ORA-SOLUTIONS.NET

Mastering Oracle Performance, High Availability, Manageability

ORACLE 10g Certified Master

Unix Workshop for DBAs – Part 2: Linux

- München, October 2008

UNIX Workshop für DBAs – 3 Teile

1 Storage

- SAN/NAS
- RAID/SAME/ASM
- MSA/EVA
- Performance, Monitoring

2 Linux

- Booting, Netzwerk (Konfiguration, TCP/IP, Tracing, Bonding), Prozesse (Tracing: strace), I/O Scheduling, Packages, LVM, Raw Devices, VLM, Hugepages, Memory Management, Monitoring, cron, Kernel-Modules, SSH

3 HP-UX

- Memory Management, Kernel Parameters, Mount Options, LVM, Filesystem, Monitoring, Shell Scripting, Networking, APA, Oracle Specifics

Inhalt – Part 2: Linux

1. Enterprise Distributionen
2. Bootvorgang
3. Networking (Bonding, TCP Tracing)
4. Filesysteme / LVM / Raw Devices
5. VLM / Hugepages
6. OS Packages
7. I/O (Scheduling, Direct I/O, Async I/O)
8. Prozesse
9. Memory Management
10. Monitoring
11. cron
12. Kernel-Module
13. SSH
14. Shell-Scripting
15. Logfiles
16. Tips & Tricks

1. Enterprise Distributionen

Kommerziell:

- SUSE (früher SuSE) LINUX ENTERPRISE SERVER
    - Version 10 (Released 07/2006)
- RedHat Enterprise Linux (RHEL) Entry Server (ES) / Advanced Server (AS)
    - Version 5 (Released 04/2007)
- Oracle Enterprise Linux
    - Linux-Support von Oracle

Open-Source:

- CentOS (The Community ENTerprise Operating System)
    - RedHat Clone (100% identisch)

2. Bootvorgang

- Bootloader: GRUB (früher LILO) /boot/grub/grub.conf
- Kernel in /boot Filesystem
- Danach wird / gemountet
- init Prozess
- inittab: definiert default Runlevel
  id:3:initdefault => Runlevel 3 ist default
  si::bootwait:/etc/init.d/boot => erstes Script, das asugeführt wird
  l3:3:wait:/etc/init.d/rc 3 => wenn default runlevel 3, dann wird das ausgeführt.
- /etc/init.d/boot => z.B. Proc Filesystem, auch /etc/init.d/boot.local danach:
- /etc/init.d/rc 3: führt Sxx Scripts in /etc/rc.d/rc3.d aus
- Pflege über chkconfig (-- add, --list, --delete)

```
chkconfig -l sysstat
sysstat                 0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

3. Networking (1) – Konfiguration

- automatisch mit Yast2 (SuSE) oder system-config-network (RedHat)
  manuell unter /etc/sysconfig/network
- NTP: /etc/ntp.conf, XNTP oder NTP daemon mit chkconfig einpflegen.
- DNS: /etc/resolv.conf
- Start/Stop des Netzwerks: /etc/init.d/network start|stop
- /sbin/ifconfig –a zeigt Interfaces
- netstat –nr zeigt Routing Table
- arp –a zeigt ARP Cache (Mapping zwischen MAC Adresse und IP Adresse)
- nslookup servername dns-server # Prüft DNS Auflösung
- ping testet ICMP Connectivity, evtl. blockiert Firewall ICMP
- telnet <hostname> <port> testet TCP/IP Connectivity (STRG+ALT+]) Escape
- netstat –i zeigt Traffic und Errors
- sar –n DEV 1 100: zeigt Traffic
- sar –n EDEV 1 100: zeigt Errors

3. Networking (2) – Bonding

Bei Bonding werden mehrere Physikalische Netzerk-Interfaces zu einem logischen Netzwerk-Interface zusammengefasst um entweder Ausfallsicherheit von einem Switch, Netzwerk-Kabel oder Netzwerk-Karte zu erreichen oder um Load-Balancing durchzuführen.

MODEs:

balance-rr or 0

Round-robin policy: Transmit in a sequential order from the first available slave through the last. This mode provides load balancing and fault tolerance.

active-backup or 1

Active-backup policy: Only one slave in the bond is active. A different slave becomes active if, and only if, the active slave fails. The bond's MAC address is externally visible on only one port (network adapter) to avoid confusing the switch. This mode provides fault tolerance.

3. Networking (3) – Bonding

•Monitoring:

```
myhost:~ # cat /etc/sysconfig/network/ifcfg-bond0
BOOTPROTO='static'
BROADCAST='192.168.0.255'
IPADDR='192.168.0.110'
NETMASK='255.255.255.0'
BONDING_MASTER=yes
BONDING_SLAVE_0='bus-pci-0000:02:0c.0'
BONDING_SLAVE_1='bus-pci-0000:02:0c.1'
BONDING_MODULE_OPTS='miimon=100 mode=1 use_carrier=0 primary=eth0'
```

•Monitoring: /etc/sysconfig/network # cat /proc/net/bonding/bond0

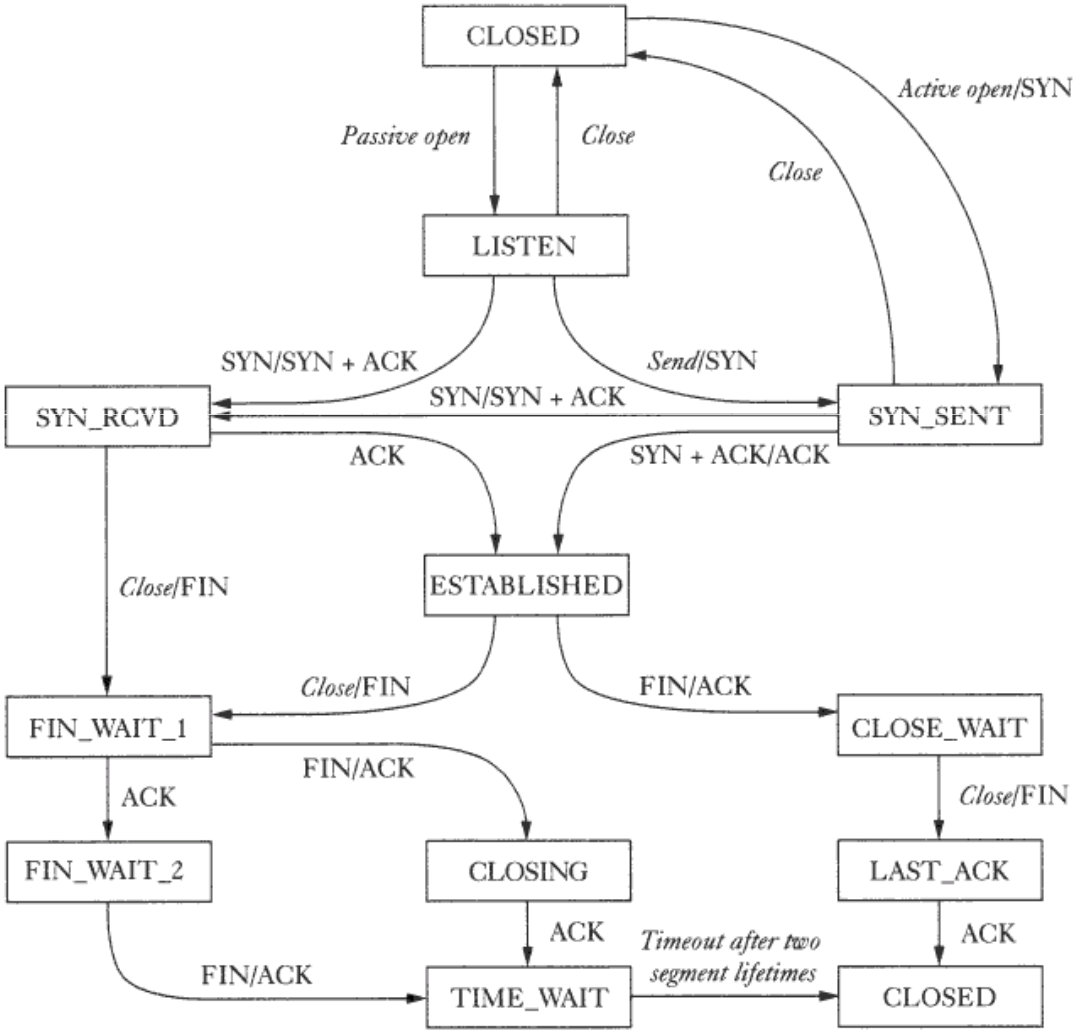
```
Ethernet Channel Bonding Driver: v2.6.5 (November 4, 2005)
Bonding Mode: fault-tolerance (active-backup)
Primary Slave: eth0
Currently Active Slave: eth0
MII Status: up
MII Polling Interval (ms): 100
Link Failure Count: 3
```

•weitere Infos:

http://support.novell.com/techcenter/sdb/en/2004/09/tami_sles9_bonding_setup.html

•/usr/src/linux/Documentation/networking/bonding.txt

3. Networking (4) – TCP/IP

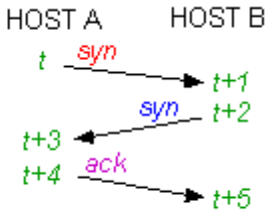


http://www.ssfnet.org/Exchange/tcp/tcpTutorialNotes.html

3. Networking (5)– TCP/IP

- Anzeichen für blockierende Firewall: Socket bleibt im SYN_SENT status
- netstat –an|grep SYN

- Verbindungsaufbau 3-Way-Handshake

Time	Event	DIAGRAM
t	**Host A sends** a TCP **SYN**chronize packet to Host B	
t+1	Host B receives A's **SYN**	
t+2	**Host B sends** it's own **SYN**chronize	
t+3	Host A receives B's **SYN**	
t+4	**Host A sends ACK**nowledge	
t+5	Host B receives **ACK**. ***TCP connection is established.***	

3. Networking (6) – TCP Tracing mit tcpdump

- root Berechtigung notwendig
- promiscous mode (bei Hub)

EXAMPLES

To print all packets arriving at or departing from sundown:

```
tcpdump host sundown
```

To print traffic between helios and either hot or ace:

```
tcpdump host helios and \( hot or ace \)
```

To print all IP packets between ace and any host except helios:

```
tcpdump ip host ace and not helios
```

To print the start and end packets (the SYN and FIN packets) of each TCP conversation that involves a non-local host.

```
tcpdump 'tcp[tcpflags] & (tcp-syn|tcp-fin) != 0 and not src and dst net localnet'
```

Beispiel:

```
muc-ora01:/var/log # tcpdump -n 'tcp[tcpflags] & (tcp-syn|tcp-fin) != 0'
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
16:00:09.730782 IP 10.51.20.110.5783 > 10.51.20.91.6101: S 519544635:519544635(0) win 5840
    <mss 1460,sackOK,timestamp 3538290927 0,nop,wscale 0>
16:00:09.731040 IP 10.51.20.110.5783 > 10.51.20.91.6101: F 519544702:519544702(0) ack
    1805297956 win 5840 <nop,nop,timestamp 3538290927 0>
16:00:11.257932 IP 10.51.20.110.1521 > 10.51.20.78.1385: S 534677494:534677494(0) ack
    3508661581 win 5840 <mss 1460,nop,nop,sackOK>
16:00:13.140970 IP 10.51.20.110.1521 > 10.51.20.78.1385: F 23945:23945(0) ack 6391 win 17520
```

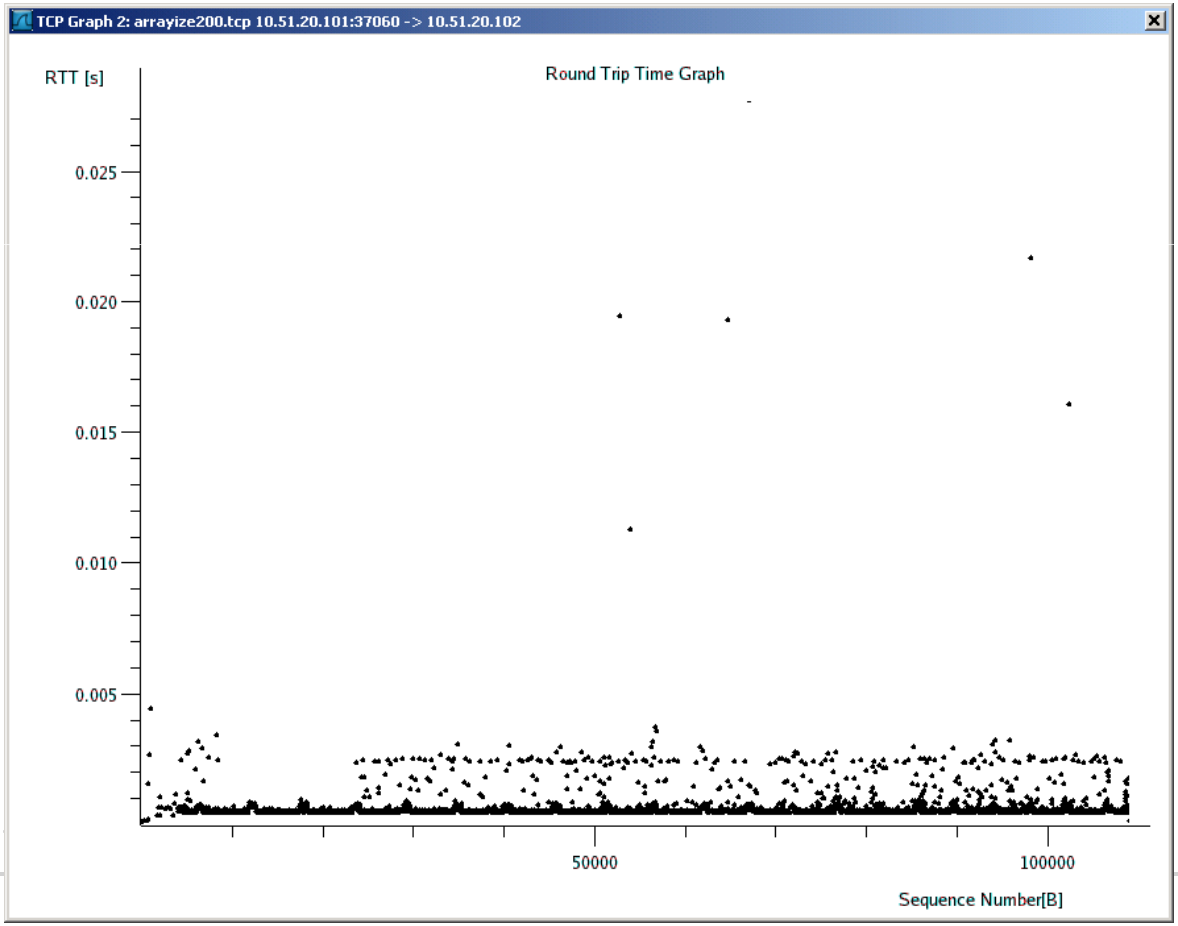
tcpdump -w tcpdump.txt -n host not lpnb125 and host muc-dba01

3. Networking (7) – TCP Tracing mit ethereal/wireshark

- kann selber Capturen, oder Trace von tcpdump einlesen und darstellen
- als root: ethereal oder wireshark

3. Networking (8) – TCP Tracing mit ethereal/wireshark

Round Trip Time Analyse



3. Networking (9) - VIPs (virtuelle IPs)

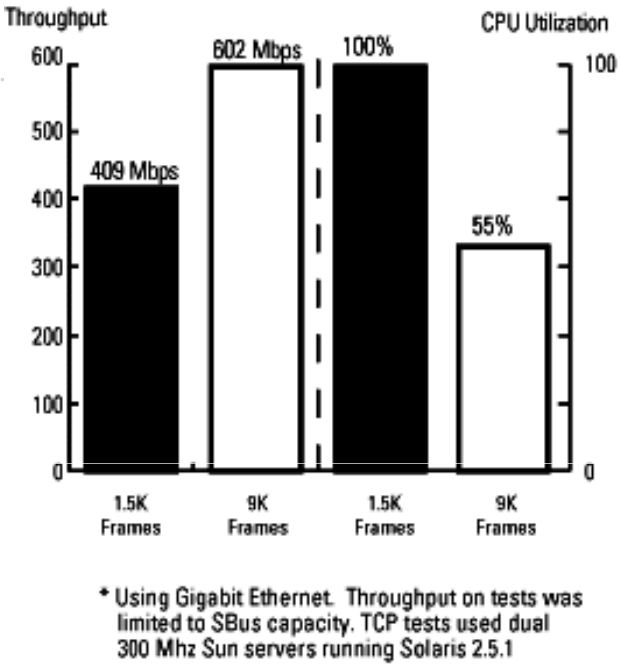
- Wird von Oracle RAC verwendet
- Vergabe von IPs für Applikationen, die im Fehlerfall auf einen anderen Host wechseln können
- Beispiel:
  - `myhost1# ifconfig eth0:1 10.10.10.10 up`

beim Wechsel:

  - `myhost1# ifconfig eth0:1 down`
  - `myhost2# ifconfig eth0:1 10.10.10.10 up`

3. Networking (10) - Jumbo Frames

ORACLE
10g Certified Master



http://sd.wareonearth.com/~phil/jumbo.html

- speziell für Gigabit Ethernet und LAN Traffic
- reduziert CPU Belastung
- erhöht Durchsatz
- z.B. RAC Interconnect
- z.B. myhost1:

```
192.168.0.0     192.168.0.2     U     2 clic1    31744
192.168.0.0     192.168.0.1     U     2 clic0    31744
```

- z.B. hier leider nicht ;-(

```
eth1      Link encap:Ethernet  HWaddr 00:11:43:1E:1D:57
          inet addr:192.168.1.4  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::211:43ff:fe1e:1d57/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2809076 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2788766 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1165676013 (1.0 GiB)  TX bytes:1134239272 (1.0 GiB)
          Interrupt:11
```

3. Networking (11) – SQL*Net over TCP - Arraysize

- Ziel: Reduzierung von Round Trips
- Lösungen:
  - Modifizierung der SDU (Session Data Unit) Size
  - Modifizierung von SQL*Plus Parameter arraysize
- Beispiel:
- # tcpdump **-s 0** -i eth0 -w default.tcp port 1521 # -s 0: snifft komplettes TCP Packet, nicht nur Header
- als Oracle User: sqlplus impuser@DEVHA1 @default.sql

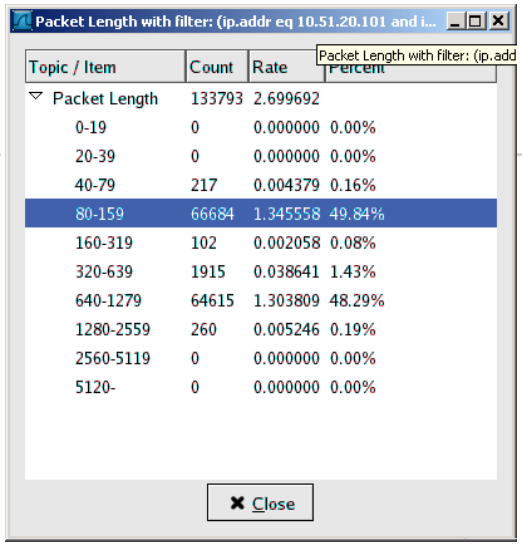
default.sql:

```
ALTER SESSION SET EVENTS '10046 trace name context forever, level 12'
/
set termout off
@sql.txt
set termout on
exit
```

sql.txt:

```
select * from SYS.UNIX_WORKSHOP1 where rownum  <1000000
/
```

3. Networking (12) – SQL*Net over TCP - Arraysize



DEFAULT ARRAYSIZE: 15

```
select * from SYS.UNIX_WORKSHOP1 where rownum  <1000000

call     count       cpu    elapsed       disk      query    current
------- ------  -------- ---------- ---------- ---------- ----------
Parse        1      0.00       0.00          0          0          0
Execute      1      0.00       0.00          0          0          0
Fetch    66668      5.05      11.57       9776      79408          0
------- ------  -------- ---------- ---------- ---------- ----------
total    66670      5.05      11.57       9776      79408          0

Elapsed times include waiting on following events:
  Event waited on                             Times   Max. Wait  Total Waited
  ----------------------------------------   Waited  ----------  ------------
  SQL*Net message to client                   66668        0.00          0.11
  SQL*Net message from client                 66668        0.02         31.87
```

- 999999 rows in 66668 RoundTrips: 999.999 rows á 15 row fetches. (default sqlplus arraysize)
- 10046 Trace:

```
WAIT #1: nam='SQL*Net message from client' ela= 488 driver id=1413697536 #bytes=1 p3=0 obj#=-1
    tim=1151594069326823
WAIT #1: nam='SQL*Net message to client' ela= 3 driver id=1413697536 #bytes=1 p3=0 obj#=-1 tim=1151594069326916
FETCH #1:c=0,e=86,p=0,cr=1,cu=0,mis=0,r=15,dep=0,og=1,tim=1151594069326979
WAIT #1: nam='SQL*Net message from client' ela= 472 driver id=1413697536 #bytes=1 p3=0 obj#=-1
    tim=1151594069327514
WAIT #1: nam='SQL*Net message to client' ela= 4 driver id=1413697536 #bytes=1 p3=0 obj#=-1 tim=1151594069327614
FETCH #1:c=0,e=121,p=0,cr=1,cu=0,mis=0,r=15,dep=0,og=1,tim=1151594069327704
```

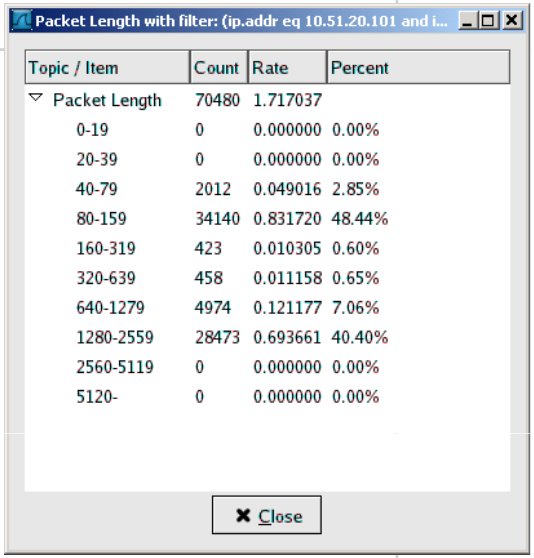
3. Networking (13) – SQL*Net over TCP - Arraysize

ARRAYSIZE 30:

```
select * from SYS.UNIX_WORKSHOP1 where rownum  <1000000

call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.00          0          0          0           0
Execute      1      0.00       0.00          0          0          0           0
Fetch    33335      3.87      10.63      10953      46565          0      999999
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total    33337      3.87      10.63      10953      46565          0      999999

Elapsed times include waiting on following events:
  Event waited on                             Times   Max. Wait  Total Waited
  ----------------------------------------   Waited  ----------  ------------
  SQL*Net message to client                   33335        0.00          0.06
  SQL*Net message from client                 33335        0.02         26.70
  SQL*Net more data to client                   217        0.00          0.00
```



999999 rows in 333.335 RoundTrips: 999.999 rows á 30 row fetches. (sqlplus arraysize 30)

- 10046 Trace:

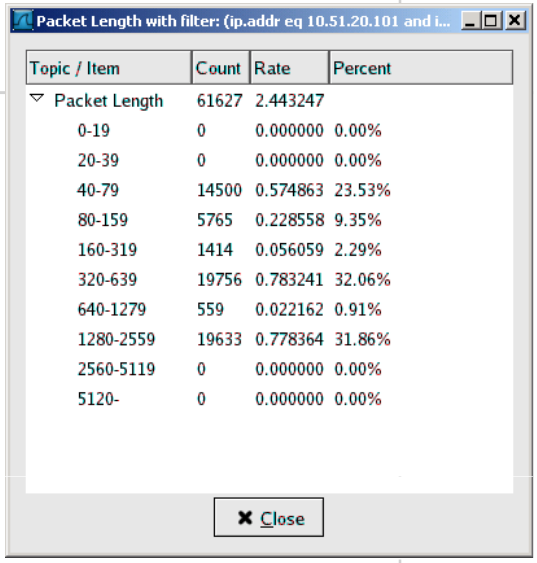
```
WAIT #1: nam='SQL*Net message from client' ela= 734 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151596942143533
WAIT #1: nam='SQL*Net message to client' ela= 1 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151596942143573
FETCH #1:c=0,e=70,p=0,cr=1,cu=0,mis=0,r=30,dep=0,og=1,tim=1151596942143632
WAIT #1: nam='SQL*Net message from client' ela= 826 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151596942144493
WAIT #1: nam='SQL*Net message to client' ela= 1 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151596942144537
FETCH #1:c=0,e=79,p=0,cr=1,cu=0,mis=0,r=30,dep=0,og=1,tim=1151596942144603
```

3. Networking (14) – SQL*Net over TCP - Arraysize

ARRAYSIZE 200:

```
call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.00          0          0          0           0
Execute      1      0.00       0.00          0          0          0           0
Fetch     5001      3.84       4.42         14      18666          0      999999
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total     5003      3.84       4.42         14      18666          0      999999

Elapsed times include waiting on following events:
  Event waited on                             Times   Max. Wait  Total Waited
  ----------------------------------------   Waited  ----------  ------------
  SQL*Net message to client                     5001        0.00          0.00
  SQL*Net message from client                   5001        0.06         19.43
  SQL*Net more data to client                  17728        0.00          0.24
```



* 999999 rows in 5001 RoundTrips: 999.999 rows á 20 row fetches. (sqlplus arraysize 200)

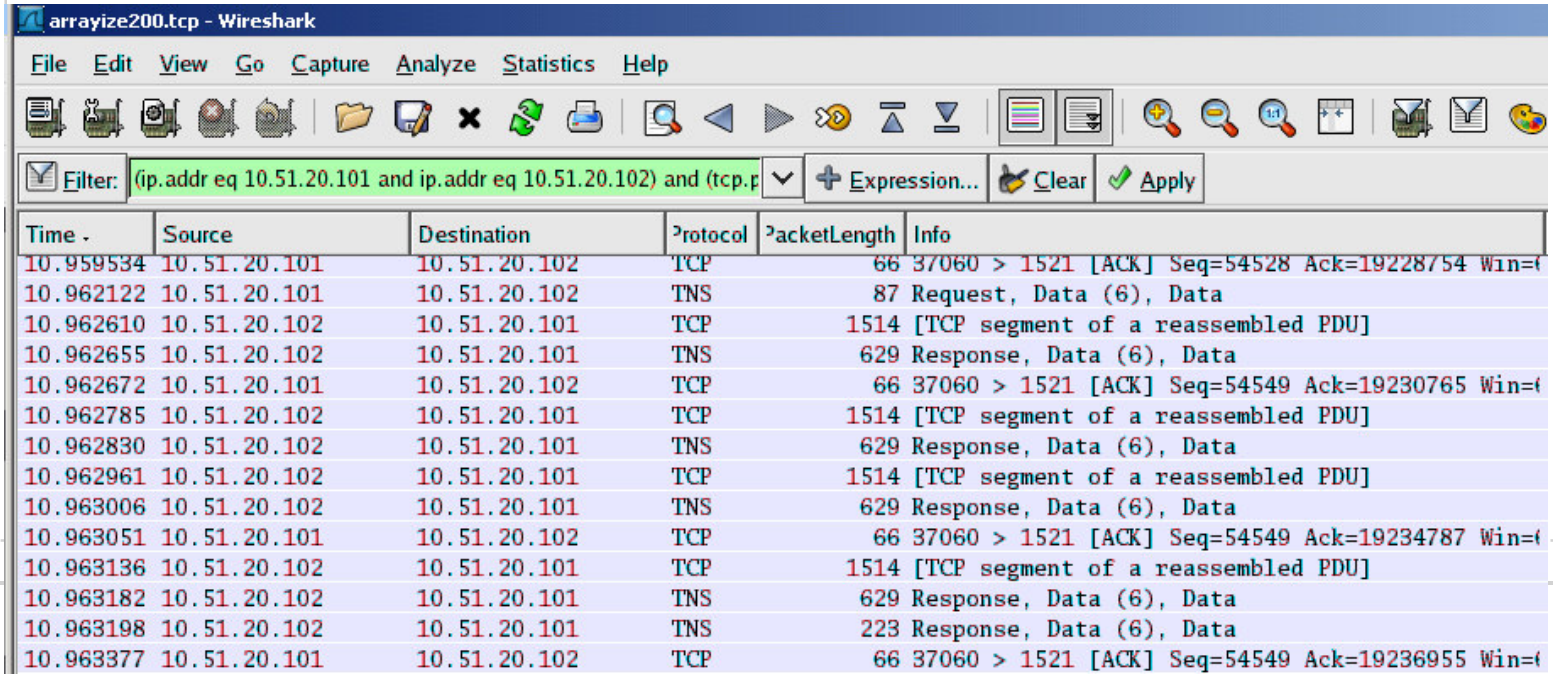
- 10046 Trace:

```
WAIT #1: nam='SQL*Net message from client' ela= 3588 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151597708710758
WAIT #1: nam='SQL*Net message to client' ela= 2 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151597708710807
WAIT #1: nam='SQL*Net more data to client' ela= 12 driver id=1413697536 #bytes=2003 p3=0 obj#=47714 tim=1151597708710931
WAIT #1: nam='SQL*Net more data to client' ela= 11 driver id=1413697536 #bytes=2004 p3=0 obj#=47714 tim=1151597708711057
WAIT #1: nam='SQL*Net more data to client' ela= 11 driver id=1413697536 #bytes=2013 p3=0 obj#=47714 tim=1151597708711181
FETCH #1:c=0,e=453,p=0,cr=4,cu=0,mis=0,r=200,dep=0,og=1,tim=1151597708711247
WAIT #1: nam='SQL*Net message from client' ela= 3458 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151597708714740
WAIT #1: nam='SQL*Net message to client' ela= 1 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151597708714781
WAIT #1: nam='SQL*Net more data to client' ela= 13 driver id=1413697536 #bytes=2007 p3=0 obj#=47714 tim=1151597708714889
WAIT #1: nam='SQL*Net more data to client' ela= 11 driver id=1413697536 #bytes=1996 p3=0 obj#=47714 tim=1151597708715000
WAIT #1: nam='SQL*Net more data to client' ela= 10 driver id=1413697536 #bytes=2000 p3=0 obj#=47714 tim=1151597708715130
FETCH #1:c=0,e=447,p=0,cr=3,cu=0,mis=0,r=200,dep=0,og=1,tim=1151597708715216
```

3. Networking (15) – SQL*Net over TCP - Arraysize

- **Ethernet Framesize: ~1500 Bytes, SDU Size: ~ 2048 Bytes, Oracle Net Message wird in mehrere Ethernet Frames aufgespalten**

```
WAIT #1: nam='SQL*Net message from client' ela= 3588 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151597708710758
WAIT #1: nam='SQL*Net message to client' ela= 2 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151597708710807
WAIT #1: nam='SQL*Net more data to client' ela= 12 driver id=1413697536 #bytes=2003 p3=0 obj#=47714 tim=1151597708710931
WAIT #1: nam='SQL*Net more data to client' ela= 11 driver id=1413697536 #bytes=2004 p3=0 obj#=47714 tim=1151597708711057
WAIT #1: nam='SQL*Net more data to client' ela= 11 driver id=1413697536 #bytes=2013 p3=0 obj#=47714 tim=1151597708711181
FETCH #1:c=0,e=453,p=0,cr=4,cu=0,mis=0,r=200,dep=0,og=1,tim=1151597708711247
WAIT #1: nam='SQL*Net message from client' ela= 3458 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151597708714740
WAIT #1: nam='SQL*Net message to client' ela= 1 driver id=1413697536 #bytes=1 p3=0 obj#=47714 tim=1151597708714781
WAIT #1: nam='SQL*Net more data to client' ela= 13 driver id=1413697536 #bytes=2007 p3=0 obj#=47714 tim=1151597708714889
WAIT #1: nam='SQL*Net more data to client' ela= 11 driver id=1413697536 #bytes=1996 p3=0 obj#=47714 tim=1151597708715000
WAIT #1: nam='SQL*Net more data to client' ela= 10 driver id=1413697536 #bytes=2000 p3=0 obj#=47714 tim=1151597708715130
FETCH #1:c=0,e=447,p=0,cr=3,cu=0,mis=0,r=200,dep=0,og=1,tim=1151597708715216
```



3. Networking (16) – SQL*Net over TCP - SDU

•Modify SDU size when

The data coming back from the server is fragmented into separate packets
You are on a wide area network (WAN) that has long delays
The packet size is consistently the same
Large amounts of data are returned

•Range: 512 bytes to 32767 bytes
•ideales Beispiel: GbE mit Jumbo Frames und hoher Datentransfer
•Empfehlung: Bandwith Delay Product:
•Beispiel: 100 Mbit / 5 ms RTT:

```
100,000,000 bits   1 byte   5 seconds
---------------- x ------ x --------- = 62,500 bytes
1 second           8 bits     1000
```

4. Filesysteme / LVM / Raw Devices (1)

FILESYSTEME:

- MetaLink Note **414673.1**: **SuSE/Novell: Linux, Filesystem & I/O Type Supportability**

- Direct I/O: Umgeht den Filesystem Buffer Cache (http://www.ixora.com.au/tips/avoid_buffered_io.htm)

- Async I/O: verwendet nicht synchrone System Calls Read(), write() sondern asynchrone System Calls. (http://www.ixora.com.au/notes/asynchronous_io.htm)
  - Vorteil: I/Os können gebündelt werden, höhere Parallelisierung
  - Ab Oracle 10.2 kein Relink notwendig. (MetaLink Note **365416.1**: **Asynchronous I/O Does Not Appear to Work In 10gR2 on Linux**)
  - Test ob Async I/O enabled: (DISK_ASYNC_IO, FILESYSTEMIO_OPTIONS=ASYNCH oder SETALL)
    cat /proc/slabinfo |grep kio # enabled, 0 if disabled
    kioctx **270** 270 128 9 9 1 : 252 126
    kiocb **66080** 66080 96 1652 1652 1 : 252 126
    kiobuf **236** 236 64 4 4 1 : 252 126

4. Filesysteme / LVM / Raw Devices (2)

FILESYSTEME

- ext3:
  - Bevorzugt von Oracle
  - Resizing (grow/shrink): http://www.howtoforge.com/linux_resizing_ext3_partitions
- reiserfs:
  - ab SLES10 nicht mehr von Oracle supported
  - Skaliert nicht gut bei > 4 CPUs
- xfs:
  - Stammt von SGI, früher kommerziell
  - Resizing möglich
  - viele Optimierungs-Optionen
  - Oracle does not run certifications on local filesystems (i.e. except for OCFS2, NFS etc.) except ext2/ext3 as it is the common default filesystem for all Linux distributions. So if a problem happens specific to XFS, the Linux vendor should be engaged.

4. Filesysteme / LVM / Raw Devices (3)

LVM – Konfiguration über Command Line (RedHat):

1. Partitionierung mit fdisk

2. pvcreate -d /dev/sdb2 # initialize a disk or partition for use by LVM

3. vgcreate -l 256 -p 256 -s 128k /dev/pv2 /dev/sdb2 # create a volume group
   # -l, --maxlogicalvolumes MaxLogicalVolumes
   # -p, --maxphysicalvolumes MaxPhysicalVolumes
   # -s, --physicalextentsize PhysicalExtentSize[kKmMgGtT]

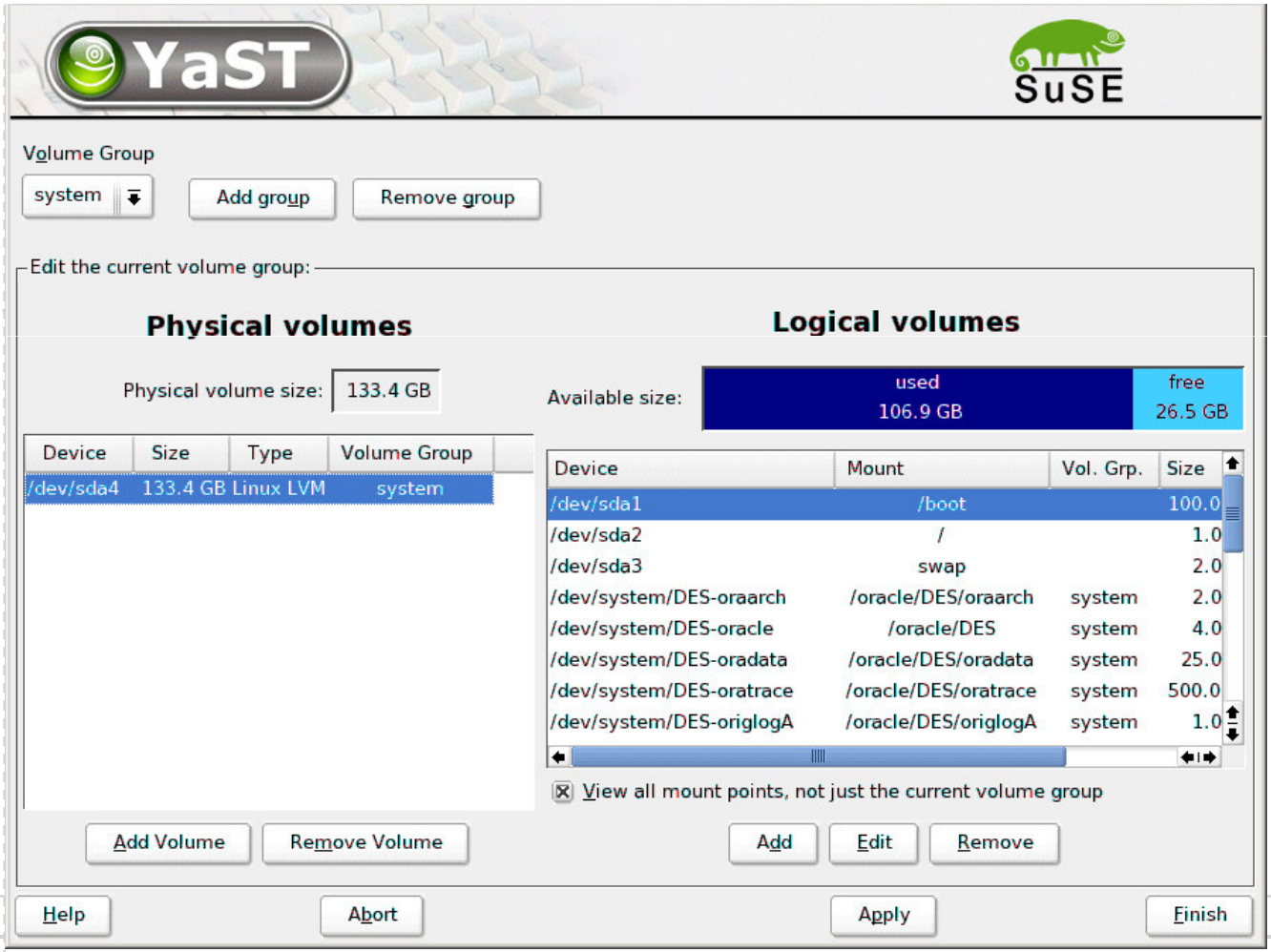
4. # create a logical volume in an existing volume group
   lvcreate -L 10000m /dev/pv2
   # -L, --size

5. /usr/bin/raw /dev/raw/raw21 /dev/pv2/lvol0 # bind a Linux raw character device

6. Anpassen von /etc/rc.local:
   vgscan
   vgchange -a y

4. Filesysteme / LVM / Raw Devices (4)

LVM – Konfiguration über Yast2 (SUSE): System->LVM



4. Filesysteme / LVM / Raw Devices (5)

RAW DEVICES:

- Raw Device muss mind. 1 Oracle Block größer sein als Größe des Datafiles
- kein Auto-extend
- Bei jedem Boot: (/etc/rc.local)
  - Binden mit: raw /dev/raw/raw1 /dev/pv2/lvol0
  - chown oracle:dba /dev/raw/raw1
  - chmod 600 /dev/raw/raw1
- Verwendung: CREATE TABLESPACE TEST datafile '/dev/raw/raw1' size 999M;

5. VLM / Hugepages (1)

VLM: Very Large Memory

- bei x86 nur < 1.7 GB SGA möglich.
- Workaround:
  - init.ora: USE_INDIRECT_DATA_BUFFERS=TRUE
  - init.ora: DB_BLOCK_BUFFERS=xxx # statt DB_CACHE_SIZE
- Dann liegt db_block_buffer in ramfs unter /dev/shm
- MetaLink Note: 317139.1: How to Configure SuSE SLES 9 32-bit for Very Large Memory with ramfs and HugePages

5. VLM / Hugepages (2)

Hugepages:

- normalerweise memory pages 4kb. bei großer sga, viele pages, hoher Verwaltungsaufwand
- Hugepages: bei SUSE x86: 2 MB oder 4MB
- Effizienterer Translation lookaside buffer (TLB)
- vm.nr_hugepages = Anzahl der 2 MB oder 4 MB Pages für SGA
- cat /proc/meminfo: (z.b. muc-dba03)

```
HugePages_Total:    44
HugePages_Free:      0
Hugepagesize:     4096 kB
```

5. VLM / Hugepages (3)

Advantages of HugePages

- **Not swappable:** HugePages are not swappable. Therefore there is no page-in/page-out mechanism overhead.HugePages are universally regarded as pinned.

- **Decreased page table overhead:** Each page table entry can be as large as 64k and if we are trying to handle 50GBof RAM, the pagetable will be approximately 800MB in size which is practically will not fit in 880MB size lowmem (in 2.4 kernels - the page table is not necessarily in lowmem in 2.6 kernels) considering the other uses of lowmem. When 95% of memory is accessed via 256MB hugepages, this can work with a page table of approximately 40MB in total. See also Note 361468.1.

- **Eliminated page table lookup overhead:** Since the pages are not subject to replacement, page table lookups are not required.

- **Faster overall memory performance:** On virtual memory systems each memory operation is actually two abstract memory operations. Since there are less number of pages to work on, the possible bottleneck on page table access is clearly avoided.

6. OS Packages

- RPM: RPM Package Manager
- Installation von Paket: `rpm -ivh paket.rpm`
- Deinstallation von Paket: `rpm -e paket`
- Was ist alles installiert: `rpm -qa |grep <suchbegriff>`
- Zu welchem Paket gehört eine Datei? `rpm -qf <pfad/zur/datei>`
- Bei SUSE über Yast2
- CentOS:
  - yum upgrade (Distributions-Update über Netz)
  - yum install <package-name> (Paket-Installation über Netz)

7. I/O Scheduling

- Linux Kernel I/O Scheduler:
- The noop scheduler is a FIFO queue. Only the I/O merging is provided. Good if your application already sorts the I/O.
- The deadline scheduler uses an round-robin algorithm to minimize the latency for any I/O request. It implements merging and sorting plus a deadline mechanism to avoid starvation. It prefers reads above writes
- The cfq is the default for SLES10 (and SLES9). It uses a round-robin trying to be fair dividing the available I/O bandwith amongst all I/O requests.
- It implements merging and sorting.
- SLES9: static kernel boot parameter elevator=[name of the scheduler]
- SLES10: Dynamic: echo deadline > /sys/block/sdb/queue/scheduler3
- http://www.nextre.it/oracledocs/ioscheduler_01.html

8. Prozesse (1)

- **Wer ist eingeloggt?**

```
myhost1:~ # w
     13:08:23 up 36 days,  4:13,  1 user,  load average: 0.10, 0.04, 0.00
     USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
     root     pts/0     11:15    0.00s  0.03s  0.00s w
myhost1:~ # who
     root     pts/0        May 29 11:15 (10.51.22.23)
```

- **Wann war wer eingeloggt?**

```
myhost1:~ # last
     root     pts/0        192.168.1.10     Tue May 29 11:15   still logged in
     root     pts/0        desktop1         Fri May 25 16:36 - 17:12  (00:35)
     root     pts/0        desktop1         Thu May 24 15:00 - 18:16  (03:16)
     oracle   pts/0        desktop1         Wed May 23 16:28 - 16:57  (00:29)
     root     pts/1        desktop1         Wed May 23 13:29 - 18:11  (04:41)
     oracle   pts/0        desktop1         Wed May 23 13:22 - 16:09  (02:47)
```

- **Was führt dieser User gerade aus?**

```
myhost1:~ # ps axuww|grep pts/0
     root     21356  0.0  0.2   7920  2556 ?        Ss   11:15   0:00 sshd: root@pts/0,pts/1
     root     21358  0.0  0.1   2892  1744 pts/0    Ss   11:15   0:00 -bash
     root     26640  0.1  0.1   1968  1076 pts/0    S+   13:11   0:00 top
```

- **Wer ist das?**

```
     $ finger mdecker
•    Login name: mdecker                    In real life: Martin Decker
•    Directory: /root/home/mdecker          Shell: /sbin/sh
•    Last login Wed May 30 10:57 on pts/6
```

8. Prozesse (2)

- Abhängigkeiten: Jeder Prozess hat einen Parent Prozess (PPID)
- pstree -a -G -p

```
├─mingetty,7112 tty5
├─mingetty,7113 tty6
├─ntpd,5369 -p /var/lib/ntp/var/run/ntp/ntpd.pid -u ntp -i /var/lib/ntp
├─opmn,32658 -d
│   └─opmn,32660 -d
│       └─httpd,32679 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,32684 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,32686 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,32687 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,32688 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,32689 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,32690 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,32691 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,32708 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,32729 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,515 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,6186 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,28671 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,28674 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,28675 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,28692 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,28695 -d /oracle/http10gR2/Apache/Apache -U 674627591
│           ├─httpd,28696 -d /oracle/http10gR2/Apache/Apache -U 674627591
```

8. Prozesse (3)

- ps auxww (BSD) oder ps –ef (System V)
- Columns:
  - USER, PID, %CPU, %MEM
  - VSZ (Virtual Size), RSS (Resident Size)
  - TTY (Terminal)
  - STAT (Status, z.B. Running, etc.),
  - START (Start-Zeitpunkt),
  - TIME (verbrauchte CPU Minuten)
  - COMMAND

Beispiel:

```
oradb   11096  0.0  0.9 1238520 38208 ?       Ss    2006   7:14
   ora_dbw0_MYDB
```

8. Prozesse (4) – LSOF (list open files)

- **Welche Files hat Prozess gerade geöffnet (Files und Netzwerk-Sockets)?**

```
myhost1:~ # lsof -nPp 3123
COMMAND  PID   USER   FD   TYPE     DEVICE     SIZE   NODE NAME
tnslsnr 3123 oracle  cwd    DIR        3,4      232  72148 /oracle/MYDB/10.2.0/network/admin
tnslsnr 3123 oracle  rtd    DIR        3,4      512      2 /
tnslsnr 3123 oracle  txt    REG        3,4   385622  81582 /oracle/MYDB/10.2.0/bin/tnslsnr
tnslsnr 3123 oracle  mem    REG        3,4   107969    783 /lib/ld-2.3.3.so
tnslsnr 3123 oracle  mem    REG        3,4 18776149  75449 /oracle/MYDB/10.2.0/lib/libclntsh.so.10.1
tnslsnr 3123 oracle  mem    REG        3,4  5486009  81779 /oracle/MYDB/10.2.0/lib/libnnz10.so
tnslsnr 3123 oracle  mem    REG        3,4    12498  52393 /lib/libdl.so.2
tnslsnr 3123 oracle  mem    REG        3,4   175353  52402 /lib/tls/libm.so.6
tnslsnr 3123 oracle  mem    REG        3,4    88731  52403 /lib/tls/libpthread.so.0
tnslsnr 3123 oracle  mem    REG        3,4    89178    795 /lib/libnsl.so.1
tnslsnr 3123 oracle  mem    REG        3,4    55728  78444 /oracle/MYDB/10.2.0/lib/libons.so
tnslsnr 3123 oracle  mem    REG        3,4  1375249  52401 /lib/tls/libc.so.6
tnslsnr 3123 oracle  mem    REG        3,4     8105  76584 /oracle/MYDB/10.2.0/lib/libskgxn2.so
tnslsnr 3123 oracle  mem    REG        3,4    41988  52397 /lib/libnss_files.so.2
tnslsnr 3123 oracle  mem    REG        3,4   745445  81198 /oracle/MYDB/10.2.0/lib/libocrutl10.so
tnslsnr 3123 oracle  mem    REG        3,4   893196  81196 /oracle/MYDB/10.2.0/lib/libocr10.so
tnslsnr 3123 oracle  mem    REG        3,4  1235376  81197 /oracle/MYDB/10.2.0/lib/libocrb10.so
tnslsnr 3123 oracle  mem    REG        3,4  2398392  81231 /oracle/MYDB/10.2.0/lib/libhasgen10.so
tnslsnr 3123 oracle  mem    REG        3,4    71577  81232 /oracle/MYDB/10.2.0/lib/libclsra10.so
tnslsnr 3123 oracle  mem    REG        3,4    32114  52395 /lib/libnss_compat.so.2
tnslsnr 3123 oracle  mem    REG        3,4    40530    800 /lib/libnss_nis.so.2
tnslsnr 3123 oracle    0u   CHR        1,3          22064 /dev/null
tnslsnr 3123 oracle    1u   CHR        1,3          22064 /dev/null
tnslsnr 3123 oracle    2u   CHR        1,3          22064 /dev/null
tnslsnr 3123 oracle    3w   REG        3,4 29322369  61904 /oracle/MYDB/10.2.0/network/log/listener_MYDB.log
tnslsnr 3123 oracle    4r  FIFO        0,7         833935 pipe
tnslsnr 3123 oracle    5r   REG        3,4    11776  81569 /oracle/MYDB/10.2.0/network/mesg/nlus.msb
tnslsnr 3123 oracle    6r   REG        3,4    47104  76065 /oracle/MYDB/10.2.0/network/mesg/tnsus.msb
tnslsnr 3123 oracle    7w  FIFO        0,7         833936 pipe
tnslsnr 3123 oracle    8u  IPv4     833939            TCP *:1521 (LISTEN)
tnslsnr 3123 oracle    9u  unix 0xf5487280         833940 /var/tmp/.oracle/s#3123.1
tnslsnr 3123 oracle   10u  unix 0xf5487a80         833942 /var/tmp/.oracle/sMYDB
tnslsnr 3123 oracle   11u  unix 0xf654c180         833944 /var/tmp/.oracle/sEXTPROC
tnslsnr 3123 oracle   12u  unix 0xf654c380         833946 /var/tmp/.oracle/s#3123.2
tnslsnr 3123 oracle   13u  IPv4     834053            TCP 10.51.20.105:1521->10.51.20.105:2561 (ESTABLISHED)
tnslsnr 3123 oracle   14u  IPv4    1560027            TCP 10.51.20.105:1521->10.51.20.105:36789 (ESTABLISHED)
```

8. Prozesse (5) – LSOF (list open files)

- **Zu welchem Prozess gehört dieser Port?**

```
netstat -an | grep EST

tcp        0      0 10.51.20.105:1521       10.51.20.104:44132      ESTABLISHED

tcp        0      0 10.51.20.105:1521       10.51.20.104:36978      ESTABLISHED

tcp        0      0 10.51.20.105:36789      10.51.20.105:1521       ESTABLISHED

tcp        0      0 10.51.20.105:1521       10.51.20.105:36789      ESTABLISHED


myhost1:~ # lsof -Pi TCP:36789

COMMAND   PID   USER   FD   TYPE  DEVICE SIZE NODE NAME

tnslsnr 3123 oracle   14u  IPv4 1560027       TCP 192.168.1.105:1521->192.168.1.105:36789
    (ESTABLISHED)

oracle  6927 oracle   15u  IPv4 1560026       TCP 192.168.1.105:36789->192.168.1.105:1521
    (ESTABLISHED)
```

8. Prozesse (6) – tracing

- Welche System-Calls führt ein Prozess aus? (=SYS CPU%, wenn Prozess nur in user CPU% arbeitet, dann bekommt man keine Tracing Lines)

```
strace -fp
```

-c **Count time, calls, and errors for each system call and report a summary on program exit. On Linux, this attempts to show system time (CPU time spent running in the kernel) independent of wall clock time.**

-f Trace child processes as they are created by currently traced processes as a result of the fork(2) system call.

-t Prefix each line of the trace with the time of day.

-tt If given twice, the time printed will include the microseconds.

-v Print unabbreviated versions of environment, stat, termios, etc. calls. These structures are very common in calls and so the default behavior displays a reasonable subset of structure members. Use this option to get all of the gory details.

-T Show the time spent in system calls. This records the time difference between the beginning and the end of each system call.

-p pid Attach to the process with the process ID pid and begin tracing.

8. Prozesse (7) – tracing

•Beispiel: Tracing eines Oracle-Server Prozesses, der eine Tabelle liest.

•myhost1: # strace -ttTvfp <pid>

•PID, Time, system call(Parameter) = Rückgabewert

•mit "man 2 system call" kann man Informationen bekommen (RPM Package man-pages)

Beispiele:

```
18:03:04.303324 open("/oracle/MYDB/oradata/users01.dbf",
O_RDWR|O_SYNC|O_DIRECT|O_LARGEFILE) = 14 <0.000022>
8427  18:03:04.303500 io_submit(0x48ea3000, 0x1, 0xbfff0890) = 1 <0.000094>
8427  18:03:04.303652 io_getevents(0x48ea3000, 0x1, 0x1, 0xbfff0868, 0xbfff0888) = 1
<0.011822>
8427  18:03:04.333409 pread(14,
"\6\242\0\0\204&\0\1\375\237\352\4\0\0\2\4=\374\0\0\1\0"..., 8192, 80773120) = 8192
<0.006976>
8427  18:03:41.693191 pread(14,
"\6\242\0\0\213&\0\1\375\237\352\4\0\0\2\4^\255\0\0\1\0"..., 1032192, 80830464) = 1032192
<0.038854>
```

9. Memory Management (1)

- Aggresives Caching
- Swap wird nur verwendet (reserviert), Programm nicht in RAM passt.
- Hugepages
- vmstat:

- cat /proc/meminfo:

```
MemTotal:      1025040 kB
MemFree:        134840 kB
Buffers:         38172 kB
Cached:         219728 kB
SwapCached:      21860 kB
Active:         606712 kB
Inactive:       173948 kB
…
HugePages_Total:     0
HugePages_Free:      0
Hugepagesize:     2048 kB
```

```
myhost1:/etc/sysconfig/network # vmstat -a 1
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free  inact active   si   so    bi    bo   in    cs us sy id wa
 0  0    344  34036 549512 396116    0    0     3   109  172   218  2  1 98  0
 0  0    344  34036 549512 396116    0    0     0     0 1022   157  0  0 100  0
 0  0    344  34036 549512 396116    0    0     0     0 1026   171  0  0 100  0
 0  0    344  34036 549512 396116    0    0     0     0 1023   160  0  0 100  0
 0  0    344  34036 549512 396124    0    0     0     0 1039   161  1  0 99  0
 0  0    344  34036 549500 396136    0    0     0    52 1031   167  0  0 100  0
```

9. Memory Management (2) - vmstat

```
FIELD DESCRIPTION FOR VM MODE
   Procs
       r: The number of processes waiting for run time.
       b: The number of processes in uninterruptible sleep.
   Memory
       swpd: the amount of virtual memory used.
       free: the amount of idle memory.
       buff: the amount of memory used as buffers.
       cache: the amount of memory used as cache.
       inact: the amount of inactive memory. (-a option)
       active: the amount of active memory. (-a option)
   Swap
       si: Amount of memory swapped in from disk (/s).
       so: Amount of memory swapped to disk (/s).
   IO
       bi: Blocks received from a block device (blocks/s).
       bo: Blocks sent to a block device (blocks/s).
   System
       in: The number of interrupts per second, including the clock.
       cs: The number of context switches per second.
   CPU
       These are percentages of total CPU time.
       us: Time spent running non-kernel code. (user time, including nice time)
       sy: Time spent running kernel code. (system time)
       id: Time spent idle. Prior to Linux 2.5.41, this includes IO-wait time.
       wa: Time spent waiting for IO. Prior to Linux 2.5.41, shown as zero.
```

10. Monitoring (1) – CPU Monitoring

- ```
  myhost1:/etc/sysconfig/network # sar -u 1 100
  Linux 2.6.5-7.283-bigsmp (muc-dba01)    03/15/07
  17:53:15          CPU     %user     %nice   %system   %iowait     %idle
  17:53:16          all      0.99      0.00      0.00      0.00     99.01
  17:53:17          all      0.00      0.00      0.00      0.00    100.00
  17:53:18          all      0.00      0.00      1.98      0.00     98.02
  17:53:19          all      0.00      0.00      0.00      0.00    100.00
  ```

- ```
  myhost1:/var # mpstat -P ALL 1 10
  Linux 2.6.5-7.283-smp (muc-dba03)       05/29/07
  14:15:44     CPU    %user   %nice %system %iowait    %irq   %soft   %idle    intr/s
  14:15:45     all     0.00    0.00    0.00    0.00    0.00    0.00  100.00   1025.00
  14:15:45       0     0.00    0.00    0.00    0.00    0.00    0.00  100.00   1025.00
  14:15:45       1     0.00    0.00    0.00    0.00    0.00    0.00  100.00      0.00
  ```

10. Monitoring (2) – I/O Monitoring

```
myhost1:/etc/sysconfig/network # iostat -x
avg-cpu:  %user   %nice    %sys %iowait   %idle
           0.00    0.00    0.00    0.00  100.00

Device:    rrqm/s wrqm/s   r/s   w/s  rsec/s  wsec/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
fd0          0.00   0.00  0.00  0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
hda          0.00   0.00  0.00  0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
hdc          0.00   0.00  0.00  0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00

avg-cpu:  %user   %nice    %sys %iowait   %idle
           0.00    0.00    0.00    0.00  100.00

Device:    rrqm/s wrqm/s   r/s   w/s  rsec/s  wsec/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
fd0          0.00   0.00  0.00  0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
hda          0.00   6.00  0.00  6.00    0.00  130.00     0.00    65.00    21.67     0.00    0.33   0.33   0.20
hdc          0.00   0.00  0.00  0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
```

10. Monitoring (3) – Network Monitoring

```
myhost1:/etc/sysconfig/network # sar -n DEV 1 10  # Netzwerk Durchsatz

17:54:20        IFACE   rxpck/s   txpck/s   rxbyt/s   txbyt/s   rxcmp/s   txcmp/s  rxmcst/s
17:54:21           lo     13.13     13.13    712.12    712.12      0.00      0.00      0.00
17:54:21         eth0     30.30     11.11   2723.23   1522.22      0.00      0.00      0.00
17:54:21         sit0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
17:54:21        IFACE   rxpck/s   txpck/s   rxbyt/s   txbyt/s   rxcmp/s   txcmp/s  rxmcst/s
17:54:22           lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00
17:54:22         eth0     24.00      1.00   1726.00    470.00      0.00      0.00      0.00
17:54:22         sit0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
17:54:22        IFACE   rxpck/s   txpck/s   rxbyt/s   txbyt/s   rxcmp/s   txcmp/s  rxmcst/s
17:54:23           lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00
17:54:23         eth0     25.00      1.00   1882.00    470.00      0.00      0.00      0.00
17:54:23         sit0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
```

```
myhost1:/var # sar -n EDEV 1 10    #   Netzwerk-Fehler (Collisions, Drops, etc.)
Linux 2.6.5-7.283-smp (muc-dba03)       05/29/07

14:22:24        IFACE   rxerr/s   txerr/s    coll/s  rxdrop/s  txdrop/s  txcarr/s  rxfram/s  rxfifo/s
     txfifo/s
14:22:25           lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
     0.00
14:22:25         eth0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
     0.00
14:22:25         sit0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
     0.00

14:22:25        IFACE   rxerr/s   txerr/s    coll/s  rxdrop/s  txdrop/s  txcarr/s  rxfram/s  rxfifo/s
     txfifo/s
14:22:26           lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
     0.00
14:22:26         eth0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
     0.00
14:22:26         sit0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
     0.00
```

11. cron – Unix Scheduler (1)

- cron Daemon (/etc/init.d/cron start|stop)
- Crontabs liegen unter /var/spool/cron/tabs
- Anzeigen/Editieren/Löschen mit crontab –l / crontab –e / crontab –r
- crontab format:

  The time and date fields are:

field	allowed values
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names, see below)
day of week	0-7 (0 or 7 is Sun, or use names)

- Beispiele:

```
# run five minutes after midnight, every day
5 0 * * *       $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 2:15pm on the first of every month -- output mailed to paul
15 14 1 * *     $HOME/bin/monthly
# run at 10 pm on weekdays, annoy Joe
0 22 * * 1-5   mail -s "It's 10pm" joe%Joe,%%Where are your kids?%
23 0-23/2 * * * echo "run 23 minutes after midn, 2am, 4am ..., everyday"
5 4 * * sun     echo "run at 5 after 4 every sunday"
```

12. Kernel-Module

- lsmod: Zeigt momentan geladene Kernel Module
- rmmod: Entfernt geladenes Kernel Modul
- insmod: Lädt Kernel Modul
- modprobe: Lädt / Konfiguriert Kernel Modul:
- Beispiele:
  - [root@muc-dba04 ~]# lsmod |grep hangcheck
  - hangcheck_timer 7897 0
  - [root@muc-dba04 ~]# modprobe hangcheck-timer hangcheck_tick=30 hangcheck_margin=180
  - dmesg|grep Hangcheck

    Hangcheck: starting hangcheck timer 0.9.0 (tick is 30 seconds, margin is 180 seconds).
    Hangcheck: Using monotonic_clock().

13. SSH – Secure Shell

- SSH Daemon (/etc/init.d/sshd start|stop)
- Konfiguration: /etc/ssh/sshd_config
- Security Issues:
  - Protocol 1 ist unsicher, nur Protocol 2 verwenden!
  - #PermitRootLogin yes # erlaubt SSH Connections zu Root
  - #UsePrivilegeSeparation yes # Solange SSH Verbindung nicht aufgebaut ist, läuft sshd Prozess unter User sshd mit Shell /sbin/nologin
- Public Key Authentication: Public Key des Clients Users (z.B. mdecker) muss ins ~/.ssh/authorized_keys File des Zielusers am Server eingetragen werden. Login dann ohne Passwort möglich, relativ sicher
- X11 Forwarding
- Passphrase schützt Private Key (siehe SecureCRT)
- SSH-Agent speichert Passphrase auf Client zwischen, sodaß nicht bei jedem Login Passphrase eingegeben werden muss. (siehe SecureCRT)
- Mit strace kann root das Passwort bei Eintippen mitloggen

14. Shell Scripting (1) – Useful commands

- Finde alle Trace Files die zwischen 8 Uhr und 9 Uhr modifiziert wurden:

```
touch -t 200705290800 timestamp_8Uhr
touch -t 200705291000 timestamp_10Uhr
find /oracle/MYDB/oratrace -type f -name "*.trc" -newer
timestamp_8Uhr \! -newer timestamp_10Uhr
```

- Lösche alle Trace Files, …

```
find / oracle/MYDB/oratrace -type f -name "*.trc" -newer
timestamp_8Uhr \! -newer timestamp_10Uhr | xargs rm
```

- Welche Ports verwenden die Oracle Server Prozesse, die über Listener zugreifen?

```
netstat -an|grep 1521 |grep EST | awk {'print $5'} | cut -d
":" -f 2
```

- Loops:

```
for i in *.trc; do tkprof $i $i.tkprof; done
while [ true ]; do du -ks /oracle/MYDB/oraarch; sleep 60; done
for i in seq 1 10; do echo $i; done
```

- Conditions:

```
if [ $? -ne 0 ]; then echo "Erfolgreich"; fi
if [ $ORACLE_SID = “MYDB" ]; then echo "Oracle Instanz heisst
   MYDB"; fi
```

14. Shell Scripting (2) – Beispiel

```
#!/bin/bash
# Check if MRP Process exists (0=exists)
/usr/bin/pgrep -f ora_mrp0_MYSTB1 >/dev/null
# $? => Return code
MRP_DOWN=$?
if [ $MRP_DOWN -ne 0 ]; # If vergleich auf Number (ne, eq, lt, gt, le, ge)
then
   if [ ! -f /home/oracle/dataguard/send_sms.sema ]; # check if file not exists
   then
      touch /home/oracle/dataguard/send_sms.sema
      # send mail (=> sms)
      tail -1 /oracle/MYSTB1/oratrace/bdump/alert_MYSTB1.log | mailx -r
    martin.decker@ora-solutions.net -s "BRONCO: DataGuard MRP Prozess laeuft nicht"
    martin.decker@ora-solutions.net
      exit 1
   fi
else
   if [ -f /home/oracle/dataguard/send_sms.sema ];
   then
      rm /home/oracle/dataguard/send_sms.sema
   fi
fi
```

14. Shell Scripting (3) – weiteres Beispiel

```
#!/bin/bash
LAG=`sqlplus -s "/ as sysdba" <<EOF
SET ECHO OFF
SET NEWPAGE 0
SET SPACE 0
SET PAGESIZE 0
SET FEEDBACK OFF
SET HEADING OFF
SET TRIMSPOOL ON
SELECT SUM(DECODE(name, 'apply lag', value, 0)) LAG
  from (SELECT name,
               extract(day from p.val) * 86400 +
               extract(hour from p.val) * 3600 +
               extract(minute from p. val) * 60 + extract(second from p.val) value
          from (SELECT name, to_dsinterval(value) val from v\\\$dataguard_stats) p);
exit;
EOF`
if [ $LAG -ge 600 ];
then
   if [ ! -f /home/oracle/dataguard/send_lag_sms.sema ];
   then
      touch /home/oracle/dataguard/send_lag_sms.sema
      tail -1 /oracle/MYDB1/oratrace/bdump/alert_MYDB1.log | mailx -r martin.decker@ora-
    solutions.net -s "BRONCO: DataGuard Apply Lag > 300 Sec ($LAG)" martin.decker@ora-solutions.net
      exit 1
   else
      echo
   fi
else
   if [ -f /home/oracle/dataguard/send_lag_sms.sema ];
   then
      rm /home/oracle/dataguard/send_lag_sms.sema
   fi
fi
```

15. Logfiles

- syslogd
  - /etc/syslog.conf: facility.priority
  - *.info;mail.none;authpriv.none;cron.none /var/log/messages
  - /var/log/messages
- /var/log/cron
- dmesg (Kernel Ring Buffer)
- /var/log/secure (sshd)
- /var/log/maillog
- /var/log/ntp
- /var/log/boot.msg
- /var/log/warn

16. Tips & Tricks (1)

- bash History mit: CTRL+R
- bash Command Completion mit TAB
- DOS2Unix in vi: [ESC], :%s#[CTRL+V][CTRL+M]##
- dos2unix in vielen Files:
  for i in `ls mig_bemas*.txt`; do tr -d '\015' < $i > $i.unix; mv $i.unix $i; done
- Zeilen zählen: wc –l
- 3. Spalte aus Liste extrahieren: awk '{print $3}' <file>
- Usernamen aus /etc/passwd extrahieren: cut –d ":" –f 1
- Remove empty lines from file: grep -v '^$' $i > $i.newfile
- Multiple File Rename: for i in *.txt; do mv -i $i mig_arvato_$i; done
- File Rename: uppercase -> lowercase:
  for i in *.TXT; do mv -i $i `echo $i | tr [A-Z] [a-z]` ; done
- Symbolic Link: ln –s <source> <destination>, wobei Destination der String ist, der neu angelegt wird und source der String ist, auf den verlinkt wird
- Recursives Copy: cp –r
- Vergleich von 2 Files: oder 2 Dirs: diff –r <dir1> <dir2>, diff <file1> <file2>
- Sonderzeichen in ASCII File: od –c <file>
- tail -100f <filename> listet die letzten 100 Zeilen und dann alle neu folgenden
- alternativ: less <filename> , dann SHIFT+f
- Zeichen zeilenweise suche mit "grep", invers suchen mit grep -v

16. Tips & Tricks (2)

- exp als komprimierte Datei.

```
# create a named pipe
mknod exp.pipe p
# read the pipe - output to zip file in the background
gzip < exp.pipe > scott.exp.gz &
# feed the pipe
exp userid=scott/tiger file=exp.pipe ...
```

- imp von komprimierter Datei (ohne unzip)

```
# create a name pipe
mknod imp_pipe p
# read the zip file and output to pipe
gunzip < exp_file.dmp.gz > imp_pipe &
# feed the pipe
imp system/pwd@sid file=imp_pipe
log=imp_pipe.log..............
```

16. Tips & Tricks (3)

- Listener Restart mit minimaler Downtime:
  `lsnrctl stop LISTENER_MYDB && lsnrctl start LISTENER_MYDB`
  (Achtung auf Syntax-Fehler in listener.ora) ;-)

- Prompt, um User, Server und Pfad anzuzeigen: export PS1='\u@\h:\w> '

Fragen?

Martin Decker
www.ora-solutions.net