RAC on Unix:
Pre-Production Testing & Troubleshooting

Martin Decker
ora-solutions.net
München

Schlüsselworte: RAC, Unix, Testing, Go-Live, Redundanz, Stabilitätstests

Einleitung

Die Inbetriebnahme einer geschäftskritischen Datenbank wird oft als Oracle Real Application Cluster (RAC) Datenbank realisiert. Das Management erwartet sich aufgrund der beachtlichen Lizenzkosten eine Erhöhung der Verfügbarkeit des Systems. Doch oftmals werden die hohen Erwartungen des Managements nicht erfüllt. Im schlimmsten Fall ist die Verfügbarkeit des Real Application Clusters deutlich geringer als bei einer Single-Instance Datenbank. Zusätzliche Komponenten wie Cluster Interconnect und Shared Storage müssen redundant konfiguriert werden und erhöhen die Komplexität der Installation. Oft kommt es bei der Implementierung dazu, daß die benötigten Redundanzen in den verschiedenen Layers nicht korrekt konfiguriert sind oder die Applikation einfach die Anforderungen für den skalierbaren Betrieb als Real Application Cluster Datenbank nicht erfüllt. Dieser Vortrag zeigt, wie während der Implementierungsphase aber vor Go-Live durch bestimmte Test-Szenarien und Benchmarks diese Risiken geprüft und behoben werden können. Des weiteren werden eine Reihe von Best Practices, die sich bei RAC Implementierungen in der Praxis bewährt haben, vorgestellt. Es werden bekannte Fehler einer out-of-the-box Installation gezeigt, die manuell behoben werden müssen. Abschließend werden einige Tools vorgestellt, mit denen RAC Troubleshooting durchgeführt werden kann, z.B. OS Watcher, Procwatcher, LTOM, etc.

Projektphasen

Damit ein RAC Implementierungsprojekt zum Erfolg wird, ist sehr viel Know-How notwendig. Dabei setzt sich ein Projektteam nicht selten aus Spezialisten aus den Bereichen Netzwerk, Storage, Systemadministration und Datenbankadministrator zusammen. Speziell im Oracle Kernel gibt es bei RAC Systemen einige Konzepte, die jeder Datenbankadministrator, der mit RAC Systemen zu tun hat, kennen sollte. Beispiele dafür sind Dynamic Remastering, Cache Fusion, Global Resource Directory, etc. Eine exzellente Erklärung dieser Konzepte findet sich im Buch "Oracle Database 10g - Real Application Clusters Handbook" von K. Gopalakrishnan.

Ein typisches RAC Implementierungsprojekt setzt sich aus folgenden Phasen zusammen:

Konzeptionierungsphase

In dieser Phase wird die gewünschte Verfügbarkeit der Datenbank vom Management bzw. Business Owner festgelegt. Es sollte beachtet werden, dass ein Real Application Cluster für hochverfügbare Anwendungen mit einer Data Guard Standby-Datenbank oder einem Disk-Backup in Form von Image Copies auf einem unabhängigen Storage-System ausgestattet sein muss. Andernfalls kommt es bei einer physikalischen oder logischen Korruption der Datenbank-Datafiles zu einer Downtime, die je nach Datenbankgröße mehrere Stunden betragen kann. Hier sei auf die von Oracle veröffentlichte Maximum Availability Architecture (MAA)[1] verwiesen. Diese MAA-Dokumente beschreiben eine Reihe von Best Practices, damit in jede Architekturebene die benötigte Redundanz integrieret wird um geplante oder ungeplante Downtimes zu vermeiden. Dabei versteht es sich von selbst, dass mehrere Netzwerk-Interfaces mittels Bonding (Teaming, Auto Port Aggregation) redudant konfiguriert werden sowie Multipathing über mehrere Host Bus Adapter (HBA) bzw. FC-Switches eingerichtet wird. Nicht zu vergessen sind auch die Themen Zeitsynchronisierung (speziell -x Option bei ntpd) und Device Persistency (udev, asmlib) sowie die Konfiguration des hangcheck-timers. Das redundate Storage-System sieht ein performantes Disk-Setup vor, d.h. RAID5[2] ist nach Möglichkeit zu vermeiden. Besonders empfehlenswert ist es, die Partitionsgrenzen an die Stripe Width anzupassen, um ein "Misalignment" zu verhindern. Dadurch kann bis zu 30% Performance gewonnen werden.[3] Bei Bedarf kann die I/O Peformance noch zusätzlich gesteigert werden, indem nur die äußersten Tracks einer Festplatte benutzt werden. Oracle ASM 11gR2 verfügt über ein Feature namens "Intelligent Data Placement - IDP", bei dem die meist verwendeten Daten im äußeren Bereich der Disk und die seltener verwendeten Daten im inneren Bereich der Disk gespeichert werden.[4]

Für den Cluster-Interconnect stehen 1/10 Gigabit Ethernet bzw. Infiniband zur Verfügung. Bei Ethernet sollte nach Möglichkeit das Feature "Jumbo Frames" aktiviert werden, wodurch Ethernet Frames von bis zu 9.000 Bytes Payload verwendet werden können. Dies ist besonders effizient, da Datenbankblöcke mit einer db_block_size von 8.192 Bytes ohne Aufsplittung in einem Frame transportiert werden können.

Aber nicht nur die Hardware und RDBMS Software muss konzeptionell geplant werden, sondern auch die Applikation. Es muss geprüft werden, ob der Applikationshersteller die Applikation für den Einsatz auf Real Application Clustern freigegeben hat und auch Support leistet. Im Idealfall sollte er darüber hinaus das Potential von RAC Systemen ausschöpfen, indem die Applikation speziell für RAC Systeme optimiert wird. Hier ist Workload Management mittels Oracle Services und Load Balancing Advisories zu erwähnen, womit ermöglicht wird, verschiedene Applikationskomponenten auf einen oder mehrere Cluster-

---

[1] http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm

[2] The Battle Agaist *Any* Raid Five - http://www.baarf.com/

[3] Aligning ASM Disks on Linux - http://www.pythian.com/news/411/aligning-asm-disks-on-linux

[4] ASM - Intelligent Data Placement,
http://download.oracle.com/docs/cd/E11882_01/server.112/e10881/chapter1.htm#FEATURENO06963

Nodes zu beschränken und statistische Auswertungen über die Belastung von verschiedenen Services zu treffen. Vorbildhafte Applikationen setzen zudem noch die Attribute Module und Action über DBMS_APPLICATION_INFO. Leider werden die Oracle Services bei geplanten Restarts der Server nicht automatisch gestartet. Falls gewünscht, lässt sich dies über FAN Callout Scripts implementieren.[5] Damit die Applikation bei Ausfall eines Nodes oder Instanz transparent verfügbar bleibt, muss die Applikation Fast Application Notification (FAN) und Fast Connection Failover (FCF) implementieren. Bei Ausfall eines Nodes wird die Applikation proaktiv informiert und kann entsprechende Gegenmassnahmen, z.B. retransmit der Transaktion, treffen.

Installationsphase

In der Installationsphase wird die Hardware aufgebaut, Firmwares geprüft/aktualisiert, die Betriebssysteme, Treiber, etc. installiert und konfiguriert. Es ist empfohlen, sowohl für Betriebssystem, als auch für Treiber die neuesten Patches/Versionen zu installieren. Es sollten vorher allerdings zwingend die Abhängigkeiten von Treiber, Kernel, ASMLIB, etc. geprüft werden.

Für derzeitige Implementierungen sind 64-bit Betriebssysteme zu präferieren. Bei Linux sollte das Feature "Hugepages" verwendet werden. Durch die Vergrößerung der Pagesize für die Oracle SGA läßt sich die CPU Belastung des Kernels für Memory Adressübersetzungen und der Kernel Memory Verbrauch für die PageTables reduzieren. Dieser "gesparte" Arbeitsspeicher kann dann effizienter für den Oracle Buffer Cache genutzt werden. Zu beachten ist allerdings, dass dieses Feature nicht mit dem Automatic Memory Management von Oracle 11g (MEMORY_TARGET) kompatibel ist, welches mittels Ramdisk implementiert ist.

Empfehlenswert ist nach der OS Installation und Konfiguration die Verwendung der Health Check Verification Engine (HCVE) des RDA, mittels dem die Voraussetzungen für eine Oracle Installation geprüft werden können.

Speziell für Oracle Real Application Cluster wurde zudem das Utility "Cluster Verification Utility - CVU" entwickelt. Dieses Tool, das ab der Version 11gR2 auch mit FixUp Scripts ausgestattet ist, bietet die Möglichkeit, das System während den einzelnen RAC Installationsphasen (Stages) zu testen um somit einen reibungslosen Installationsprozess zu gewährleisten. Es wird empfohlen vor Installationsbeginn die aktuellste Version von CVU[6] von OTN herunterzuladen.

Für Oracle Clusterware veröffentlicht Oracle regelmäßig "CRS Recommended Bundle" Patches, die bekannte, teilweise kritische, Bugs beheben. Beispiele dafür sind Node Reboots, Prozess Coredumps, OCR Device Inconsistencies, etc. Für das RDBMS veröffentlicht Oracle

---

[5] Oracle Services Autostart - http://www.ora-solutions.net/web/2009/03/19/autostart-oracle-services-at-clusterware-startup/

[6] CVU - http://www.oracle.com/technology/products/database/clustering/cvu/cvu_download_homepage.html

seit Juli 2009 quartalsweise Patch Set Updates (PSU), die neben sicherheitsrelevanten Patches auch sonstige kritische Bugfixes enthalten. Es wird empfohlen, bei Neuinstallationen jeweils das aktuellste Patch Set Update zu installieren. Dies enthält neben generischen Bugfixes auch Bugfixes in Bezug auf Oracle Services, Oracle RAC, Oracle physical/logical Data Guard sowie Data Guard Broker wofür bisher eigenständige Patch Bundles notwendig waren.

Bei der Standardinstallation der Version 10.2.0.4 sind noch folgende Korrekturen notwendig:

- Der Initialisierungsparameter *local_listener* ist standardmäßig nicht gesetzt und bewirkt, dass der Listener die physikalische IP statt der virtuellen IP bei den Listenern registriert wird. Die physikalische IP Adresse sollte allerdings bei RAC Systemen nicht für Client Connectivity genutzt werden. Das Problem kann gelöst werden, indem der Initialisierungsparameter *local_listener* auf einen TNS Eintrag mit der lokalen virtuellen IP gesetzt wird.

- Das zweite Problem besteht darin, dass standardmäßig die Abhängigkeit zwischen ASM Instanz und Datenbank-Instanz in Clusterware nicht konfiguriert ist.[7] Dies kann zu Verzögerungen und Fehlern beim Startup der Resourcen führen.

- Der Clusterware Parameter *DIAGWAIT* sollte auf 13 gesetzt werden, damit bei Node Evictions die für die Analyse benötigten Logfiles erzeugt werden.

Abschließend wird das Storage-System für die Benutzung konfiguriert und eine Demo-Datenbank erstellt.

Test-Phase

Dieser Phase kommt neben der Konzeptionierungsphase am meisten Bedeutung zu. Hier wird die Konfiguration auf Herz und Nieren getestet. Neben Unstimmigkeiten der Konfiguration können in dieser Phase auch eventuelle Bugs in OS, Treibern, RDBMS, ASM, Clusterware identifiziert und gegebenenfalls behoben werden.

Zu Beginn empfiehlt es sich, einen synthetischen I/O Benchmark-Test mittels iozone, bonnie oder "ORION (Oracle I/O Calibration Tool)" durchzuführen. ORION simuliert Oracle I/O Zugriffe von unterschiedlichen Workloads, z.B. small random I/O, large sequential I/O und large random I/O. Für spätere Performance-Analysen ist es sehr hilfreich, die physikalischen Grenzen des I/O Subsystems ermittelt zu haben.

Zur Prüfung des Datendurchsatzes des Private Interconnect empfiehlt es sich, Datentransfers zwischen den Nodes zu testen. Ideal dafür ist das Tool *iperf*[8] unter Linux. Hier sollten Datentransferraten von mind. 900 MBit/s bei einem 1 GbE Interconnect erzielbar sein.

---

[7] Add Dependency of Database Instance on ASM to OCR - http://www.ora-solutions.net/web/2008/12/03/add-dependency-of-database-instance-on-asm-to-ocr/

[8] iperf - http://iperf.sourceforge.net

Anschließend folgt der Benchmark des Datenbank-Systems mittels eines Benchmark-Tools, z.B. Swingbench[9] oder Hammerora[10]. Idealerweise wird der Benchmark über mehrere Tage durchgeführt, sodaß die Stabilität der Hardware unter Belastung verifiziert werden kann. Gesammelt werden neben den Transactions per Minute (TPM) noch relevante Systemdaten wie CPU Utilization, I/O Operationen pro Sekunde (IOPS), Durchsatz in MB/sek., etc.

Nach diesen synthetischen Tests sollte ein funktionaler Test der Applikation mit der Real Application Cluster Datenbank durchgeführt werden. Ist dieser Test zufriedenstellend, kann mit den Stabilitätstests fortgesetzt werden. Bei den Stabilitätstests werden verschiedene Fehlerszenarien herbeigeführt und die Reaktion des Systems überprüft und protokolliert. Bei jedem Szenario sollte das erwartete Resultat sowie die tatsächliche Reaktion des Systems verglichen werden. Zudem wird der Datenbankadministrator in der Behebung von verschiedenen Fehlerzuständen trainiert.

Folgende Testszenarien sind sinnvoll:

Testszenario	Beschreibung
One RAC Instance Failure	*kill -9 <pmon pid>*
All Instance Failure	*kill -9 <pmon pid>* auf allen Nodes
ASM Instance Failure	*kill -9 <pmon pid>* von ASM Instanz
All ASM Instance Failure	*kill -9 <pmon pid>* von ASM auf allen Nodes
Listener Failure	*kill -9 <listener pid>*
Node Failure	Poweroff, Power Kabel ziehen, Blade aus Bladecenter herausziehen, Kernel Panic: *echo c > /proc/sysrq-trigger*
All Node Failure	Poweroff, Power Kabel ziehen, Blade aus Bladecenter herausziehen, Kernel Panic: *echo c > /proc/sysrq-trigger*
crsd.bin Process Failure	*kill -9 <crsd.bin process pid>*
evmd Process Failure	*kill -9 <evmd process pid>*
ocssd Process Failure	*kill -9 <ocssd.bin process pid>*
Korruption OCR Device	Überschreiben von Teilen des OCR Devices mit Nullen (*dd*)
Reparatur OCR Device	*ocrconfig -replace ocr|ocrmirror /dev/raw/rawX*
Korruption Voting Disk	Überschreiben von einer Voting Disk mit Nullen (*dd*)
Reparatur Voting Disk	Bei Oracle 10gR2 nur mit gestoppter Clusterware möglich: *crsctl delete css votedisk /dev/raw/rawX -force* *crsctl add css votedisk /dev/raw/rawX -force*
Public NIC Failure	Kabel ziehen bzw. Switch Port deaktivieren. Bonding sollte Failover durchführen und Applikation nichts davon

[9] Swingbench - http://www.dominicgiles.com/swingbench.html

[10] Hammerora - http://hammerora.sourceforge.net/

	bemerken
Public NIC Failure (beide NICs)	Kabel ziehen bzw. Switch Port deaktivieren. Clusterware sollte VIP relocation durchführen
Private NIC Failure	Kabel ziehen bzw. Switch Port deaktivieren. Bonding sollte Failover durchführen und Applikation nichts davon bemerken
Private NIC Failure (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 hat.
Interconnect Switch Failure	Stromkabel von Interconnect Switch ziehen.
Lost 1 connection path to storage	FC Kabel ziehen oder FC Switch Port deaktivieren. Multipathing sollte dazu führen, dass die Applikation weiterhin funktioniert.
Lost All connection paths to storage	FC Kabel ziehen oder FC Switch Port deaktivieren. Aufgrund fehlendem Disk Heartbeat wird der Cluster Node rebooted (Eviction)
1 Node verliert Storage Zugriff auf LUN der Voting Disk	Temporäres Disablen des Zugriffs eines Hosts auf LUN mittels Zoning.
1 Node verliert Storage Zugriff auf LUN des Control-Files	Temporäres Disablen des Zugriffs eines Hosts auf LUN mittels Zoning.
Lost ASM Disk	Herausziehen einer physikalischen Disk des Storage-Systems bzw. temporäres Disablen des Zugriffs eines Hosts auf LUN mittels Zoning und anschließende Behebung.
Entfernen einer Disk aus der Diskgroup und wieder Hinzufügen der Disk zur Diskgroup	Die ASM Disk soll entfernt und anschließend wieder zur selben Diskgroup hinzugefügt werden.
CPU Starvation	Durch Dummy-Prozesse wird das System in eine CPU Überlast-Situation geführt. Durch Implementierung bestimmte Clusterware / RAC Prozesse in der Realtime Scheduling Klasse wird das System dadurch nicht instabil.
Process Table overflow (fork bomb)	Das System wird einer "fork bomb" ausgesetzt. Das Kommando dafür unter Linux lautet: `:(){ :|:& };:`
Memory Starvation (Swapping)	Durch synthetisches Allokieren von Arbeitsspeicher durch ein C-Programm wird das System zum Swapping gezwungen. Solange allerdings genügend Swap-Speicher zur Verfügung steht, beleibt das System stabil.

Treten während diesen Tests Node Evictions auf, die nicht erwartet werden, sollte die Ursache dafür untersucht werden. Hilfreich hierbei ist MetaLink Note 265769.1 - "Troublesooting CRS Reboots", die anhang eines Flussdiagramms verschiedene Lösungsansätze gibt. Ursache für die meisten Node Evictions ist ein fehlerhaft konzeptionierter oder konfigurierter Cluster Interconnect. Oracle empfiehlt die Verwendung eines dedizierten Switches, an dem außer die

privaten Netzwerk-Interfaces der Cluster Nodes keine Komponenten angeschlossen werden. Zudem sollte verifiziert werden, ob RX Flow Control für die Netzwerk-Interfaces entweder auf „Autonegotiate“ oder auf ON gesetzt ist.

Sind diese Tests erfolgreich absolviert, kann ein Performance-Test der Applikation auf dem System durchgeführt werden. Eventuelle Performance-Bottlenecks können hier identifiziert und gegebenenfalls vor Go-Live behoben werden. Potentielle Kandidaten hierfür sind Sequences mit den Attributen *NOCACHE* und *ORDER* bzw. Block Contention durch sequentiell steigende Primary-Key Index Werte bei gleichzeitigen Inserts von mehreren Nodes. Durch verschiedene Techniken, z.B. reverse key index bzw. Applikation Partitioning, d.h. Verteilung von ähnlichen Applikationsmodulen auf selbe Nodes können die Probleme teilweise eliminiert werden.

Inbetriebnahme

Nachdem nun die Stabilität des Systems verifiziert wurde, steht einer Inbetriebnahme nichts mehr im Wege. Bei dieser Phase muß lediglich die Überwachung des Systems mit geeigneten Monitoring-Lösungen, z.B. Oracle Enterprise Manager Grid Control, implementiert und konfiguriert werden.

Troubleshooting

Für das Troubleshooting während der Testphase bzw. später im Betrieb können folgende Tools nützlich sein:

Tool	Beschreibung
OS Watcher	Das Tool „OS Watcher“ ist sehr zu empfehlen. Es sollte bei jedem Cluster-System ständig in Betrieb sein, um bei Node Evictions Informationen über die Belastung des Betriebssystems zu erhalten. Aufgezeichnet werden CPU Belastung, Memory Auslastung, private Interconnect Latency, u.v.m. Das Monitoring-Intervall kann neben der Historisierungsdauer frei definiert werden und historische Daten können automatisch komprimiert werden. Das Tool kann von MetaLink heruntergeladen werden und es existiert ein RPM Package um die Datensammlung auch nach Reboots automatisch zu wieder zu starten.
Procwatcher	Speziell bei sporadischen Problemen mit Oracle Datenbank bzw. Clusterware Prozessen, z.B. CPU Spinning, ist es notwendig, Oracle Support mit Stacktraces der Prozesse bei bestimmten Situationen zu versorgen. Das Tool procwatcher kann hier Hilfe leisten. Es kann konfiguriert werden, für welche Prozesse und in welchem Intervall ein „*oradebug short_stack*“ durchgeführt werden soll.
LTOM	Dieses Tool ist besonders bei Hängezuständen von Instanz bzw. Datenbank hilfreich. Es ermöglicht, bei Auftreten eines Hängezustands „*oradebug hanganalyze*“ auszuführen und damit die Ursache des Hängers zu ermitteln. Ebenso kann mit dem Tool

	Session SQL Tracing (Event 10046) für bestimmte Sessions unter bestimmten Bedingungen aktiviert und wieder deaktiviert werden.
Hang File Generator(HANGFG)	Kommt es zu Performance-Hängern muß der DBA rasch entscheiden, welche Diagnose-Mittel eingesetzt werden sollen, bevor das System entweder repariert oder neu gestartet werden soll. HANGFG unterstützt den DBA hierbei indem nur das Programm gestartet werden muß und HANGFG entscheidet dann, welche *hanganalyze* Optionen (Level) bzw. System State Dumps ausgeführt werden.
RAC-DDT	Um bei Problemen die benötigten Diagnose-Daten aus den verschiedenen Logfiles zu sammeln, kann das Tool RAC-DDT benutzt werden. Das erzeugte Output-Archiv kann dann zu Oracle Support hochgeladen werden.

Referenzen / Weiterführende Literatur

- Oracle White Paper - RAC System Load Testing Tools https://metalink2.oracle.com/cgi-bin/cr/getfile.cgi?p_attid=810394.1:RACSystemLoadTesting
- Oracle White Paper - RAC System Test Plan Outline
- MetaLink Note 810394.1 - RAC Assurance Support Team: RAC Starter Kit and Best Practices (Generic)
- MetaLink Note 811306.1 - RAC Assurance Support Team: RAC Starter Kit and Best Practices (Linux)
- MetaLink Note 361468.1 - HugePages on 64-bit Linux
- MetaLink Note 564580.1 - Configuring raw devices (multipath) for Oracle Clusterware 10g Release 2 (10.2.0) on RHEL5/OEL5
- MetaLink Note 265769.1 - Troubleshooting CRS Reboots
- MetaLink Note 563566.1 - gc lost blocks diagnostics

Kontaktadresse:

Martin Decker

ORACLE
10*g* Certified Master

ora-solutions.net
Franz-Fischer-Str. 7
D-81677 München

Telefon: +49 (0) 176 787 627 88
E-Mail martin.decker@ora-solutions.net
Internet: http://www.ora-solutions.net
Blog: http://www.ora-solutions.net/web/blog/