ORA-SOLUTIONS.NET
Mastering Oracle Performance, High Availability, Manageability

RAC on Unix:

Pre-Production Testing & Troubleshooting

DOAG Konferenz 2009

Martin Decker

Agenda

- Vorstellung
- Projektphasen eines RAC Projektes
  - Konzeptionierung
  - Installation
  - Testphase: Benchmark-, Performance und Stabilitätstests
- RAC Troubleshooting (Performance, Node Evictions, Lost Blocks)
- Tools (OS Watcher, LTOM, HANGFG, procwatcher, RH Hangwatch, HANGFG, IPD)

Wer bin ich?

- unabhängiger Oracle Consultant
- 6 Jahre Erfahrung als DBA in komplexen Umgebungen
- davor 2 Jahre Solaris Sysadmin
- Spezialisierung auf:
  - Performance Management
  - Hochverfügbarkeit (RAC, DataGuard)
  - Manageability (OEM Grid Control)
  - Unix (Linux, Solaris, HP-UX)
- Website & Blog: ora-solutions.net

Who am I

- Zertifizierungen
  - Oracle Database: SQL Certified Expert (09/2009)
  - Red Hat Certified Engineer – RHCE (07/2009)
  - Oracle 11g Certified Professional (07/2008)
  - Oracle 10g Certified Master (05/2008) - http://www.oracle.com/technology/ocm/mdecker.html
  - Oracle 10g Real Application Clusters Administrator Certified Expert (09/2007)
  - Oracle Enterprise Linux Fundamentals (08/2007)
  - Oracle Database 10g Managing Oracle on Linux Certified Expert (07/2007)
  - ...

Umfrage

- RAC im Einsatz?
- Schon einmal **Node Eviction** gehabt?

Projektphasen eines RAC Projekts

Projektphasen

- Kickoff
- Projektteam
- Erwartungshaltung
- Verfügbarkeits-anforderungen
- Hardware-Sizing
- Interconnect Technologie
- Redundanzen
- Backup/Recovery Strategie
- Applikations-zertifizierung
- ...

- HW-Aufbau
- Verkabelung
- OS Installation + Updates
- Firmware Updates
- Driver Updates
- Netzwerk-Bonding
- Storage-Konfiguration
- Storage Multipathing
- Hugepages
- ASMLIB
- CRS/ASM/RDBMS + Patches

- I/O Benchmark
- Network Benchmark
- Oracle Benchmark (Swingbench/Hammerora)
- Testszenarien
- Troubleshooting

Konzeptionierungsphase

- Projektteam
  - Business Owner (definiert SLAs, Verfügbarkeiten, Erwartungen)
  - RAC Datenbank-Administrator
  - System-Administrator
  - Netzwerk-Administrator
  - Storage-Administrator

- Verfügbarkeitsanforderungen
  - Single Instance (Oracle Restart, Oracle RAC One in 11gR2)
  - Real Application Cluster Database
    - Node/Instance Crash einer Instance verursacht kurzzeitigen Freeze (no logins!) bzw. teilweisen Freeze der „surviving Instance“ aufgrund von Remastering von Global Enqueue Resources
    - „Shared Storage“ – Downtime aufgrund von Datafile corruption
  - Oracle Data Guard
  - Maximum Availability Architecture (MAA)
    - Sammlung von Oracle Best Practices, um in jede Technologie-Schicht die benötigte Redundanz zu bringen
    - Oracle RAC + Data Guard

- Hardware Sizing - „Well Balanced" – speziell bei DWH
  - # Nodes
  - # CPU-Cores (Lizenzfrage), Intel Nehalem CPUs
  - # RAM (4 GB pro Core)
  - # HBAs
    - 2 Gb => 150-180 MB/s
    - 4 Gb => 375 MB/s
    - 16 Port Switch => (8*2 Gb) => 1200 MB/s
  - # Netzwerk-Interfaces
    - 1 GbE => 80 MB/s (Latenz 60 µ) (~10-12.000 á 8 KB)
    - Infiniband => 390 MB/s (Latenz 20 µ)
  - # Disks
    - 10,000 rpm FCAL => 90 IOPS / 20-30 MB/s
    - 15,000 rpm FCAL => 130 IOPS / 25-35 MB/s
    - 7,200 rpm SATA => 70 IOPS / 20-30 MB/s

- Storage
  - OLTP: Fokus auf IOPS / DWH: Fokus auf Durchsatz (MB/sek)
  - RAID Level (RAID5 vermeiden)
  - Stripe Width Misalignment (evtl. offset bei fdisk partitionierung - http://www.pythian.com/news/411/aligning-asm-disks-on-linux)
  - Data Placement (Inner Tracks, Outer Tracks)
    - ab 11gR2: Intelligent Data Placement (ASM) auf File-Basis
  - Platzierung von 3. Voting Disk bei System mit 2 unabhängigen Storages
  - Device Persistency

- Storage - HBA Multipathing
  - Storage Connectivity über mehrere HBAs, Switches, Pfade zu selben LUNs
  - Nutzung des richtigen Block Device (ASM / ASMLIB Konfiguration) => ORACLEASM_SCANEXCLUDE="sd"
  - Linux: Device-Mapper, scsi_id, /etc/multipath.conf (MetaLink Note 564580.1)

- Netzwerk (Interconnect)
  - Interconnect - Häufigster Grund für Node Evictions
  - Compatible Switch Port Settings (Bitrate, Duplex, Autonegotiate, MTU)
  - Jumbo Frames (MTU 9000), kein IEEE Standard, teilweise max 8992
  - kein Crossover-Kabel: PowerOff kann zu Node Eviction führen!
  - dedizierte, redundante Interconnect Switches oder dediziertes, non-routed VLAN
  - Bonding / Teaming / Port Aggregation
    - empfohlen active/backup -> keine Switch Config notwendig

- Netzwerk - Bonding
  - Bonding / Teaming / Port Aggregation (802.3ad)
    - active-backup: Keine Switch-Konfiguration notwendig
    - active-active: Switch-Konfiguration und ausgiebiges Testen notwendig
    - Linux - /etc/modprobe.conf:

```
alias bond0 bonding
alias bond1 bonding
options bonding mode=1 miimon=100 # mode=1 -> active-backup
```

ifcfg-bond0:

```
DEVICE=bond0
BOOTPROTO=none
ONBOOT=yes
NETWORK=192.168.0.0
NETMASK=255.255.255.0
IPADDR=192.168.0.101
USERCTL=no
```

ifcfg-eth0:

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
HWADDR=00:1b:03:ad:0c:1f
USERCTL=no
```

ifcfg-eth1:

```
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
HWADDR=00:1b:2a:a4:dc:1f
USERCTL=no
```

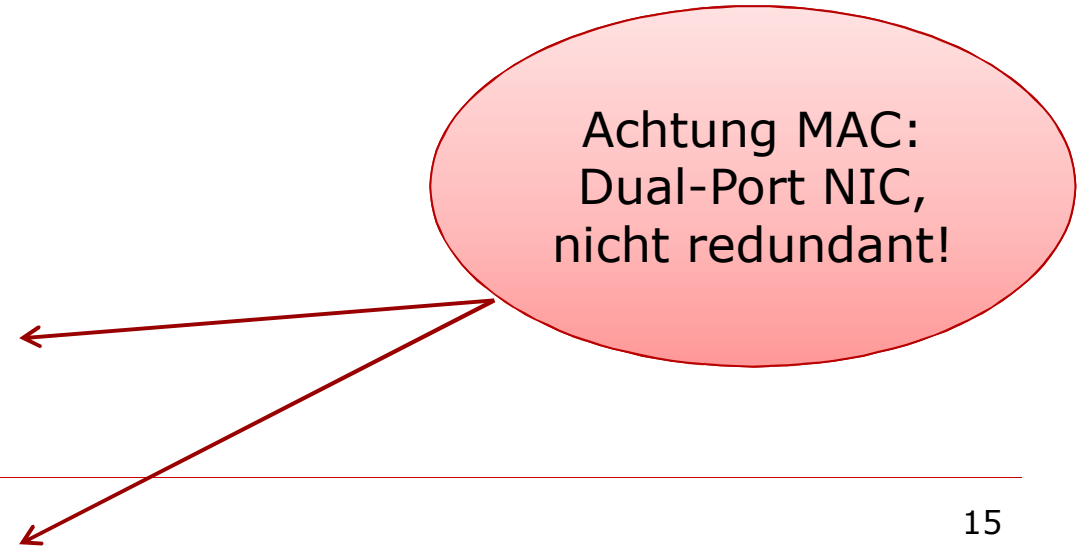
- Netzwerk – Bonding
  - häufiger Fehler

```
cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.2.4 (January 28, 2008)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth0
MII Status: up
MII Polling Interval (ms): 500
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth0
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:14:4f:f4:18:ee

Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:14:4f:f4:18:ef
```



- Applikation
  - Freigabe der Applikationssoftware für RAC
  - Skalierfähig? Scale-Out möglich?
    (Contention: z.B. NOCACHE|ORDER Sequences, etc.)
  - Partitionieren der Application Workload möglich?
  - Dynamic Remastering (ab 10g auf Segment-Basis)
  - DBMS_APPLICATION_INFO
  - Services (Autostart, Failback)
  - Workload Management
  - Fast Connection Failover bei Pools (Code Anpassung bzgl. Behandlung von SQLException)

- Applikation
  - Transparent Application Failover (TAF): kaum praxisrelevant
  - Connect Time Load Balancing (Client/Server-side)
  - Workload Management: Oracle Services und Run-time connection load balancing (Transparent, keine Code-Änderung notwendig)
  - Fast Application Notification (FAN)
  - Fast Connection Failover (FCF): Abfangen von SQLException in Code notwendig.
  - Java Connection Pool:
    - 10g: JDBC Implicit Connection Cache (ICC)
    - 11g: Universal Connection Pool (UCP)
  - Web Applications (z.B. PHP)
    - 11gR1: Database Resident Connection Pool (DRCP)

Konzeptionierungsphase

- CVU – Cluster Verification Utility
  - FixUp-Scripts ab 11gR2
- Known-issues von Patchset 10.2.0.4, teilweise kritisch (Bug 6931689 / Patch #7298531 / Note 739557.1. – filehandle leak)
- NUMA
  - 10.2.0.4 aktiviert per Default,
  - oder Patch 8199533 (MetaLink 769565.1, Bugs 8244734, 7232946, 6689903)
  - deaktivierbar über _db_block_numa/_enable_numa_optimization
- Recommended Patches
  - CRS Patch Bundle | CRS Patch Set Update
  - ab 07/2009 quarterly Patch Set Updates (letztes: vom 20 Okt. 2009)
    - enthält RAC/Generic Patch Bundles
    - PSU 07/2009: 10.2.0.4.1 for RDBMS
    - PSU 10/2009: 10.2.0.4.2 for RDBMS|CRS , 11.1.0.7.1 for RDBMS|CRS, 10.2.0.5.1 for OEM Grid Control
    - einmal PSU immer PSU (kein CPU mehr)
- RAC Starter Kit
  - Generic: MetaLink 810394.1, Linux: MetaLink 811306.1

Installationsphase

- LUN Alignment testen
- 10g Bug: Local Listener Konfiguration
  - per default nicht gesetzt, deshalb <physical ip>:<1521>
  - Problem: Remote Listener redirect auf physical IP
- 10g Bug: ASM Dependency
  - Dependency zwischen ASM Instanz und DB Instanz per default nicht vorhanden.

```
srvctl modify instance -d PRDDB -i PRDDB1 -s +ASM1
srvctl modify instance -d PRDDB -i PRDDB2 -s +ASM2
```

  - MetaLink Note 387217.1
- Diagwait 13
  - Bei Node Evictions darf Node Diag Logs schreiben, bevor er rebooted wird.
  - Für Analyse notwendig
- NTP
  - -x Option in /etc/sysconfig/ntp Options

Installationsphase

- Clusterware 10g Installation
  - VIP-Netmask bei Node Konfiguration in OUI beachten (default 255.255.255.0)
- Linux: Hugepages
  - Reduzieren Größe der Pagetables (PTE)
  - weniger Kernel-Memory
  - weniger % Sys CPU
  - Achtung: nicht kompatibel mit AMM von 11g (MEMORY_TARGET)
- 10g AMM (sga_target)
  - spezifizieren von minimum Werten für db_cache_size und shared_pool_size nötig
  - viele Bugs (MetaLink **567078.1)**

Installationsphase

- Verfügbarkeitsprobleme von „Surviving Instance“ bei Instance Crash / Recovery
- Tune RAC Instance/Crash Recovery
  - recovery_parallelism (Werte zw. 0/1 für serial bis PARALLEL_MAX_SERVERS)
  - Single-Instance: FAST_START_MTTR_TARGET
  - RAC: _FAST_START_INSTANCE_RECOVERY_TARGET:
    - Maximale Dauer (sek.) des (teilweisen) Surviving Instance Freeze bei Instance Crash“ zwischen Beginn Instance Recovery und Rolling Forward
  - PARALLEL_EXECUTION_MESSAGE_SIZE 4k oder 8k

Testphase

- Tests
  - Benchmarks
    - I/O Benchmark
    - Netzwerk-Benchmark
    - Database Benchmark
  - Stabilitätstests (vorher Backup!)

  weiters:
  - Applikations-Performance-Tests mit Single Instance
  - Applikations-Performance-Tests mit RAC
  - Langzeittests (24-48h)
  - Backup/Recovery-Tests

Testphase

- I/O Benchmark - ORION (ORacle IO Numbers)
  - Download: http://www.oracle.com/technology/software/tech/orion/index.html
  - Konfigurationsfile enthält LUNs:

```
t1.lun:
/dev/raw/raw1
/dev/raw/raw2
/dev/raw/raw3
```

  - Aufruf: `./orion -run simple -testname t1 -num_disks 8 -cache_size 128`
  - Resultat: TXT/CSV Files für Excel-Import
  - Run Levels:
    - simple: (size_small 8k, size_large 1M, isoliert small random I/O, danach large random I/O)
    - normal: (gleich wie simple, aber auch Kombination von small/large random I/O)
    - advanced: Volle Kontrolle.
  - Parameter entsprechend nach späterem Produktions-Workload setzen (OLTP/DWH)
  - http://download.oracle.com/otn/utilities_drivers/orion/Orion_Users_Guide.pdf

- I/O Benchmark - ORION (ORacle IO Numbers)

```
# ./orion -run simple -testname orion -num_disks 5 -cache_size 128
ORION: ORacle IO Numbers -- Version 11.2.0.0.1
orion_20091029_0946
Test will take approximately 37 minutes, Larger caches may take longer
# cat orion_20091029_0946_summary.txt
ORION VERSION 11.2.0.0.1
Commandline: -run simple -testname orion -num_disks 5 -cache_size 128
Small IO size: 8 KB
Large IO size: 1024 KB
IO Types: Small Random IOs, Large Random IOs
Simulated Array Type: CONCAT
Write: 0%
Cache Size: 128 MB
Duration for each Data Point: 60 seconds
...
Maximum Large MBPS=88.19 @ Small=0 and Large=10
Maximum Small IOPS=589 @ Small=24 and Large=0
Minimum Small Latency=9.59 @ Small=1 and Large=0
```

Testphase

- iperf – Netzwerk-Benchmarking (http://sourceforge.net/projects/iperf/)
  - Ziel: Throughput von 900 MBit/s bei 1 GbE
  - nützlich u.a. für Jumbo-Frame Testing
  - Client/Server Prinzip:
    - iperf server: iperf -s
    - iperf client: iperf –c <server ip>

Server:

```
[root@rac1 ~]# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:   256 KByte (default)
------------------------------------------------------------
[  4] local 192.168.102.60 port 5001 connected with 192.168.0.50 port 55606
[  4]  0.0-10.0 sec  1.09 GBytes    933 Mbits/sec
```

Client:

```
[root@rac2 src]# iperf -c 192.168.102.60
------------------------------------------------------------
Client connecting to 192.168.102.60, TCP port 5001
TCP window size:   256 KByte (default)
------------------------------------------------------------
[  3] local 192.168.102.50 port 55606 connected with 192.168.0.60 port 5001
[  3]  0.0-10.0 sec  1.09 GBytes    933 Mbits/sec
```

Testphase

Swingbench - Database Benchmark

- Download: http://dominicgiles.com/swingbench.html
- Interessante Metriken: TPM (transactions per minute), CPU Utilization, I/O Operations per second (IOPS), Interconnect Latency/Throughput
- 1 Load Generator CPU lastet 2 DB CPUs aus, für 8 CPU Core DB Host benötigt man 4 Load Generator CPU Cores
- Durchschnittsbelastung < 70%
- 3 verschiedene Frontends: swingbench, minibench, charbench
- Setup:
  - Installation von Swingbench auf Load Generator Host und DB Host
  - Benchmark Konfiguration (Schema)

Testphase

- Swingbench - Database Benchmark
- Steps:
  - Start Coordinator

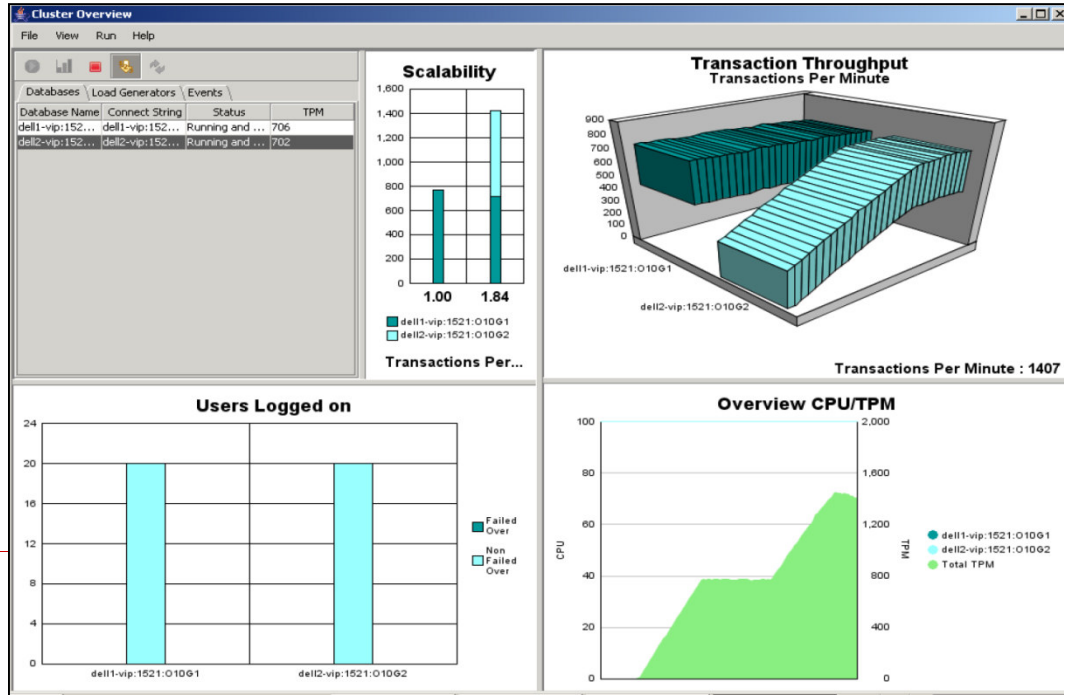
```
$./coordinator -g
```

  - Start Load Generator

```
$./minibench -g group1 -cs //rac1//ORC1 -co localhost &
$./minibench -g group2 -cs //rac2//ORC2 -co localhost &
```

  - Setup clusteroverview.xml
  - Start clusteroverview

```
$./clusteroverview
```



Testphase

- Stabilitätstests
  - Applikationsworkload empfehlenswert
  - zusätzlich: Dummy-Connect Transaction

```
while true; do
sqlplus /nolog <<EOF
CONNECT RACTEST/RACTEST@RACTEST
INSERT INTO RACTEST VALUES (sysdate,
    SYS_CONTEXT('USERENV','INSTANCE_NAME'));
COMMIT;
EOF
sleep 1
done
```

  - Alternativ: Swingbench

Stabilitätstests (1) – RDBMS Prozessfehler

Testszenario	Beschreibung	Erwartetes Resultat
RAC Instanz-Ausfall	*kill -9 <pmon pid>*	Clusterware startet Instance wieder. Testergebnis: 15 sec
Alle RAC Instanzen fallen aus	*kill -9 <pmon pid>* auf allen Nodes	Clusterware startet Instances wieder. Testergebnis: 30 sec
ASM Instanz Ausfall	*kill -9 <pmon pid>* von ASM Instanz	Datenbank-Instance crashed, Clusterware startet ASM und Datenbank Instance wieder.
Alle ASM Instanzen fallen aus	*kill -9 <pmon pid>* von ASM auf allen Nodes	Datenbank-Instances crashen, Clusterware startet ASM und Datenbank Instances wieder.
Listener-Ausfall	*kill -9 <listener pid>*	Clusterware startet Listener wieder. Keine Auswirkung auf Application aufgrund von Client Side Connect Time Failover

Stabilitätstests (2) – Node Failure

Testszenario	Beschreibung	Erwartetes Resultat
Node Ausfall	Poweroff, Power Kabel ziehen, Blade aus Bladecenter herausziehen, Kernel Panic: *echo c > /proc/sysrq-trigger*	Node Eviction nach Ablauf von Misscount, Kurzzeitiges Freeze des überlebenden Nodes, Applikations-Sessions des betroffenen Nodes erhalten Fehlermeldung.
Ausfall mehrerer/aller Nodes	Poweroff, Power Kabel ziehen, Blade aus Bladecenter herausziehen, Kernel Panic: *echo c > /proc/sysrq-trigger*	Restart der Nodes

Stabilitätstests (3) – CRS Failures

Testszenario	Beschreibung	Erwartetes Resultat
crsd.bin/evmd.bin/ocssd.bin Ausfall	*kill -9 <crsd.bin/ocssd.bin/evmd.bin process pid>*	Transparent für crsd.bin/evmd.bin, Node Eviction bei ocssd.bin
Korruption OCR Device	Überschreiben von Teilen des OCR Devices mit Nullen (*dd*)	Keine Auswirkung auf Verfügbarkeit aufgrund von Redundancy, Meldungen im crsd.log
Reparatur OCR Device	***ocrconfig –replace ocr|ocrmirror <device>***	
Korruption Voting Disk	Überschreiben von einer Voting Disk mit Nullen (*dd*)	Keine Auswirkung auf Verfügbarkeit aufgrund von Redundancy solange die Mehrheit der Voting Disks erreichbar ist. (2 von 3)
Reparatur Voting Disk	Bei Oracle 10gR2 nur mit gestoppter Clusterware möglich: ***crsctl delete css votedisk <device> –force*** ***crsctl add css votedisk <device> –force***	

Stabilitätstests (4) - Network

Testszenario	Beschreibung	Erwartetes Resultat
Public NIC Ausfall (ein NIC)	Kabel ziehen bzw. Switch Port deaktivieren	Bonding sollte Failover durchführen und Applikation nichts davon bemerken
Public NIC Ausfall (beide NICs)	Kabel ziehen bzw. Switch Port deaktivieren	Clusterware sollte VIP relocation durchführen, ab 10.2.0.3 sollte Instance+ASM+Listener oben bleiben, kein VIP Relocate bei Reaktivierung der NIC
Private NIC Ausfall (ein NIC)	Kabel ziehen bzw. Switch Port deaktivieren	Bonding sollte Failover durchführen und Applikation nichts davon bemerken
Private NIC Ausfall (beide NICs)	Kabel ziehen bzw. Switch Port deaktivieren	Clusterware führt Node Eviction durch. Bei 2 Node Cluster überlebt Node mit niedrigerer NODE_ID, d.h. Node der zuerst gestartet wurde.
Interconnect Switch Ausfall	Stromkabel von Interconnect Switch ziehen.	Bei redundanten Switches sollte der Ausfall transparent sein. Ansonsten Node Evictions

Stabilitätstests (5) - Storage

Testszenario	Beschreibung	Erwartetes Resultat
Verlust eines Storage-Pfads	FC Kabel ziehen oder FC Switch Port deaktivieren.	Multipathing sollte dazu führen, dass die Applikation weiterhin funktioniert.
Verlust aller Storage-Pfade	FC Kabel ziehen oder FC Switch Port deaktivieren.	Aufgrund fehlendem Disk Heartbeat wird der Cluster Node rebooted (Eviction)
1 Node verliert Storage Zugriff auf Voting Disk	Temporäres Disablen des Zugriffs eines Hosts auf LUN mittels Zoning.	Keine Auswirkung auf Verfügbarkeit aufgrund von Redundancy solange die Mehrheit der Voting Disks erreichbar ist. (2 von 3)
Verlust ASM Disk	Herausziehen einer physikalischen Disk des Storage-Systems bzw. temporäres Disablen des Zugriffs eines Hosts auf LUN mittels Zoning und anschließende Behebung.	Abhängig von Redundancy (external, normal, high)
Entfernen einer Disk aus der Diskgroup	Die ASM Disk soll entfernt und anschließend wieder zur selben Diskgroup hinzugefügt werden.	Rebalancing

Testphase

Stabilitätstests (6) – CPU Starvation

Testszenario	Beschreibung	Erwartetes Resultat
CPU Starvation	Durch Dummy-Prozesse wird das System in eine CPU Überlast-Situation geführt. Dies wurde mittels dd erreicht.	Da hochpriosierte Clusterware / RAC Prozesse in der Realtime Scheduling Klasse laufen, wird das System dadurch nicht instabil.

```
[1]   Running                 dd if=/dev/zero of=/dev/null ibs=99999999 &
. . . . . .
[118]   Running                 dd if=/dev/zero of=/dev/null ibs=99999999 &
[119]-  Running                 dd if=/dev/zero of=/dev/null ibs=99999999 &
[120]+  Running                 dd if=/dev/zero of=/dev/null ibs=99999999 &
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
25  0      0 14524128  58048 1158900    0    0    27     8  142  295  0  3 97  0  0
q34  0       0 14521204  58148 1158908    0    0    40    85 1128 2403  3 97  0  0  0
33  0      0 14518956  58248 1158868    0    0    38    88 1110 2280  3 97  0  0  0
33  0      0 14519152  58332 1158944    0    0    38    69 1120 2278  3 97  0  0  0
50  0      0 14513172  58404 1158952    0    0    38    77 1127 2189  3 97  0  0  0
50  0      0 14512068  58496 1158964    0    0    39    64 1135 2391  3 97  0  0  0
50  0      0 14512072  58576 1158968    0    0    38    60 1126 2307  3 97  0  0  0
49  0      0 14510884  58676 1158980    0    0    39    67 1154 2372  3 97  0  0  0
51  0      0 14510856  58764 1158964    0    0    38    61 1127 2217  3 97  0  0  0
50  0      0 14509760  58864 1158868    0    0    39    86 1138 2396  3 97  0  0  0
50  0      0 14510488  58968 1158768    0    0    39    94 1106 2286  3 97  0  0  0
70  0      0 14502752  59044 1159000    0    0    38    76 1136 2274  3 97  0  0  0

287  0      0 14409860  61064 1159420    0    0    39    88 1111 2260  3 97  0  0  0
289  0      0 14405236  61112 1159412    0    0    39    40 1080 2236  3 97  0  0  0
286  0      0 14412020  61152 1159424    0    0    37    45 1095 2194  3 97  0  0  0
284  0      0 14412832  61192 1159416    0    0    38    42 1089 2135  3 97  0  0  0
286  0      0 14411384  61260 1159404    0    0    39    66 1126 2310  3 97  0  0  0
286  0      0 14411444  61308 1159468    0    0    41    72 1164 2396  3 97  0  0  0
286  0      0 14403504  61372 1159472    0    0    39    60 1132 2292  3 97  0  0  0
284  0      0 14410468  61424 1159476    0    0    38    91 1115 2200  3 97  0  0  0
285  1      0 14408900  61612 1159724    0    0    64    58 1106 2225  3 97  0  0  0
```

Testphase

- Exkurs: „Realtime Prozesse“
  - RDBMS: `_high_priority_processes = LMS*  (default)`
  - CRS: in init.cssd:

```
# Command to make a process run in realtime
        REALTIME_CMD=/usr/bin/chrt
        if [ ! -f $REALTIME_CMD ]; then
          REALTIME_CMD=/bin/chrt
        fi
        if [ -x $REALTIME_CMD ]; then
          REALTIME_PID="$REALTIME_CMD -r -p 99"
        fi
        $REALTIME_PID $$
```

  - Scheduling Class RR, Prio 1 (niedrig) – Prio 99 (hoch)

```
ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,args
 5539  5539 RR       99   - 139   3  0.0 S    wait           /bin/sh /etc/init.d/init.cssd oclsomon
 5569  5569 RR       99   - 139   0  0.0 S    wait           /bin/sh /etc/init.d/init.cssd daemon
 5884  5884 RR       99   - 139   3  0.0 SL   select         /oracle/CRS/bin/oprocd.bin run -t 1000 -m 10000
                                                -hsi 5:10:50:75:90 -f
 5889  5889 RR       99   - 139   0  0.0 S    wait           /sbin/runuser -l oracle -c /bin/sh -c 'cd
                                                /oracle/CRS/log/rac1/cssd/oclsomon; ulimit -c unlimited;
                                                /oracle/CRS/bin/oclsomon  || exit $?'
 5890  5890 RR       99   - 139   0  0.0 S    wait           /bin/sh -c cd
                                                /oracle/CRS/log/rac1/cssd/oclsomon; ulimit -c unlimited;
                                                /oracle/CRS/bin/oclsomon  || exit $?
 5914  5914 RR       99   - 139   1  0.0 Ss   nanosleep      /oracle/CRS/bin/oclsomon.bin
 6032  6032 RR       99   - 139   0  1.6 SLl  futex          /oracle/CRS/bin/ocssd.bin
 7016  7016 RR        1   -  41   1  0.0 Ss   poll           asm_lms0_+ASM1
 7556  7556 RR        1   -  41   2  0.8 Ss   poll           ora_lms0_RAC1
 7563  7563 RR        1   -  41   2  0.8 Ss   poll           ora_lms1_RAC1
```

Testphase

Stabilitätstests (7) – Process Table overflow

Testszenario	Beschreibung	Erwartetes Resultat
Process Table overflow	Das System wird einer "fork bomb" ausgesetzt. Kommando dafür unter Linux lautet: `:(){ :|:& };:`	Trotz einer Run Queue von bis zu 30.000 konnte das System nicht wesentlich beeinträchtigt werden. Die Antwortzeiten waren erhöht, aber die Stabilität war gegeben.

```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
0  0      0 14687312  34364 1024748    0    0    19     3 1181 2292  0  0 99  0  0
 0  0      0 14687436  34372 1024756    0    0    51    52 1086 2145  0  0 99  0  0
7758  3      0 13305356  34380 1024748    0    0    35    51 1205 16227  6 64 29  0  0
17299 11      0 11597652  34380 1024764    0    0    19     4 1090 18833  7 93  0  0  0
25502 12      0 10132592  34396 1024748    0    0    67   115 1210 16466  6 94  0  0  0
29425  8      0 9156500  34396 1024760    0    0    35     4 1067 13002  6 94  0  0  0
26359 12      0 8842736  34396 1024760    0    0    19     1  918 8312  8 92  0  0  0
23429 15      0 8546360  34436 1024728    0    0    51   114  889 8236  8 92  0  0  0
11713 22      0 7955596  34436 1024728    0    0    19     1 1115 19641 13 87  0  0  0
44  0      0 7220364  34460 1024748    0    0    51   178 2023 24591 17 82  1  0  0
127  0      0 6868652  34460 1024748    0    0    51    43 4090 17279 13 87  0  0  0
24971  0      0 7317348  34460 1024804    0    0    19     6 2134 11623 10 90  0  0  0
2808  5      0 13364384  34476 1024788    0    0    35     3 1282 35819  3 97  1  0  0
8909  4      0 12439300  34496 1024788    0    0    51    36 1449 24735  8 92  0  0  0
2380  1      0 13957212  34524 1024760    0    0    35   109 1161 30713  6 94  0  0  0
13733  7      0 12062884  34532 1024920    0    0    35    90 1184 23334  8 92  0  0  0
23080  7      0 10197948  34540 1024524    0    0    51    53 1290 24035  8 92  0  0  0
12034  6      0 9248472  34540 1024668    0    0    19     2 1111 28634 11 89  0  0  0
5212  6      0 8023104  34540 1024668    0    0    35     3 1255 27195 12 88  0  0  0
```

Testphase

Stabilitätstests (8) – Memory Starvation

Testszenario	Beschreibung	Erwartetes Resultat
Memory Starvation (Swapping)	Durch synthetisches Allokieren von Arbeitsspeicher durch ein C-Programm wird das System zum Swapping gezwungen. `leak2 <mb> <seconds>` (erhältlich auf Anfrage)	Die Performance des Systems ist aufgrund des Swappings beeinträchtigt. Solange allerdings genügend Swap-Speicher zur Verfügung steht, beleibt das System stabil. Antwortzeiten stark beeinträchtigt.

```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0 24 3220120  79160    700  93704  156 45804   372 45827 1492 2473  0  3  1 95  0
10 17 3411768  89700    716  95324  180 63920   448 63969 1726 3027  0  4 13 84  0
 0 24 3546520  80804    716  94700  332 44955   663 44978 1565 2553  0  4  4 92  0
 0 25 2996248  80356    732  92736  528 52165   990 52219 1384 2619  1 14  4 81  0
 0  6 2414488 3954928    752  95936  817 20824  2010 20898 1321 2588  1 11 42 46  0
 0  5 2412840 3935344    908 114456 1872     0  7686    27 1432 2677  0  0 84 15  0
 0  2 2411468 4921480   1252 127232 1743     0  5668   119 1412 3104  1  1 83 15  0
 0  0 2410888 4914720   1532 134876  632     0  3050   125 1266 2535  1  0 93  6  0
 0  0 2410632 4914804   1556 135420  332     0   367    90 1172 2418  0  0 99  1  0
```

```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
16  7 8193140  78844    748  96068  387  159  1775   292 1238 2095  0 99  0  1  0
14 14 8193140  78980    724  95064  844  231  1488   287 1190 2117  0 100  0  0  0
32  5 8193140  79604    580  89188  939  392  2305   453 1800 3487  0 100  0  0  0
24  3 8193140  78832    524  82936 1187  553  2754   609 1307 2329  0 100  0  0  0
20  5 8193140  79732    484  78924   21   21   380    63 1081 1790  0 100  0  0  0
29  2 8193140  79172    524  78436  313   85  1700   155 1201 2204  0 100  0  0  0
25  3 8193140  79684    488  77164  381   93  1107   125 1129 2034  0 100  0  0  0
33  1 8193140  79060    396  72360 1024  289  3505   365 3055 5100  0 100  0  0  0
31  6 8193140  78856    372  74528  227   61   819    89 1161 1942  0 100  0  0  0
45  4 8193140  78836    356  71812  420  107   916   127 1239 2149  0 100  0  0  0
37  7 8193140  80424    336  70880  244   57   687    78 1142 1835  0 100  0  0  0
46  8 8193140  78924    332  72872   32    5   996    25 1178 1805  0 100  0  0  0
40  4 8193140  82500    328  69888  217   36   726    70 1134 1918  0 100  0  0  0
```

ORA-SOLUTIONS.NET
Mastering Oracle Performance, High Availability, Manageability
Martin Decker

RAC Troubleshooting

RAC Troubleshooting

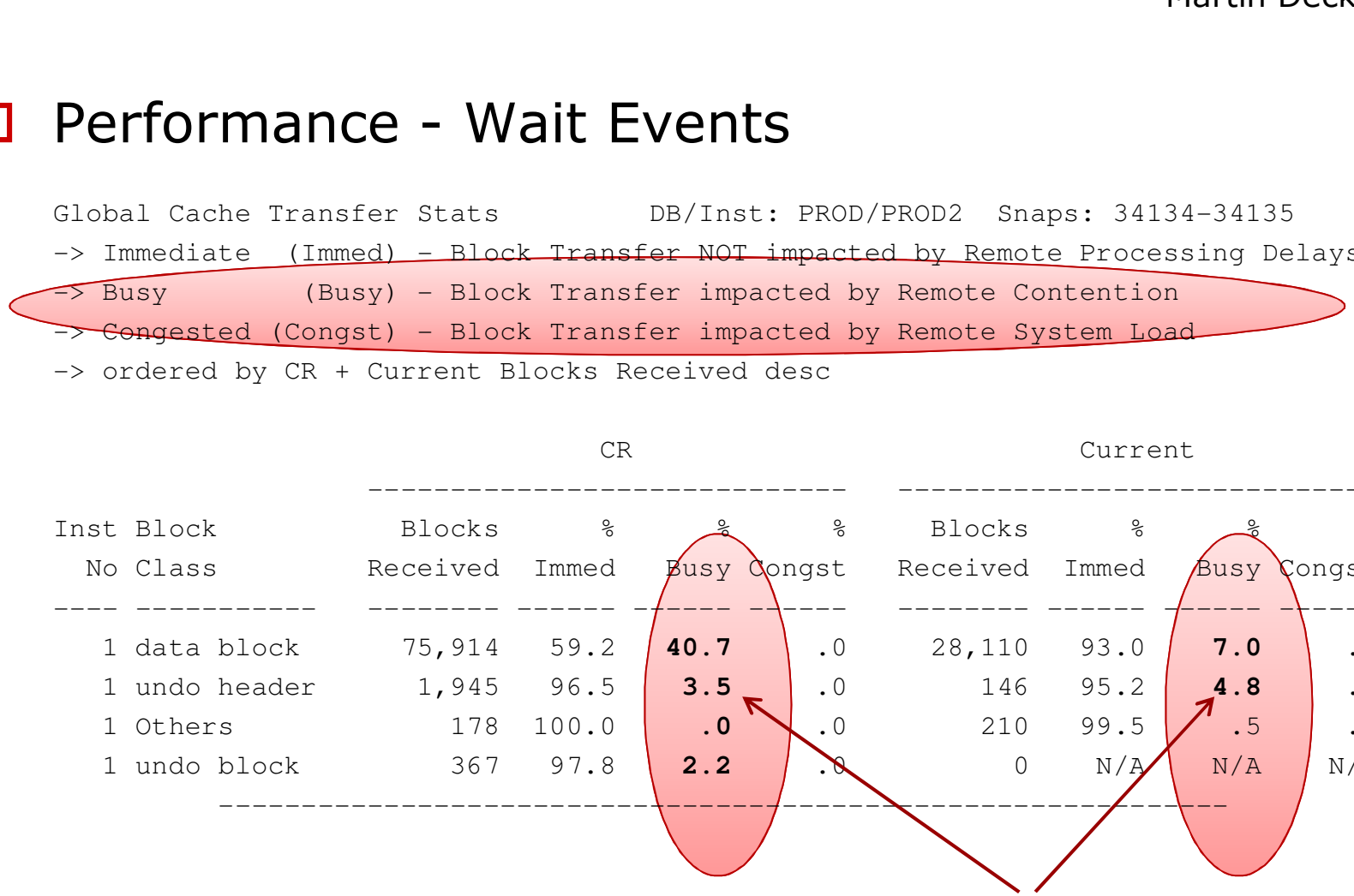
- Performance – Top Timed Wait Events
  - gc [current /cr] [multiblock] request **(Placeholder)**
  - gc [current/cr] block [2/3]-way (requestor/master/holder)
  - gc [current/cr] block **busy** - Verzögerung aufgrund von high concurrency, contention (Current: w/w)
  - gc [current/cr] grant 2-way – Physical Read Granted, nicht in Global Cache
  - gc current grant **busy** – z.B. HWM
  - gc [current/cr] [block/grant] **congested**: Starke Verzögerung aufgrund von CPU Überbelastung, Run Queue, Memory Starvation (Paging, Swapping), LMS Überlastung
  - gc [current/cr] [**failure/retry**]: Checksum Fehler oder Paket verloren
  - gc buffer **busy:** local contention, andere session auf lokalem Node hat lock angefordert

Quelle:http://oukc.oracle.com/static05/opn/oracle9i_database/49466/040908_49466_source/index.htm

Performance - Wait Events

```
Global Cache Transfer Stats                DB/Inst: PROD/PROD2  Snaps: 34134-34135
-> Immediate   (Immed) - Block Transfer NOT impacted by Remote Processing Delays
-> Busy          (Busy) - Block Transfer impacted by Remote Contention
-> Congested (Congst) - Block Transfer impacted by Remote System Load
-> ordered by CR + Current Blocks Received desc
```

		CR				Current			
Inst No	Block Class	Blocks Received	% Immed	% Busy	% Congst	Blocks Received	% Immed	% Busy	% Congst
1	data block	75,914	59.2	**40.7**	.0	28,110	93.0	**7.0**	.0
1	undo header	1,945	96.5	**3.5**	.0	146	95.2	**4.8**	.0
1	Others	178	100.0	**.0**	.0	210	99.5	.5	.0
1	undo block	367	97.8	**2.2**	.0	0	N/A	N/A	N/A



Contention / Concurrency

RAC Troubleshooting

- Performance – Latenzzeiten

Request-Time	Best Case	Avg.	Problem
Average time to process cr block request	0,1 ms	1 ms	> 10 ms
Average time to process current block request	0,1 ms	3 ms	> 23 ms
Receive-Time			
Avg global cache cr block receive time	0,3 ms	4 ms	> 12 ms
Avg global cache current block receive time	0,3 ms	8 ms	> 30 ms

AWR:

```
          Avg global cache cr block receive time (ms):      87.4
    Avg global cache current block receive time (ms):      11.5

        Avg global cache cr block build time (ms):         0.0
         Avg global cache cr block send time (ms):         0.0
        Avg global cache cr block flush time (ms):         2.2

      Avg global cache current block pin time (ms):        0.4
     Avg global cache current block send time (ms):        0.0
    Avg global cache current block flush time (ms):        3.0
```

CR Block request time

Current Block request time

Node Evictions

- Node Fencing – Schutz vor Datenkorruption durch Split Brain (STONITH)

- Analyse:
  - MetaLink Note: 265769.1 – Troubleshooting CRS Reboots
  - DIAGWAIT 13:
    `CRS_home/bin/crsctl set css diagwait 13 –force`
  - OSWatcher (Load, Interconnect Latency)
  - /var/log/messages
  - CRS_HOME/log/<hostname>/cssd/ocssd.log
  - /etc/oracle/oprocd/oprocd.log
  - IPD

Lost Blocks

- Problematisch nur, wenn Metrik „gc cr block lost" einen wesentlichen Anteil ausmacht.
- Prüfung von Cluster Interconnect (Bonding, ISL Link bei mehreren Switches, Spanning Tree)
- ifconfig –a: TX/RX dropped/errors/frame

```
eth0      Link encap:Ethernet
          inet addr:192.168.10.1  Bcast:192.168.10.255  Mask:255.255.255.0
          RX packets:72358474 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72612322 errors:0 dropped:0 overruns:0 carrier:0
```

- netstat –s: packet reassembles failed, fragments dropped after timeout
- Wait Event: gc cr block lost
- MetaLink Note 563566.1

Tools

Tools – OS Watcher

- Download MetaLink Note 301137.1
- sehr empfehlenswert für jede RAC Installation
- geringer Overhead
- Sammelt OS Statistiken in $OSWHOME/archive (vmstat, mpstat, netstat, private interconnect latency, ps, top, iostat)

```
./startOSW.sh [interval in secs] [history in hours] [gzip]
./startOSW.sh 20 24 gzip
```

- Startup Package osw-service.rpm (MetaLink 580513.1)
  - /etc/sysconfig/osw
  - Bei Verwendung von gzip - editieren von /etc/init.d/osw:

```
./startOSW.sh ${OSWINTERVAL} ${OSWRETENTION}
in
./startOSW.sh ${OSWINTERVAL} ${OSWRETENTION} gzip
```

- Graphische Auswertung:

```
$ORACLE_HOME/jre/1.4.2/bin/java -jar oswg.jar -i archive -B Sep 01 00:00:00 2009 -E
     Oct 01 00:00:00 2009 -P SEP2009
```

Tools – LTOM

- LTOM – Lite OnBoard Monitor
- automatische Echtzeit Problemerkennung und Sammlung von Diagnose-Daten
- Installationsanleitung MetaLink 352363.1 oder http://www.ora-solutions.net/web/2008/12/17/installing-ltom-for-rac-hanganalyze/
- Features
  - **Automatic Hang Detection**: automatisches Hanganalyze bei DB / Instance Hangs
  - **System Profiler:** Profil der System-Auslastung über Zeit (OS + DB)
  - **Automatic Session Tracing:** automatisches Aktivieren von 10046 tracing nach bestimmten Regeln (z.B. Wait Events, CPU consumption, users)
- Empfehlung: sehr resourcenintensiv, deshalb nur für Analyse von wiederkehrendem Problem zu empfehlen
- Disclaimer von Oracle bzgl. Automatic Hang Detection und Automatic Session Tracing:
  - This feature should only be used at the direction of Oracle Support or by experienced dba's. The automated collection of heavy tracing on a production system can have a significant performance impact on that system. The user needs to be aware of the consequences of generating this level of tracing and should proceed with caution.

Tools – procwatcher

- Download MetaLink Note 459694.1
- Tool von Oracle zur Ermittlung von Stacktraces von Oracle Prozessen (RDBMS und Clusterware)
- Empfehlung: nur für Analyse von wiederkehrendem Problem zu empfehlen
- Konfiguration:

```
INTERVAL=300 (time between runs)
THROTTLE=5 (stacktraces at once)
IDLECPU=3 (prw sleeps until if host is >97% busy)
EXAMINE_CRS=false
EXAMINE_BG=true
BGPROCS="_dbw|_smon|_pmon|_lgwr|_lmd|_lms|_lck|_lmon|_ckpt|_arc|_rvwr|_gmo
    n"
CRSPROCS="crsd.bin|ocssd.b|evmd.bin|evmlogge|racgimon|racge|racgmain|racgo
    ns.bin|ocls"
```

- Start/Stop/Status

```
./prw.sh start <days of history>
./prw.sh stop
./prw.sh stat
```

Tools - HANGFG

- HANGFG (Hang File Generator) – MetaLink 362094.1
- im Falle eines Instance / Database Hangs manuell vom DBA auszuführen
- ./hangfg.sh <ARG>

ARG	Description
1	geringe Belastung: 2 x hanganalyze level 3, bestimmt dann automatisch ob zusätzlich 1 x hanganalyze level 4 ausgeführt werden kann.
2	mittlere Belastung: (default) 1 x hanganalyze level 3, bestimmt dann automatisch ob zusätzlich 2 x hanganalyze level 4 ausgeführ werden kann, wenn nicht 1 x hanganalyze level 3 1 x systemstate level 266
3	hohe Belastung: 2 x hanganalyze level 4 2 x systemstate level 266

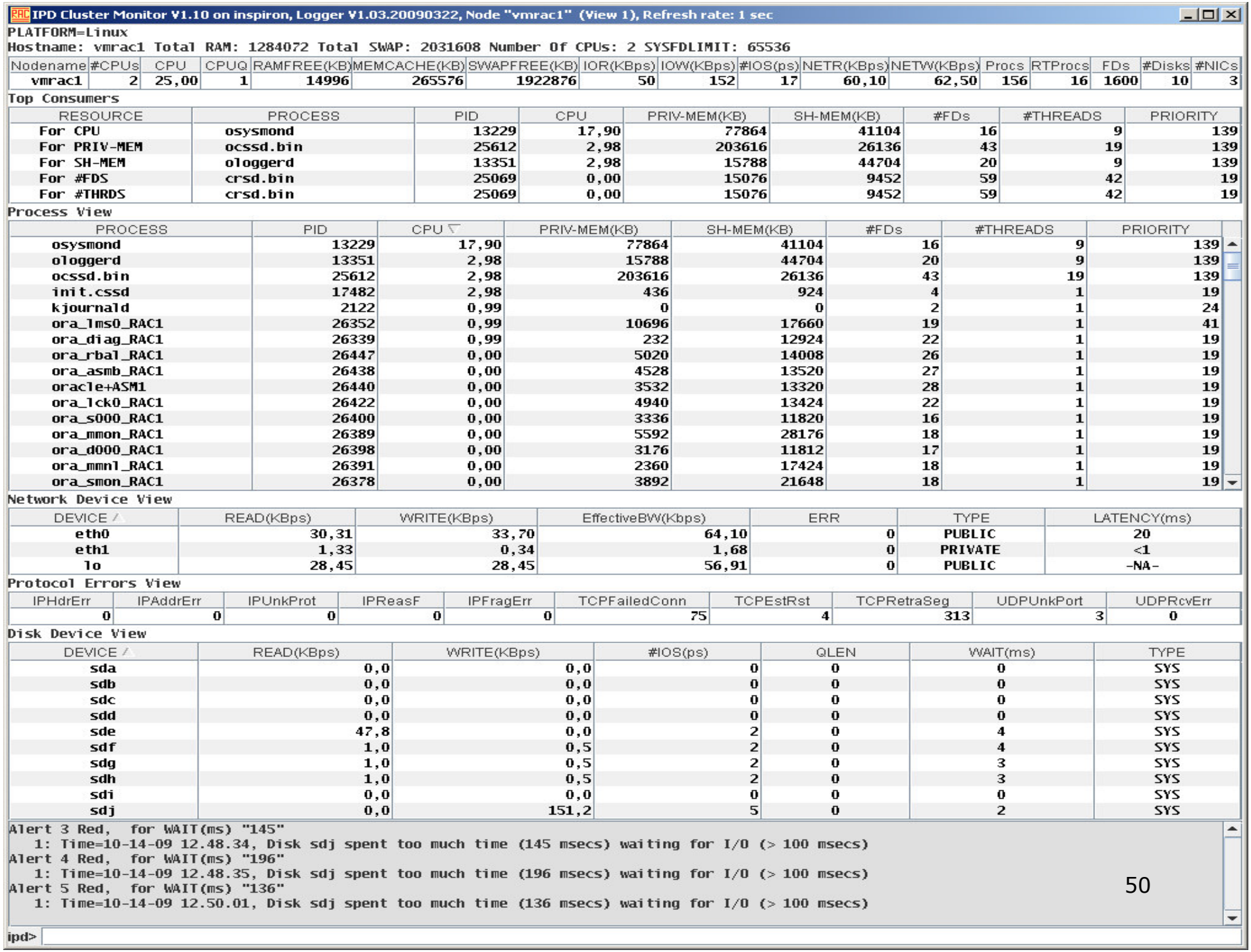
Tools – IPD/OS

- Instantenous Problem Detection (IPD/OS) / Cluster Health Monitor
- http://www.oracle.com/technology/products/database/clustering/ipd_download_homepage.html
- Echtzeit-Aufzeichnen von OS Statistiken für Diagnose von Node Evictions
- GUI Anzeige von Echtzeit-Daten oder in BerkeleyDB aufgezeichneten Daten
- Installation auf beiden Nodes
- Resourcenverbrauch
  - mind. 2 GB Plattenplatz pro Node
  - 2 Realtime Prozesse
  - ca. 20% CPU Utilization auf Testserver
  - 200 MB RAM (hoher Overhead)
  - 200 MB Messdaten in BerkeleyDB für 2 Stunden Messung
- Installation

```
$ ./crfinst.pl -i node1,node2 -b /opt/oracrfdb -m node1
vmrac1# /opt/crfuser/install/crfinst.pl -f -b /opt/oracrfdb
vmrac2# /opt/crfuser/install/crfinst.pl -f -b /opt/oracrfdb
```

- Stop/Start:

```
/etc/init.d/init.crfd enable
/etc/init.d/init.crf disable
/etc/init.d/init.crf stop
```


IPD Cluster Monitor V1.10 on inspiron, Logger V1.03.20090322, Node "vmrac1" (View 1), Refresh rate: 1 sec
PLATFORM=Linux
Hostname: vmrac1 Total RAM: 1284072 Total SWAP: 2031608 Number Of CPUs: 2 SYSFDLIMIT: 65536
Nodename #CPUs CPU CPUQ RAMFREE(KB) MEMCACHE(KB) SWAPFREE(KB) IOR(KBps) IOW(KBps) #IOS(ps) NETR(KBps) NETW(KBps) Procs RTProcs FDs #Disks #NICs
vmrac1 2 25,00 1 14996 265576 1922876 50 152 17 60,10 62,50 156 16 1600 10 3
Top Consumers
RESOURCE PROCESS PID CPU PRIV-MEM(KB) SH-MEM(KB) #FDs #THREADS PRIORITY
For CPU osysmond 13229 17,90 77864 41104 16 9 139
For PRIV-MEM ocssd.bin 25612 2,98 203616 26136 43 19 139
For SH-MEM ologgerd 13351 2,98 15788 44704 20 9 139
For #FDS crsd.bin 25069 0,00 15076 9452 59 42 19
For #THRDS crsd.bin 25069 0,00 15076 9452 59 42 19
Process View
PROCESS PID CPU PRIV-MEM(KB) SH-MEM(KB) #FDs #THREADS PRIORITY
osysmond 13229 17,90 77864 41104 16 9 139
ologgerd 13351 2,98 15788 44704 20 9 139
ocssd.bin 25612 2,98 203616 26136 43 19 139
init.cssd 17482 2,98 436 924 4 1 19
kjournald 2122 0,99 0 0 2 1 24
ora_lms0_RAC1 26352 0,99 10696 17660 19 1 41
ora_diag_RAC1 26339 0,99 232 12924 22 1 19
ora_rbal_RAC1 26447 0,00 5020 14008 26 1 19
ora_asmb_RAC1 26438 0,00 4528 13520 27 1 19
oracle+ASM1 26440 0,00 3532 13320 28 1 19
ora_lck0_RAC1 26422 0,00 4940 13424 22 1 19
ora_s000_RAC1 26400 0,00 3336 11820 16 1 19
ora_mmon_RAC1 26389 0,00 5592 28176 18 1 19
ora_d000_RAC1 26398 0,00 3176 11812 17 1 19
ora_mmnl_RAC1 26391 0,00 2360 17424 18 1 19
ora_smon_RAC1 26378 0,00 3892 21648 18 1 19
Network Device View
DEVICE READ(KBps) WRITE(KBps) EffectiveBW(Kbps) ERR TYPE LATENCY(ms)
eth0 30,31 33,70 64,10 0 PUBLIC 20
eth1 1,33 0,34 1,68 0 PRIVATE <1
lo 28,45 28,45 56,91 0 PUBLIC -NA-
Protocol Errors View
IPHdrErr IPAddrErr IPUnkProt IPReasF IPFragErr TCPFailedConn TCPEstRst TCPRetraSeg UDPUnkPort UDPRcvErr
0 0 0 0 0 75 4 313 3 0
Disk Device View
DEVICE READ(KBps) WRITE(KBps) #IOS(ps) QLEN WAIT(ms) TYPE
sda 0,0 0,0 0 0 0 SYS
sdb 0,0 0,0 0 0 0 SYS
sdc 0,0 0,0 0 0 0 SYS
sdd 0,0 0,0 0 0 0 SYS
sde 47,8 0,0 2 0 4 SYS
sdf 1,0 0,5 2 0 4 SYS
sdg 1,0 0,5 2 0 3 SYS
sdh 1,0 0,5 2 0 3 SYS
sdi 0,0 0,0 0 0 0 SYS
sdj 0,0 151,2 5 0 2 SYS
Alert 3 Red, for WAIT(ms) "145"
1: Time=10-14-09 12.48.34, Disk sdj spent too much time (145 msecs) waiting for I/O (> 100 msecs)
Alert 4 Red, for WAIT(ms) "196"
1: Time=10-14-09 12.48.35, Disk sdj spent too much time (196 msecs) waiting for I/O (> 100 msecs)
Alert 5 Red, for WAIT(ms) "136"
1: Time=10-14-09 12.50.01, Disk sdj spent too much time (136 msecs) waiting for I/O (> 100 msecs)
ipd>

Tools – RH Hangwatch

- Download: http://people.redhat.com/astokes/hangwatch/
- Messung von 1-min Load Average und automatisches Sammeln von Kernel Informationen (sysRq) bei Überschreitung von Schwellwert
- Installation

```
rpm –Uvh hangwatch*rpm
service hangwatch start
```

- Sysrq Trigger (http://www.kernel.org/doc/Documentation/sysrq.txt)
  - m – Outputs memory statistic
  - p – Outputs all flags and registers
  - t - Outputs a list of processes (tasks)

Fazit

- Test / Test / Test: Ausfallsicherheit (Redundanz) muss getestet und dokumentiert werden

- „globales“ Performance-Monitoring
- „Know Your Tools“

Q & A

Martin Decker
ora-solutions.net

ORACLE
10g Certified Master

E-Mail: martin.decker@ora-solutions.net
Internet: http://www.ora-solutions.net
Blog: http://www.ora-solutions.net/web/blog/