Was DBAs über virtualisierte Umgebungen wissen sollten

Siegfried Langer, IBM Deutschland Research & Development

2014-09-08

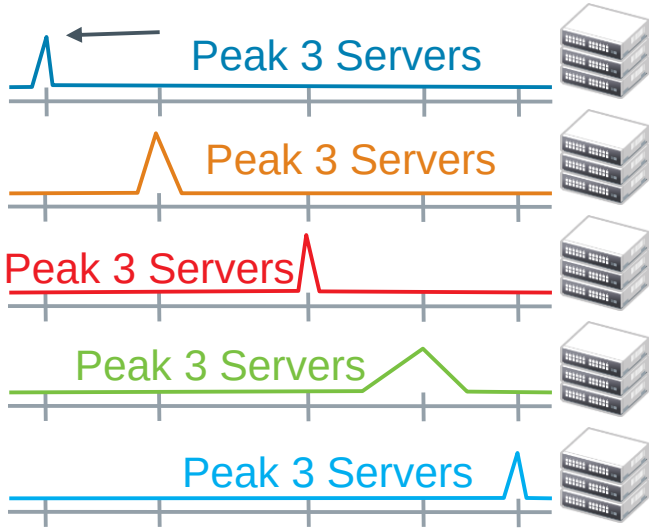
Agenda

- **Warum Virtualisierung?**
- **Performance Tuning in virtualisierten Umgebungen**
- **Best Practices im Performancemanagement**
  - **Besonderheiten in virtualisierten Umgebungen**
  - **Beispiele und Empfehlungen**
    - **VMware**
    - **z/VM Hypervisor**

Warum Virtualisierung?

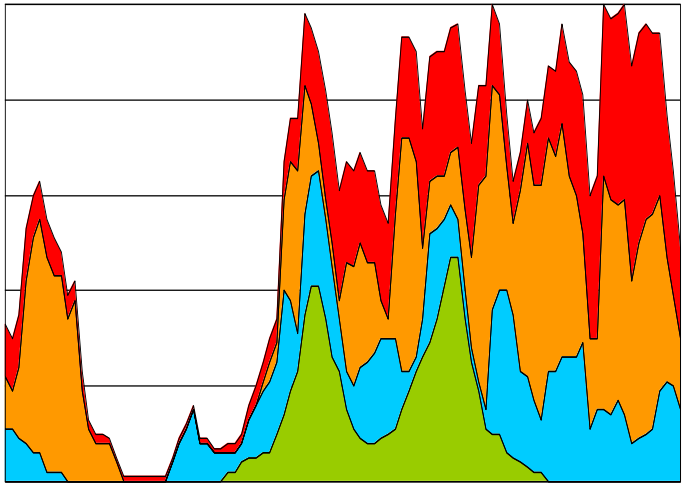
Primäres Ziel ist, eine Abstraktionsschicht zwischen Anwender (etwa einem Betriebssystem) und Ressource (etwa die Hardware des Computers, über die ein Betriebssystem üblicherweise exklusive Kontrolle hat) bereitzustellen. (Wikipedia)

Auslastung auf dedizierten Systemen



Laut einer Gartner Studie liegt die CPU-Auslastung in Rechenzentren ohne Virtualisierung im Durchschnitt bei nur 15%.

Auslastung auf großen virtualisierten Servern



IBM High-End Server: bis zu 100% Auslastung

- Hoch-virtualisiert und gemeinsame Ressourcennutzung
- Weniger Server, weniger Stromverbrauch, Kühlung und Administration
- Optimierte Nutzung der Softwarelizenzen

Reduzierung der Softwarekosten durch Konsolidierung
Beispiel: Oracle Datenbanken

- Lizenzen und jährliche “Software Update License & Support” Kosten basieren auf Prozessorkernen (Enterprise Edition) oder Sockets (Standard Edition)
- Ein “processor core factor” dient zur Anpassung an unterschiedliche Technologien

ORACLE

Oracle Technology Global Price List
August 07, 2014
Software Investment Guide

http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf

Software Update Licenses & Support (jährlich) ist typisch 22% der Prozessor Lizenz (Einmalzahlung)

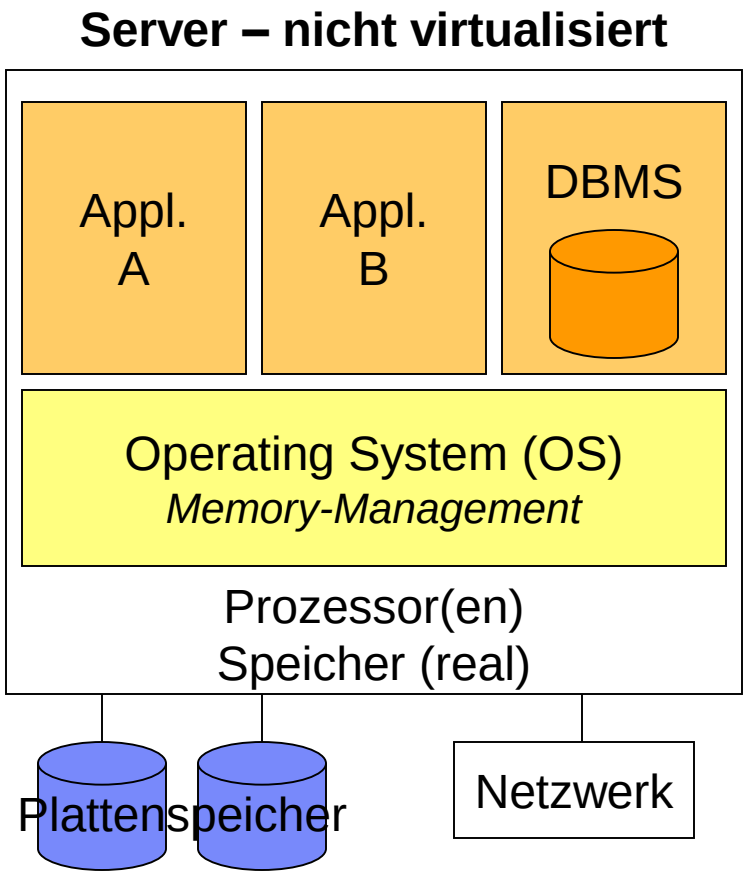
ORACLE

Oracle Processor Core Factor Table
Effective Date: March 16, 2009

Vendor and Processor	Core Processor Licensing Factor
Intel Xeon Series 56XX, Series 65XX, Series 75XX, Series E7-28XX, Series E7-48XX, Series E7-88XX, Series E5-24XX, Series E5-26XX, Series E5-46XX, Series E5-16XX, Series E3-12XX or earlier Multicore chips	0.5
Intel Itanium Series 93XX (For servers purchased on or after Dec 1st, 2010)	1.0
IBM POWER6	1.0
IBM POWER7	1.0
IBM System z (z10 and earlier)	1.0
All Other Multicore chips	1.0

http://www.oracle.com/us/corporate/contracts/processor-core-factor-table-070634.pdf

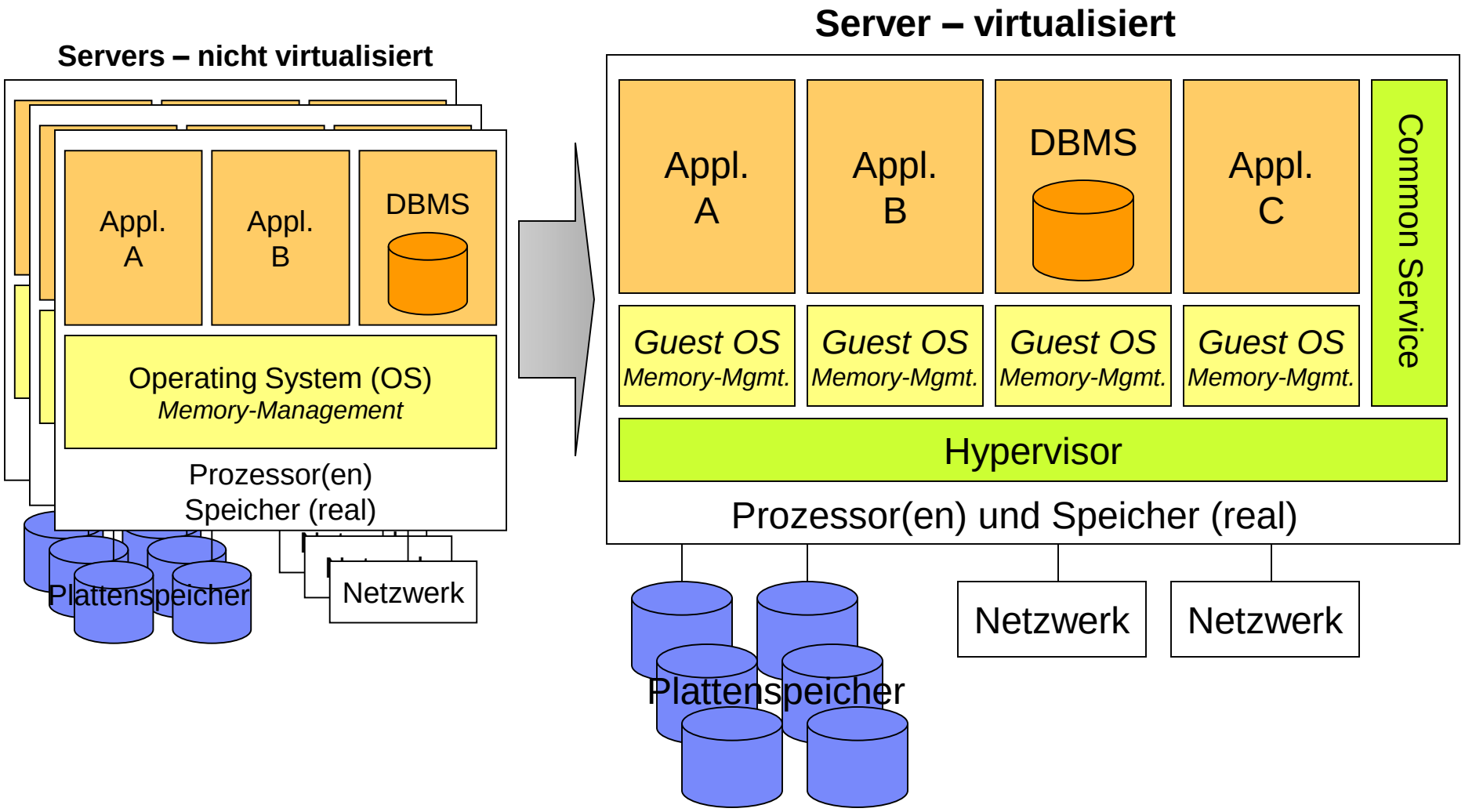
Tuning-Aspekte



- Plattenspeicher-Layout
  - Striping
- Memory-Management
  - Speicher-Layout (Heap, etc.)
  - Data in Memory
  - Virtueller Speicher
- Prioritätseinstellungen
- Puffer
- Anwendungstuning/-optimierung
- Database Management System (DBMS)
  - Datenbank physisches Design
  - DB logisches Design
  - Buffer/Cache Größe
- Netzwerkeinstellungen
  - MTU Größe
  - Buffers
-

Tuning und Optimierung der kritischsten Anwendung(en)!

Tuning-Aspekte in virtualisierter Umgebung

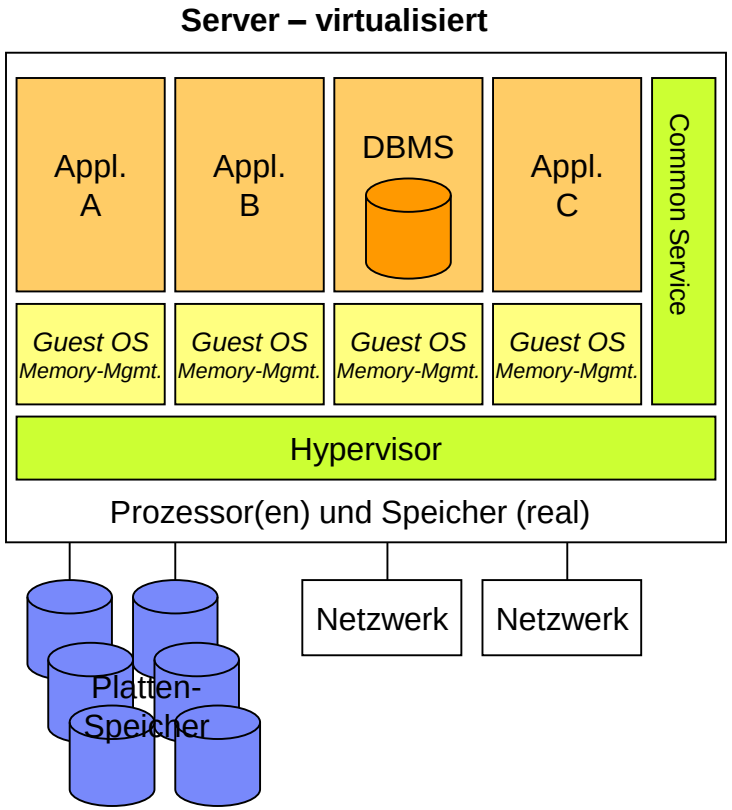


Tuning-Aspekte in virtualisierter Umgebung

- Plattenspeicher-Layout
  - Striping
- Memory-Management
  - Speicher-Layout (Heap, etc.)
  - Data in Memory
  - Virtueller Speicher
- Prioritätseinstellungen
- Puffer
- Anwendungstuning/-optimierung
- Database Management System (DBMS)
  - Datenbank physisches Design
  - DB logisches Design
  - Buffer/Cache Größe
- Netzwerkeinstellungen
  - MTU Größe
  - Buffers

Plus:

- Mehrere Server in einem System
- Allokation der Ressourcen (Prozessoren, Speicher, I/O, Netzwerk)
- Mehrstufiges Speichermanagement
- Internes und externes Netzwerk
- Virtualisiertes I/O
- Allgemeine Systemservices (z.B. Security)
- mehr Benutzer



Tuning und Optimierung für ein ausbalanciertes System!

Definition von Performance

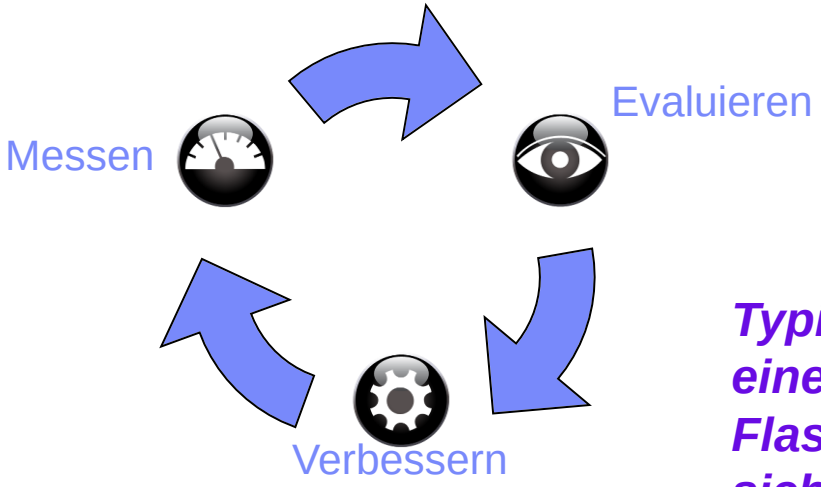
Performance-Tuning ist die Verbesserung der System-Performance.

- Antwortzeiten
- Stapelverarbeitungszeiten (Batch)
- Durchsatz
- Ressourcennutzung (Utilization)
- Anzahl der unterstützten Benutzer
- Internal Throughput Rate (ITR)
- External Throughput Rate (ETR)
- Ressource Verbrauch pro durchgeführter Arbeitseinheit (unit of work)
-
-

Performance Tuning

Systematisches Tuning folgt diesen Prozessschritten:

- Untersuchung des Problems und Definition numerischer Werte, die ein akzeptables Systemverhalten darstellen
- Messen der Performance des Systems bevor Modifikationen vorgenommen werden
- Identifikation der Systemteile, die für die Verbesserung kritisch sind. Dies sind die Flaschenhälse (bottleneck).
- Modifikation eines Teils des Systems um den Flaschenhals zu beseitigen.
- Messen der Performance des Systems nach der Modifikation.



Typischerweise führt die Beseitigung eines Flaschenhalses dazu, dass neue Flaschenhälse in einem anderen Bereich sichtbar werden!

Quelle: Wikipedia

Agenda

- Warum Virtualisierung?
- Performance Tuning in virtualisierten Umgebungen
- **Best Practices im Performancemanagement**
  - **Besonderheiten in virtualisierten Umgebungen**
  - **VMware**
  - **z/VM Hypervisor**
- Performance-Tipps und Beispiele

Tuning recommendations

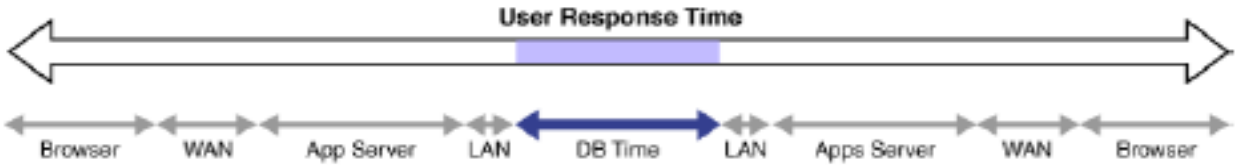
Oracle® Database
2 Day + Performance Tuning Guide
12*c* Release 1 (12.1)
E17635-10

Using the Oracle Performance Method

Performance tuning using the Oracle performance method is driven by identifying and eliminating bottlenecks in the database, and by developing efficient SQL statements. Database tuning is performed in two phases: proactively and reactively.

- In the proactive tuning phase, you must perform tuning tasks as part of your daily database maintenance routine, such as reviewing ADDM analysis and findings, monitoring the real-time performance of the database, and responding to alerts.
- In the reactive tuning phase, you must respond to issues reported by users, such as performance problems that may occur for only a short duration of time, or performance degradation to the database over a period of time.

SQL tuning is an iterative process to identify, tune, and improve the efficiency of high-load SQL statements.



Common Performance Problems Found in Databases

- **CPU bottlenecks**
- **Undersized memory structures**
- **I/O capacity issues**
- Sub optimal use of Oracle Database by the application
- **Concurrency issues**
- Database configuration issues
- **Short-lived performance problems**
- **Degradation of database performance over time**
- Inefficient or high-load SQL statements
- Object contention
- Unexpected performance regression after tuning SQL statements

Oracle® Database
2 Day + Performance Tuning Guide
12*c* Release 1 (12.1)
E17635-10

These problems may be aggravated in a virtualized environment

Virtuelle CPUs und Speicher

- Definiere nicht mehr virtuelle CPUs für einen Linux-Gast als nötig!
  - Die Nutzung mehrerer Prozessoren benötigt Software-Locks, sodass Daten oder Kontrollblöcke nur von einem Prozessor zu einer Zeit geändert werden können.
  - Linux nutzt ein globales Lock. Wenn das Lock gehalten wird und ein anderer Prozessor es benötigt, muss er warten.
  - Die Zahl der virtuellen Prozessoren sollte nach dem Bedarf gesetzt werden und nicht einfach der Anzahl der realen Prozessoren entsprechen.
  - Vorsicht beim Clonen: einige Linux-Gäste brauchen mehr virtuelle CPUs als andere, z.B. Oracle.

- Definiere den (virtuellen) Speicher des Linux nicht größer als nötig!
  - Exzessive virtuelle Speichergrößen haben eine negative Auswirkung auf die Performance.
  - Linux nutzt freien Speicher für das Caching von Daten. Für gemeinsam genutzte (shared) Ressourcen hat dies negative Auswirkungen.
  - Reduziere die Größe des Linux-Gastes, bis er beginnt Speicher auszulagern (Swap).
  - Benutze VDISK für Swap (wenn genügend realer Speicher verfügbar).
  - Vergleiche die Linux Speichernutzung mit den im Hypervisor definierten Größen des Gastes.

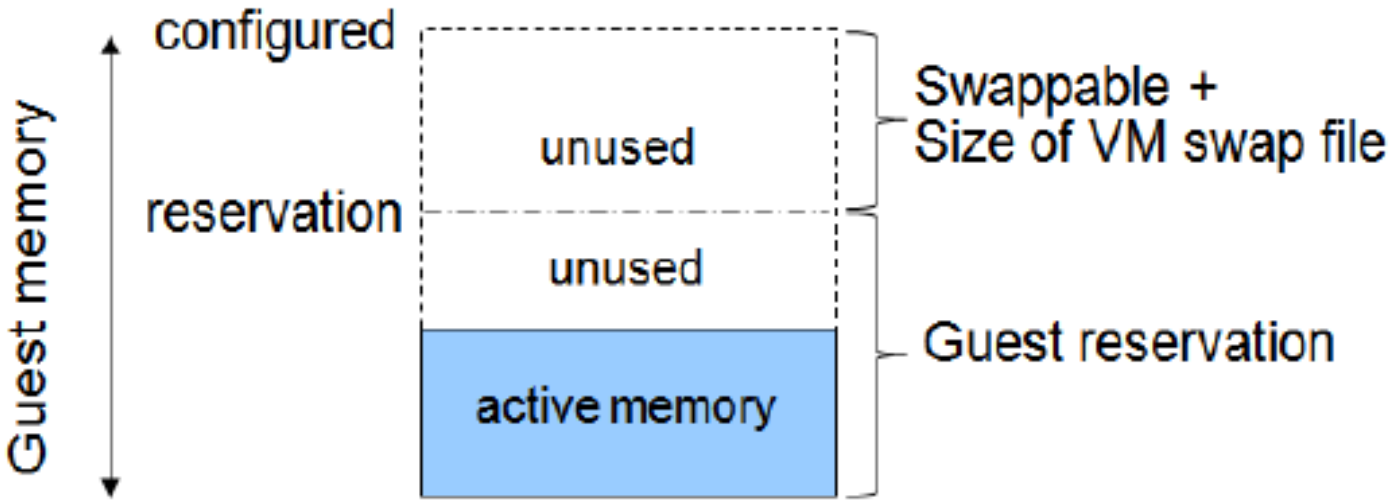
Oracle Databases on VMware – Best Practices Guide

- **4.2 Memory**
  - Recommendation:
    - Set memory reservations equal to the size of the Oracle SGA.
  - Justification:
    - The memory reservation should be large enough to avoid kernel swapping between ESX and the guest OS because Oracle databases can be memory-intensive.

- **4.3 Virtual CPU**
  - Recommendation:
    - Use as few virtual CPUs (vCPUs) as possible.
  - Justification:
    - If monitoring of the actual workload shows that the Oracle database is not benefiting from the increased virtual CPUs, the excess vCPUs impose scheduling constraints and can degrade overall performance of the virtual machine.

Source: Oracle Databases on VMware Best Practices Guide

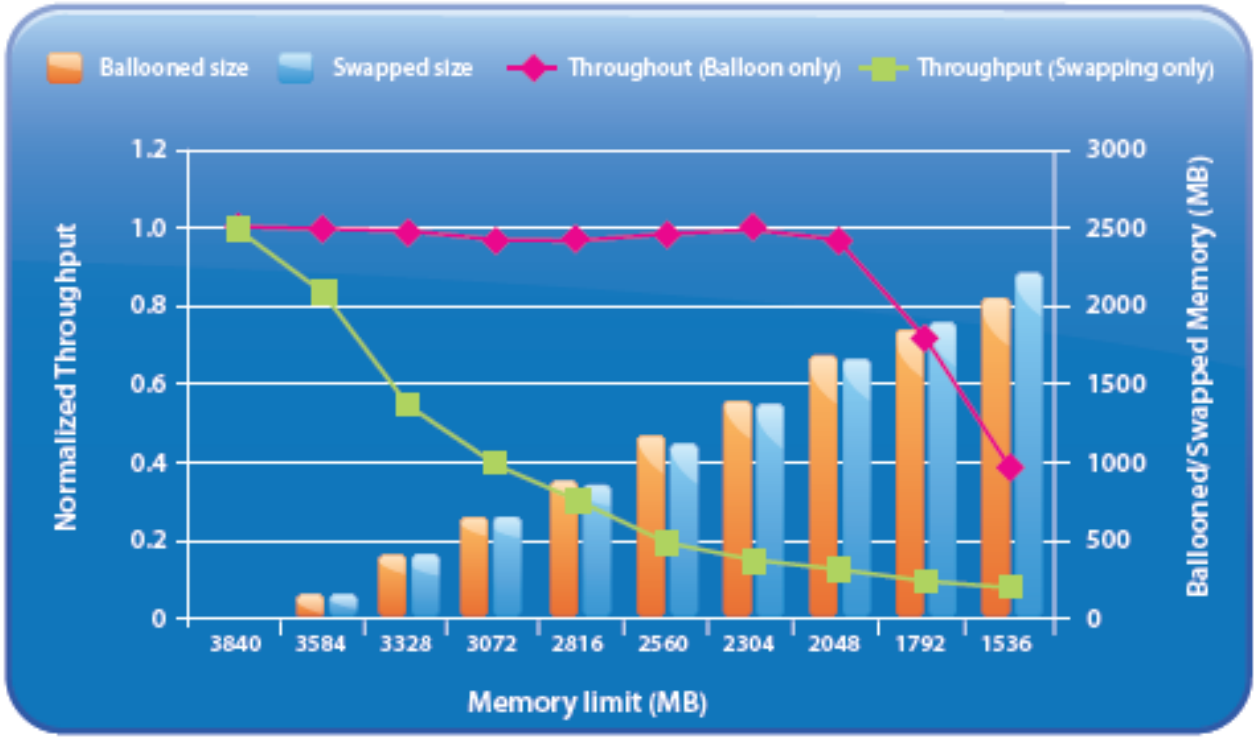
Virtual Machine Memory Settings



- Definition of terms:
  - Configured memory – Memory size of virtual machine assigned at creation.
  - Active memory – Memory recently accessed by applications in the virtual machine.
  - Reservation – Guaranteed lower bound on the amount of memory that the host reserves for the virtual machine, which cannot be reclaimed by ESX/ESXi for other virtual machines.
  - Swappable – Virtual machine memory that can be reclaimed by the balloon driver or, in the worst case, by ESX/ESXi swapping. This is the automatic size of the swap file that is created for each virtual machine on the VMFS file system (.vswpfile).

Source: Oracle Databases on VMware Best Practices Guide

VMware mit Oracle/Swingbench Performance Test



- Ballooning vs. swapping
  - Throughput is normalized to the case where virtual machine memory is not reclaimed
  - Using ballooning barely impacts the throughput of the Swingbench virtual machine until the memory limit decreases below 2048MB.
  - This occurs when the guest operating system starts to page out the physical pages that are heavily reused by the Oracle database.

Source: VMware whitepaper: Understanding Memory Resource Management in VMware® ESX™ Server

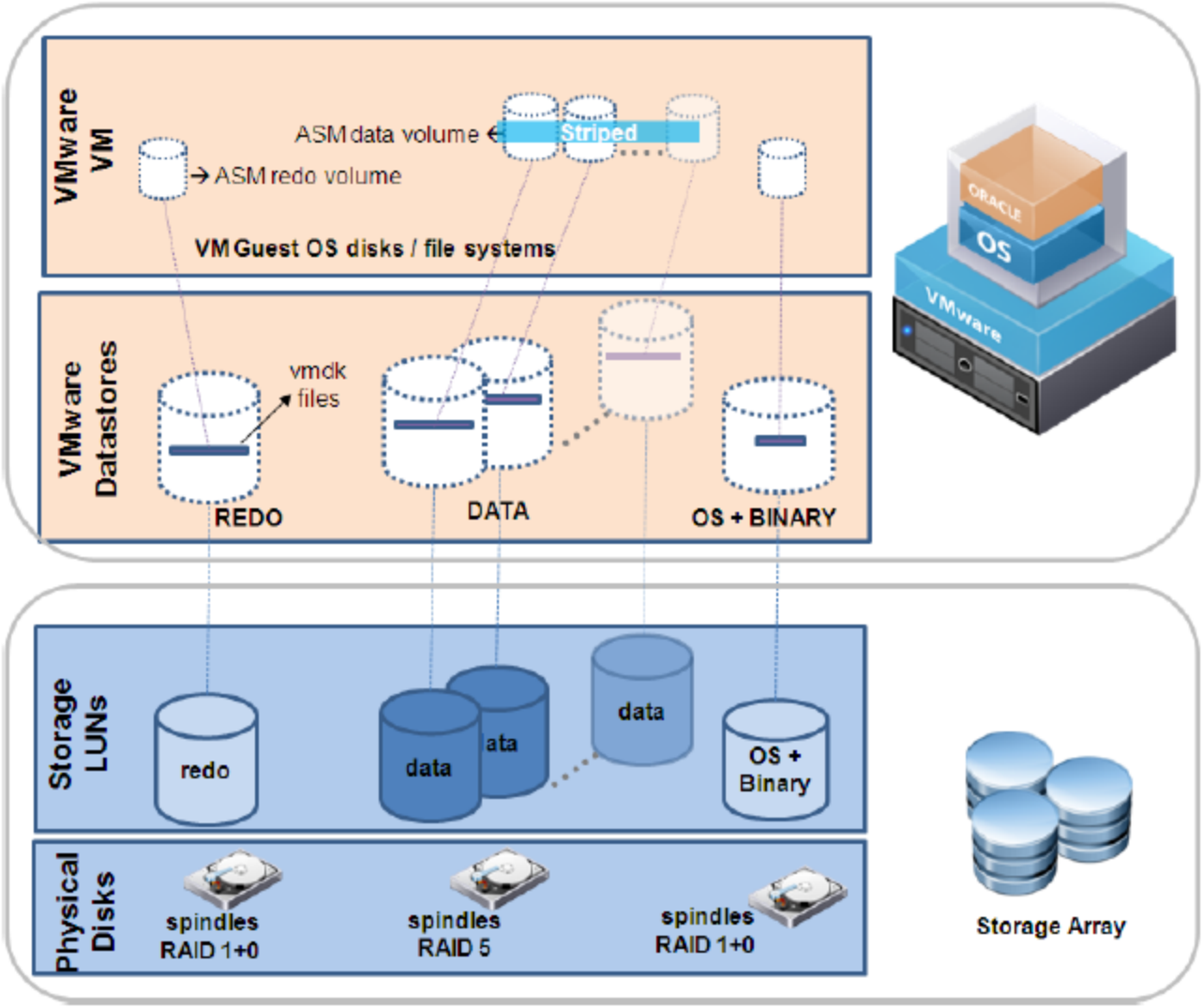
Oracle Databases on VMware – Best Practices Guide

- **5. Storage Guidelines**
  - Recommendation:
    - Use Oracle automatic storage management.
  - Justification:
    - Oracle ASM provides integrated clustered file system and volume management capabilities for managing Oracle database files. ASM simplifies database file creation while delivering near-raw device file system performance.
  - Recommendation:
    - Use your storage vendor's best practices documentation when laying out the Oracle database.
  - Justification:
    - Oracle ASM cannot determine the optimal data placement or LUN selection with respect to the underlying storage infrastructure. For that reason, Oracle ASM is not a substitute for close communication between the storage administrator and the database administrator.

- **5.3.1 Automatic Storage Management**
  - Recommendation:
    - Create ASM disk groups with equal disk types and geometries.
  - Justification:
    - Create multiple ASM disk groups based on I/O characteristics. At a minimum, create two ASM disk groups - one for log files, which are sequential in nature, and another for datafiles, which are random in nature.

Source: Oracle Databases on VMware Best Practices Guide

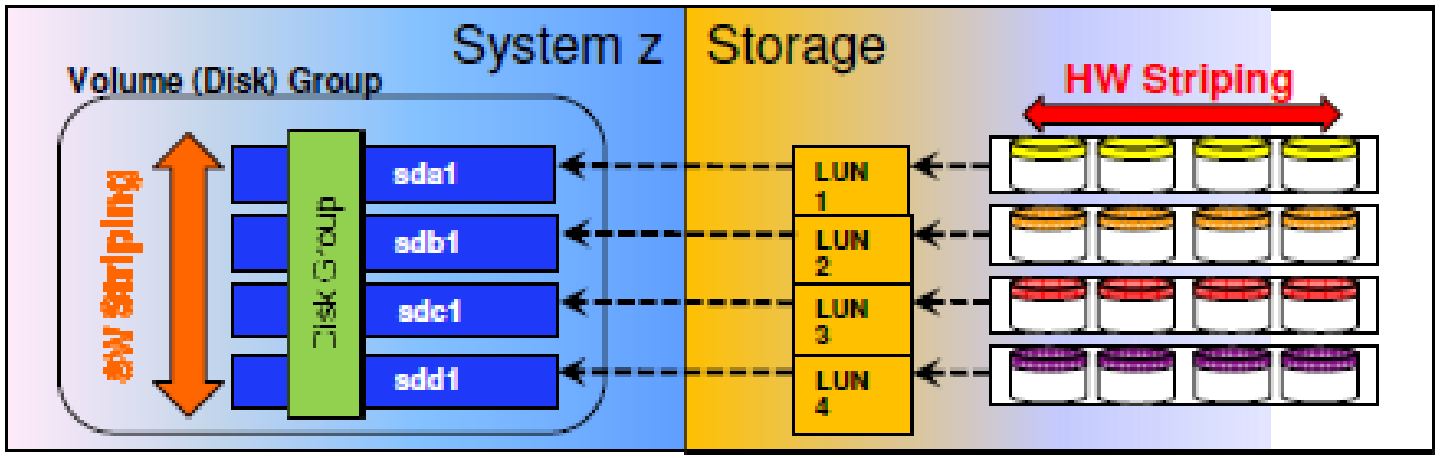
Example Storage Layout of Oracle OLTP Database on VMware



Source: Oracle Databases on VMware Best Practices Guide

Data Striping um I/O Hotspots zu vermeiden

- **Verteilen (Stripe) der Datenobjekte über möglichst viele physische Platten**
  - Minimaler manueller Eingriff
  - Gleichmäßig ausbalanciertes I/O-Verhalten über all verfügbaren Komponenten
  - Im Durchschnitt gute I/O Antwortzeiten und Durchsatz ohne hotspots
  - Implementations-Optionen:
    - Kann mit konventionellem Volume Manager und File-System implementiert werden
    - Oracle DB: ASM kann dies automatisch innerhalb einer Disk Group oder eines File-Systems machen



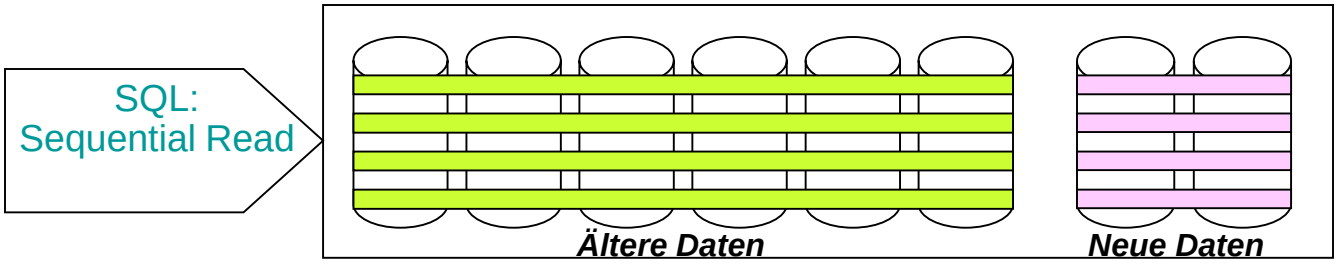
Ein „typisches“ Performance-Problem

Performance-Probleme:

- Lange Laufzeiten der monatlichen Auswertungs- oder Berichtsprogramme
- Anwenderteam sagt, dass keine Änderungen an den Anwendungen gemacht wurden

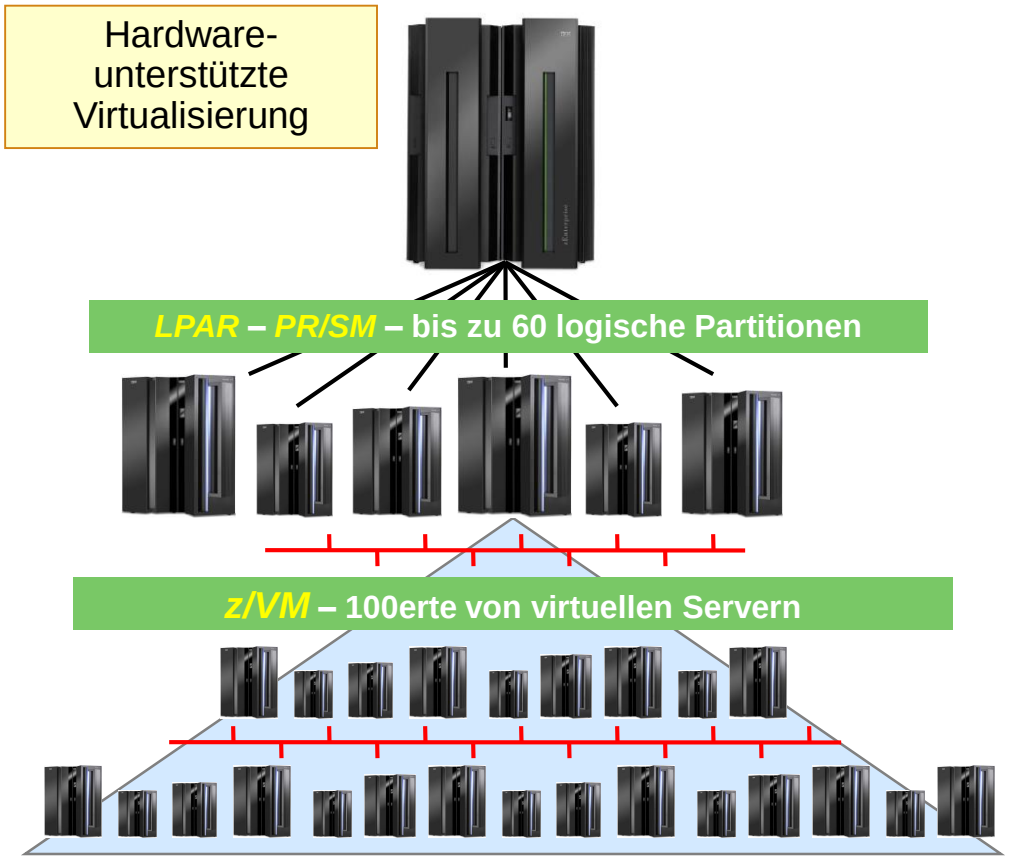
Aber…..

- Starkes Wachstum der Datenbankgröße
  - ca. 12% in 3 Monaten
    - April +45 GB, Mai +27 GB, Juni +32 GB
  - Vergrößerung des zugeordneten Plattenspeichers - 2 weitere Platten (ca. 360 GB)
- Späteres Hinzufügen weiterer Plattenspeicher hat Einfluss auf die Datenverteilung (Striping)
  - Neuere Daten sind nur über 2 Platten verteilt (2 Platten statt 6)
  - Anmerkung: kein ASM



System z – Extreme Virtualisierung

Integrierte und gemeinsam genutzte Architektur (Shared Everything)

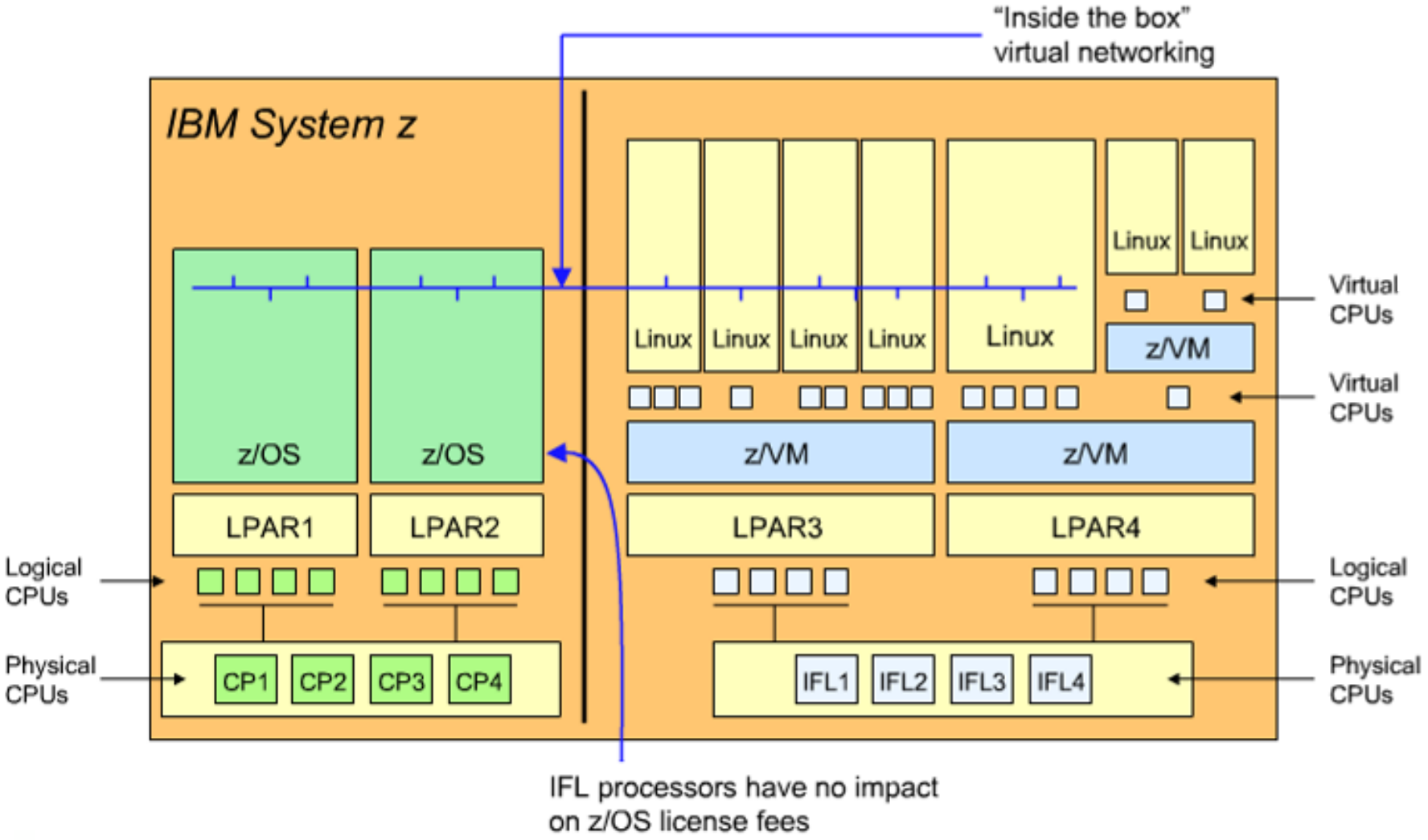


System z

- Inbetriebnahme von virtuellen Servern in Sekunden
- Hohe Granularität der Ressourcennutzung (<1%)
- Aufrüstung der physischen Ressourcen ohne Systemstillstand
- Skalierbarkeit bis zu mehreren 1000 virtuellen Servern
- Mehr mit weniger: mehr virtuelle Server pro Kern, gemeinsame Nutzung der physischen Ressourcen
- Extensives Life-Cycle Management
- HW-unterstützte Isolation, hohe Sicherheit (EAL5 oder EAL4+ zertifiziert)

Extreme Virtualisierung mit z/VM

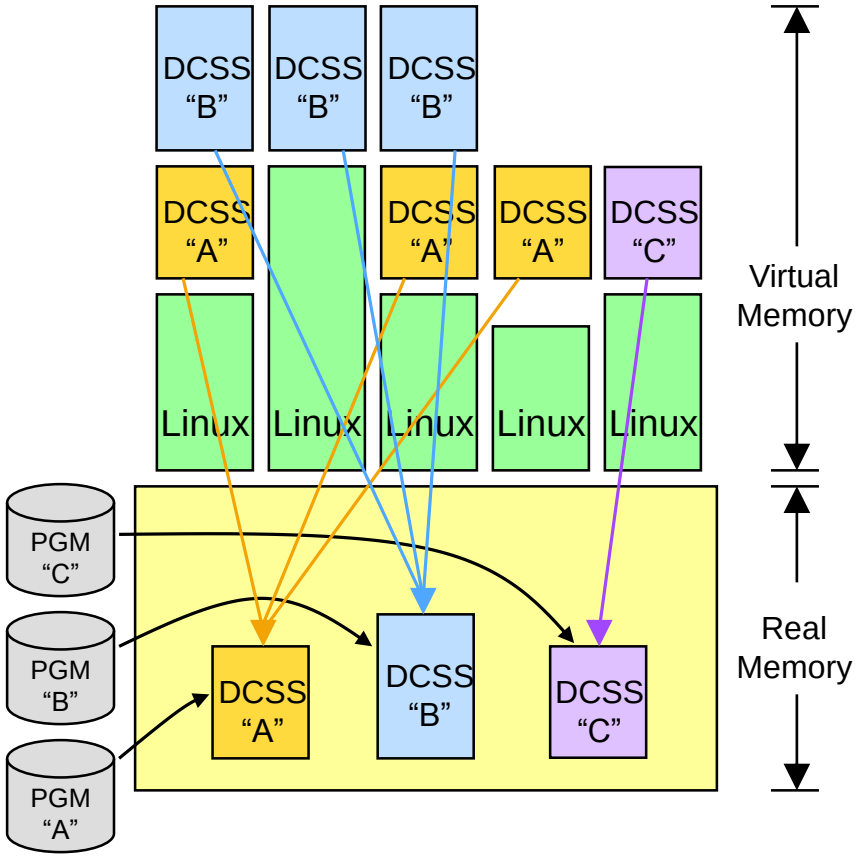
z/VM bietet höchste Skalierbarkeit für eine virtuelle Serverumgebung durch die Kombination von virtuellen und realen Ressourcen für jede virtuelle Maschine



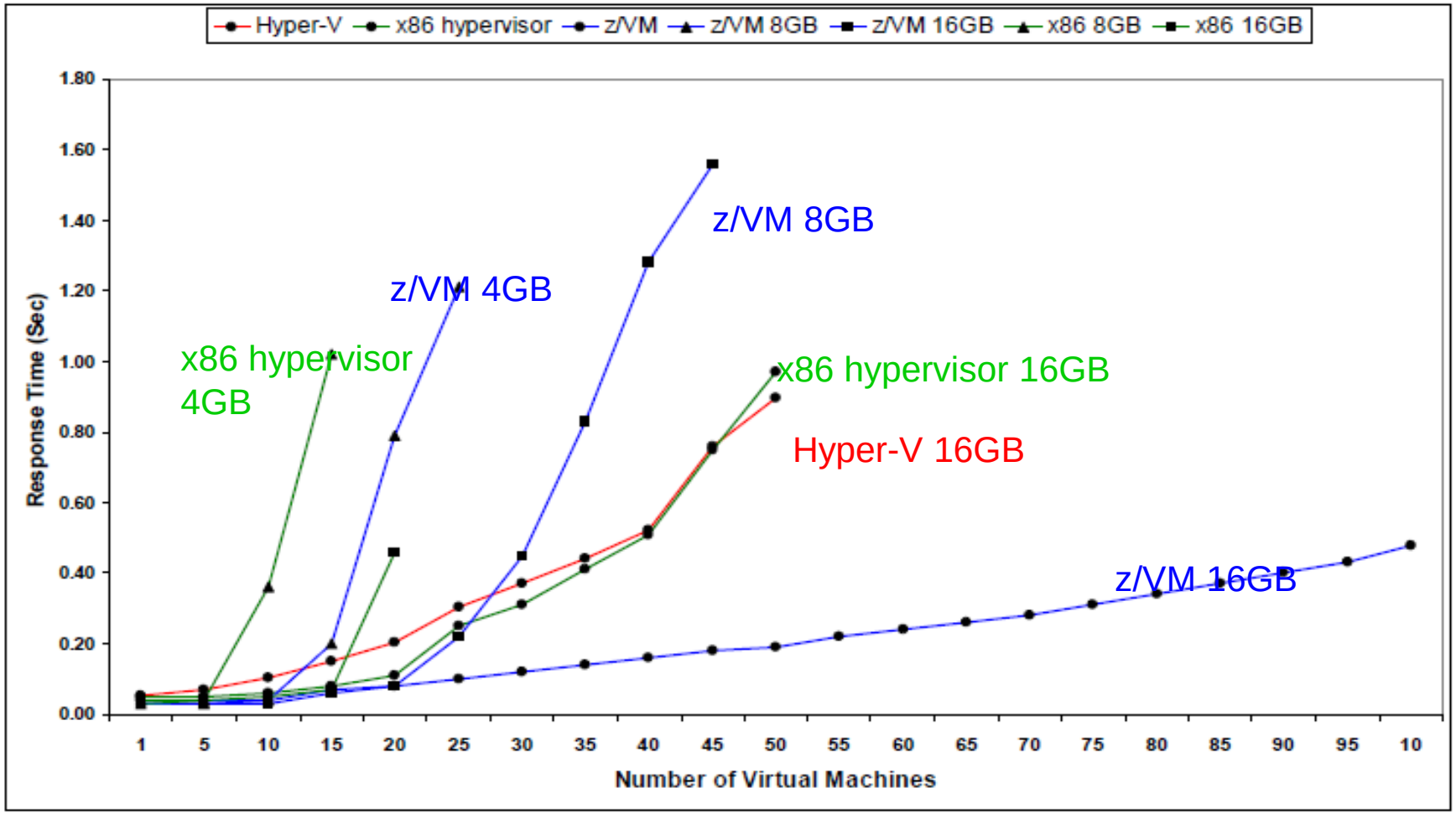
Extreme Virtualisierung mit Linux und z/VM

Linux kann z/VM Discontiguous Saved Segments (DCSS) nutzen

- **DCSS ist Data-in-Memory Technologie**
  - Gemeinsame Nutzung von realen Speicherbereichen durch mehrere virtuelle Gäste
  - Kann den (realen) Speicherbedarf verringern
- **Linux: Nutzung von “shared program executables”**
  - Program executables werden als ein “execute-in-place” File-System gespeichert und dann in ein DCSS geladen
  - Execute-in-place (xip2) File-System
  - Zugriff auf das File-System mit Speichergeschwindigkeit; Executables werden direkt aus dem File-System gestartet (es müssen keine Daten verschoben werden)
  - Vermeidet die Duplizierung von virtuellem Speicher und auf Platten (Disk) gespeicherten Daten
  - Hilft die gesamte Systemperformance und Skalierbarkeit zu verbessern

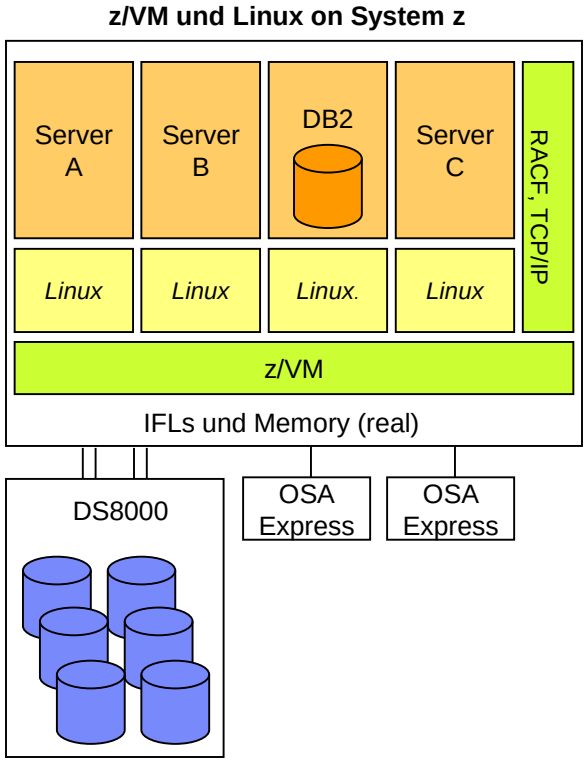


Die Auswirkungen von Speichermangel auf die Antwortzeiten



Generelle Empfehlungen – Linux Speicherbedarf

- Virtuelle Speichergröße für Linux sollte nicht zu groß bemessen werden, weil -
  - Linux nutzt überschüssigen Speicher für I/O-Puffer und File System Cache.
  - In einer virtualisierten Umgebung unter z/VM führen übergroße (oversized) Gäste zu unnötigem Stress auf das VM Paging Subsystem.
  - Realer Speicher ist eine gemeinsam genutzte (shared) Ressource. Caching von Seiten im Linux Gast verringert den Speicher, der für andere Gäste zur Verfügung steht.
  - Größere virtuelle Speicherzuordnungen benötigen mehr Speicher für Address Space Management im Kernel.
- Datenbank-Server haben spezielle Anforderungen
  - Oracle nutzt Pufferspeicher um exzessive I/O Zugriffe zur Platte zu vermeiden.
  - Diese Puffer (SGA, PGA) sollten im Speicher resident bleiben, da sonst Performanceeinbußen drohen.
- Potentiell für alle Datenbanksysteme *vm.swapiness* auf 0 setzen (sysctl.conf)
  - Definiert eine Präferenz den Page Cache zu nutzen, statt Swap durch Linux



Empfehlung für Speicherbedarfsplanung

- Standard Speicherbedarfsplanung (virtueller Linux-Speichergröße) = Summe von:
  - Speicher für Linux Kernel: 512 MB
  - Speicher für Oracle SGA: laut DBA Abschätzung
  - Speicher für Oracle PGA: laut DBA Abschätzung
  - Speicher für Oracle ASM: 256 MB bis 512 MB (falls ASM verwendet wird)
  - Speicher für zusätzliche Agenten, wie OEM, Tivoli etc., nach Bedarf der Anwendung
  - Linux Overhead: 5 % des gesamten Speichers für den Gast

Startgröße = SGA + PGA + 0.5GB für Linux + ASM (falls verwendet)

- Speicher Over-commitment (Verhältnis von virtuellem zu realem Speicher)
  - Kein oder geringes Speicher Over-commitment für kritische Produktions-Datenbanken
  - Test und Development Gäste können von der z/VM Speicher Over-commitment-Funktion profitieren

- Virtual:Real Verhältnis sollte ≤ 3:1 sein (oder das Paging-System muss sehr robust sein)
  - Um Performanceeinschränkungen in kritischen Produktionsanwendungen zu vermeiden, kann es notwendig sein, das Verhältnis auf 1:1 zu bringen
  - 1,5:1 ist ein guter Startpunkt/Kompromiss für viele Anwendungen
  - 3:1 sind häufig akzeptabel in Test- und Entwicklungsumgebungen

Manage CPU und Memory Ressourcen mit Linux cpuplugd daemon

- Die Größenbestimmung von Linux z/VM Gästen kann ein komplexes Unterfangen sein
  - Zu große Gäste kosten zusätzlichen Managementaufwand des Hypervisors
  - Zu kleine Gäste führen häufig zu Performanceproblemen in Spitzenzeiten

➡ Überlass dem System das automatische Management der Ressourcen, basierend auf den Anforderungen des Gastes

- Linux cpuplugd (oder 'hotplug') daemon
  - Kann CPUs und Speicher für den Gast kontrollieren
  - Hinzufügen oder Entfernen von Ressourcen nach vordefinierten Regeln
  - Verfügbar ab SLES 11 SP2 oder 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

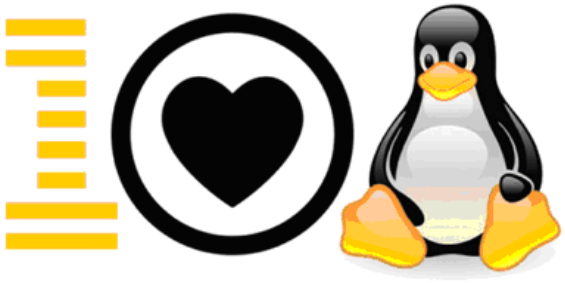
  http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/topic/liaag/l0cpup00.pdf

System z on demand capacity

- On/Off Capacity on Demand (On/Off CoD)
  - Erlaubt es, temporär weitere Prozessoren (IFLs) zu aktivieren um Belastungsspitzen abzufangen.
    - Die Hardwareausstattung muss nicht genutzte Prozessorkapazität haben
    - CoD sieht keine Lizenzvereinbarungen für temporäre Kapazitäten vor
- Capacity Backup Upgrade (CBU)
  - Eine Funktion, die es erlaubt, nicht aktivierte Prozessoren für einen begrenzten Zeitraum zu aktivieren, um Kapazität von einem Betriebsrechner auf einen anderen Rechner im Unternehmen zu verlagern.
  - Typischerweise wird CBU eingesetzt, wenn ein Rechner ausfällt oder aufgrund eines Katastrophenfalls nicht genutzt werden kann.
    - Der Reserverechner kann im Normalbetrieb mit geringer Kapazität laufen und im Katastrophenfall kurzfristig in seiner Kapazität vergrößert werden, was zu erheblichen Kosteneinsparungen führen kann.

Empfehlungen

- Etablieren Sie ein permanentes Monitoring
- Sammeln Sie Systemdaten als Basisbewertung für gute Performance
- Implementieren Sie einen Change-Management-Prozess
- Führen Sie so wenige Änderungen wie möglich zu einer Zeit durch
- Performance ist oft nur so gut, wie das schwächste Glied
- Die Beseitigung eines Flaschenhalses führt zu weiteren
- Erwarten Sie Veränderungen an anderer Stelle, wenn eine Ressource verändert wird



Fragen?

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

Was DBAs über virtualisierte Umgebungen wissen sollten

Server-Virtualisierung ist eine etablierte Methode, um Kosten zu sparen und Ressourcen flexibler nutzen zu können. Dies bedingt aber, dass Hardwareressourcen nun gemeinsam genutzt werden und einzelne Server sich in diese "Gemeinschaft von Servern" einfügen müssen. Daraus ergeben sich neue Herausforderungen an das Kapazitäts- und Performancemanagement. Das Tuning einzelner Anwendungen kann Auswirkungen auf andere Anwendungen in der virtuellen Umgebung haben und ein Systemadministrator muss sicherstellen, dass eine faire Zuteilung von Ressourcen erfolgt, gleichzeitig aber auch Servicelevel-Agreements (SLA) eingehalten werden können.

Dieser gesteigerten Komplexität stehen allerdings auch Vorteile gegenüber: neben Kosteneinsparungen durch effizientere Nutzung der Infrastruktur wird der individuelle Wartungsaufwand für einzelne, jetzt virtuelle, Server geringer, das Betriebsmanagement, wie beispielsweise Backup und Vorsorge für den Katastrophenfall können einheitlicher gestaltet werden.

Für den DBA ist es wichtig zu verstehen, dass Performanceprobleme gelöst werden sollten, anstatt sie mit mehr Ressourcen (mehr Cores, mehr Speicher) herunterzuspielen (was zu erheblichen Mehrkosten führen kann - z.B. Softwarelizenzkosten). In einer virtuellen Umgebung kann "mehr" manchmal sogar "weniger" [Performance] sein.

Der Vortrag betrachtet die grundsätzlichen Unterschiede zwischen physischen Einzelsystemen und virtualisierten Umgebungen, die allgemeingültig sind und für alle virtualisierten Serversysteme, wie VMware, XEN, KVM, HyperV, Oracle VM Server, z/VM, gelten. Praktische Beispiele werden anhand von Oracle DB auf einer hoch-virtualisierten Linux on System z Umgebung mit z/VM erläutert.