workspaces that display:

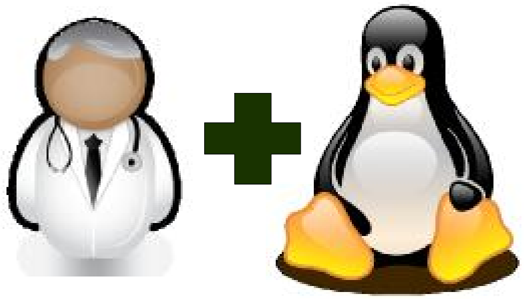
- Overall System Health
- Workload metrics for logged-in users
- Individual device metrics
- LPAR Data

- Composite views of Linux running on z/VM

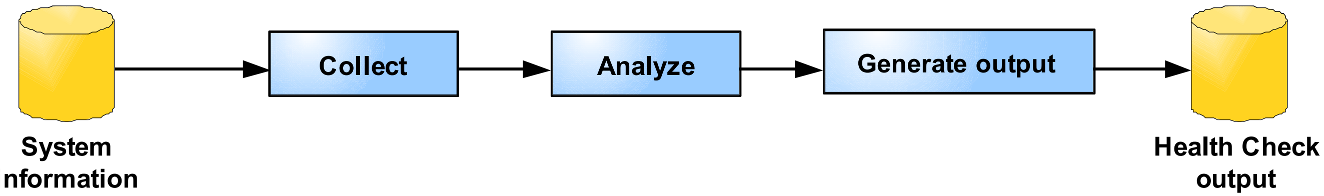
Linux Health Checker

- A tool that performs an automated health check of a Linux system
- Checks status and configuration
- Presents report on identified problems

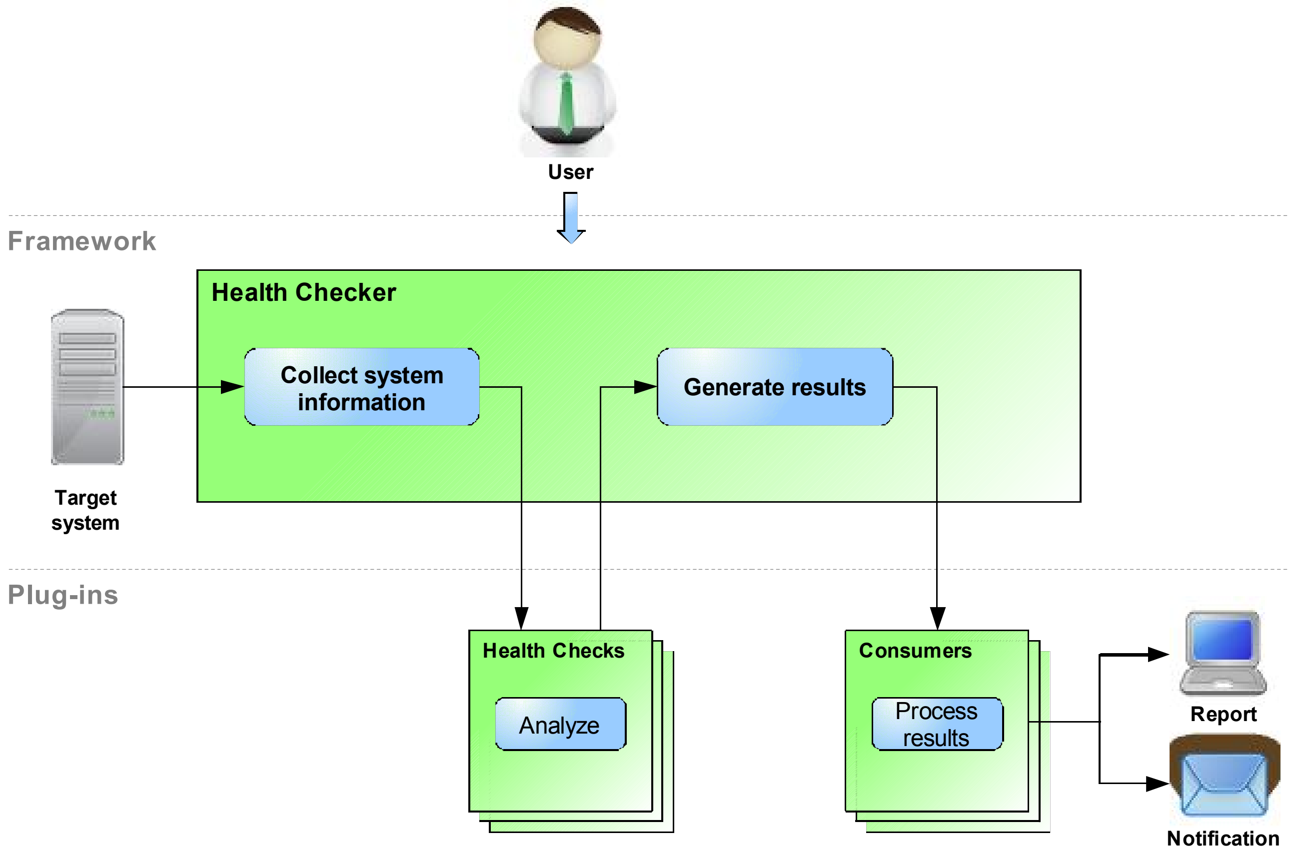
→ **Helps keeping Linux systems healthy (operational)**



- Example problem classes
  - Configuration errors
  - Deviations from best-practice setups
  - Hardware running in degraded mode
  - Unused accelerator hardware
  - Single point-of-failures
- Detailed problem report
  - Enable users to understand and solve problems
  - Make expert knowledge available to wider audience



Linux Health Checker - System overview



- Obtaining the Linux Health Checker
  - Open source under Eclipse Public License v1.0
  - Download RPM or source package from http://lnxhc.sourceforge.net
  - Install using RPM command or make install

Health checks in version 1.0

Verify that the bootmap file is up-to-date

Check whether the path to the OpenSSL library is configured correctly

Identify unusable I/O devices

Check for CHPIDs that are not available

Identify I/O devices that are in use although they are on the exclusion list

Identify I/O devices that are not associated with a device driver

Check for an excessive number of unused I/O devices

Check Linux on z/VM for the "nopav" DASD parameter

Check file systems for adequate free space

Check file systems for an adequate number of free inodes

Check whether the recommended runlevel is used and set as default

Check the kernel message log for out-of-memory (OOM) occurrences

Identify bonding interfaces that aggregate qeth interfaces with the same CHPID

Check for an excessive error ratio for outbound HiperSockets traffic

Check the inbound network traffic for an excessive error or drop ratio

Identify qeth interfaces that do not have an optimal number of buffers

Confirm that the dump-on-panic function is enabled

Screen users with superuser privileges

Identify network services that are known to be insecure

Identify multipath setups that consist of a single path only

Confirm that automatic problem reporting is activated

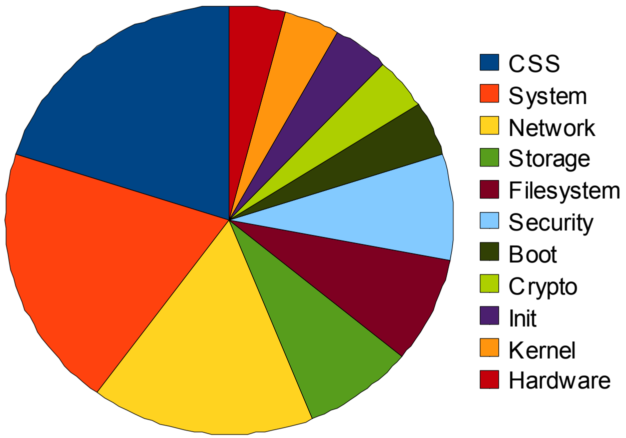
Ensure that panic-on-oops is switched on

Check whether the CPUs run with reduced capacity

Spot getty programs on the /dev/console device

Identify unused terminals (TTY)

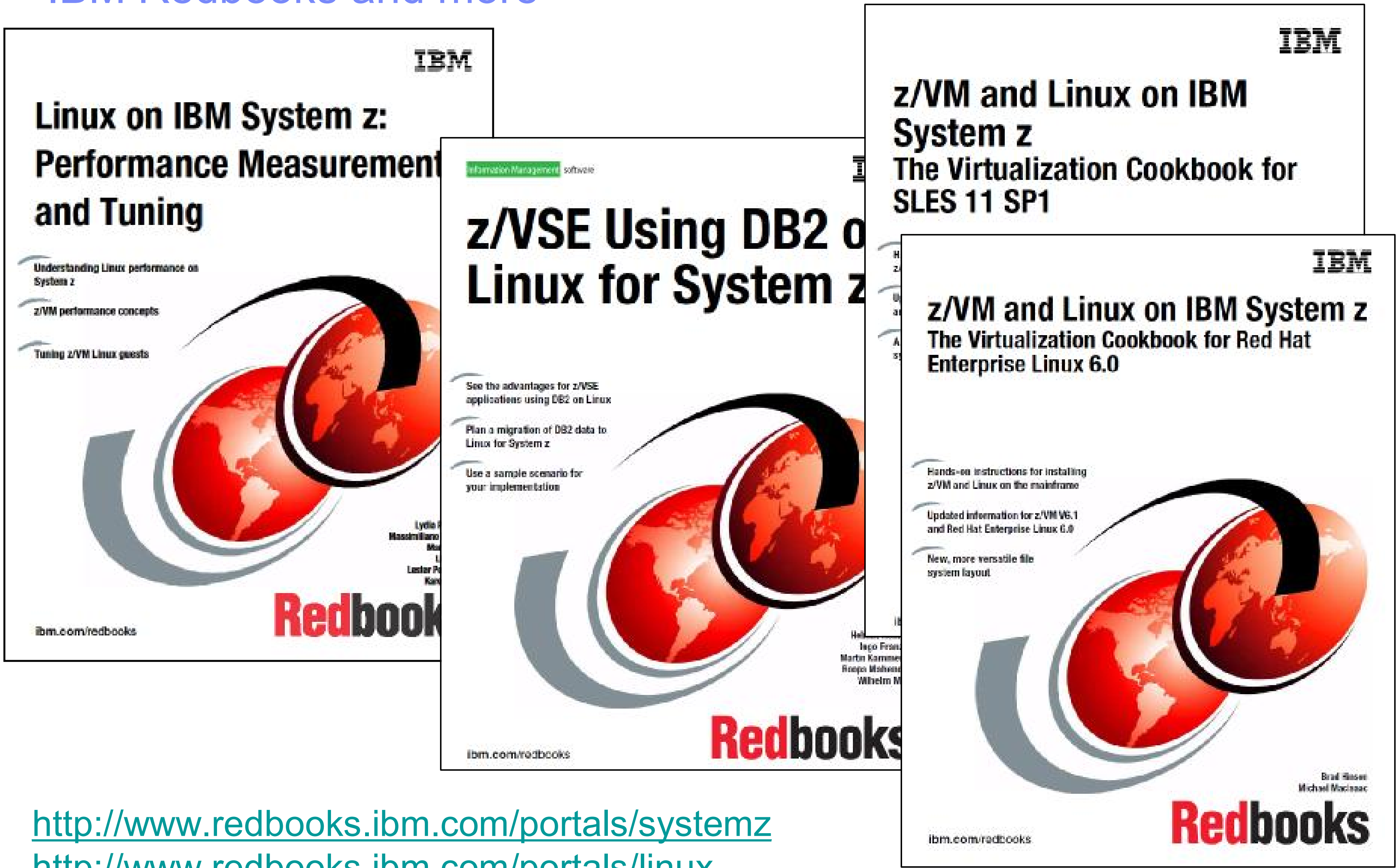
Checks by component



More info sources on performance

- z/VM performance
  - http://www.vm.ibm.com/perf/
  - http://www.vm.ibm.com/perf/tips/linuxper.html
- Linux on System z
  - http://www-03.ibm.com/systems/z/os/linux/resources/doc_pp.html
  - http://www.ibm.com/developerworks/linux/linux390/perf/index.html
- Linux – VM Organization
  - http://www.linuxvm.org/
- IBM Redbooks
  - http://www.redbooks.ibm.com/
- IBM Techdocs
  - http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs

IBM Redbooks and more



http://www.redbooks.ibm.com/portals/systemz

http://www.redbooks.ibm.com/portals/linux

Some final thoughts

- Collect data for a base line of good performance.
- Implement change management process.
- Make as few changes as possible at a time.
- Performance is often only as good as the weakest component.
- Relieving one bottleneck will reveal another. As attributes of one resource change, expect at least one other to change as well.
- Latent demand is real.

Questions?

Siegfried Langer
Business Development Manager
z/VSE & Linux on System z

IBM Deutschland Research & Development GmbH
Schönaicher Strasse 220
71032 Böblingen, Germany

Phone: +49 7031 - 16 4228

Siegfried.Langer@de.ibm.com

Notices

This information was developed for products and services offered in the U.S.A.

Note to U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to: IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of 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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

- This presentation contains trade-marked IBM products and technologies. Refer to the following Web site:

  http://www.ibm.com/legal/copytrade.shtml